PACKET PRIORITIZING AND DELIVERING FOR MULTIMEDIA STREAMING

NGUYEN VU THANH

B.Eng. (1st Hons.), UTas

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2007

Acknowledgements

First of all, I would like to extend a special note of thanks to my thesis advisor, Dr. Chang Ee Chien, for his whole-hearted support and guidance. He not only teaches me what are the right things to do but more importantly, always shows his tremendous patience and encouragement when things go wrong. Without his enormous help, it would be impossible for me to accomplish this journey.

I particularly want to thank Dr. Ooi Wei Tsang, who has been my "de facto" co-advisor throughout these years. His extensive and profound knowledge of multimedia, theoretically as well as technically, is amazing. I am very fortunate to work with him in various projects, to be inspired, to learn and receive countless valuable advices from him.

I am indebted to Dr. Chan Mun Choon for his help on network technology; Dr. Chin Wei Ngan, Dr. Lee Wei Sun, Prof. Ooi Beng Chin, Prof. Tan Kian Lee and Dr. Yong Chiang Tay for their support and guidance; Dr. Roger Zimmermann, Dr. Wang Ye and anonymous reviewers for their valuable comments and suggestions. I am also grateful to Loo Line Fong and Theresa Koh at School of Computing, Tan Chui Hoon and Ho Hwei Moon at Registrar's Office, as well as many others in NUS for their generous and agile support.

My sincere thanks to my friends and collaborators at NUS and I2R — especially Cheng Wei, Gu Yan, Li Qiming, Ma Lin, Pavel Korshunov, Sujoy Roy, Ye Shuiming, Yang Xianfeng — for their sharing. Countless friends, whom I could neither list all of their names here nor single out any individual, have been always nice and fun to be with. Thank you for making my years, in and after NUS, memorable and enjoyable.

Last, but certainly not least, I would like to thank my family, especially my brother Dr. Nguyen Vu Thinh, and friends for always supporting, encouraging or just simply being there with me. Many have been very generous to spend their time to comment, edit and clarify my writing — Ankur Samtaney, Le Thuy Duong, my sister-in-law Nguyen Thi Thu Trang, Pham Quang Duc, Roma Singhal, and Tran Thi Minh Phuong. I am really thankful.

This thesis is dedicated to my fiancée Tran Thi Hai Thanh (My My) for her endless sacrifice, encouragement, understanding and love.

Nguyen Vu Thanh

Singapore, 27 November 2007

Contents

Summary

In this thesis, we investigated the problems of prioritizing and delivering packets in multimedia streaming. Under a lossy network, the sender has to decide which packets are to be further protected from losses, which packets are to be sent, how to send them, and when to send them. The priority of a packet could be either based on its position in the coding interdependencies (syntax-based) or based on its semantic content (content-based). We studied these problems under different network scenarios, with different types of information available to the sender and found that significant quality improvements could be obtained if a good packet allocation, protection and/or scheduling scheme is employed. Besides, content-based prioritization could greatly improve the perceived quality compared to syntax-based prioritization.

The main cause of quality degradation in multimedia streaming is packet loss. In Chapter 1, we present a review on common approaches that minimize the effects of packet loss, with a focus on transmission-based methods. We observed that user requirements and network characteristics are not as stringent as they are often described. For example, streaming audio and video can tolerate a one-way delay up to 10s, according to ITU standards. Such observation motivates us to investigate and compare FEC-based and retransmission-based delivery methods in better light, as well as lay the foundation for subsequent chapters.

Chapter 2 studies the problem of streaming multimedia packets over multiple paths. A common way is to use Multiple Description Coding (MDC) to create independent packets with similar quality contribution, thus any packet could be sent over any path. By using Layered Coding (LC, in which packets are implicitly prioritized

by grouping into different layers based on their interrelationships) instead of MDC, a sender could cleverly decide which packets to send over which path, therefore could provide much better quality under critical network conditions. We demonstrate this observation by observing the quality difference between streaming LC and streaming MDC over a two-path network. The experimental results show that with an optimal allocation scheme, LC provides significantly better quality than MDC, in contrast with what has been suggested in the literature.

In Chapter 3, we address the question of what to prioritize and argue that instead of prioritizing syntax data, we should prioritize the contents that are important to users. For example, in video surveillance, we can identify the regions of interest, where users are more likely to pay attention to. We found that prioritizing packets based on such regions can achieve dramatic quality improvement compared to syntax-based prioritizing. To objectively measure quality improvements, we propose a new performance metric called Focused-PSNR (F-PSNR). Our experiments show that content-based prioritization can provide videos with 6–11dB higher in F-PSNR than the standard method does. Subjective measurements with users also show a substantial improvement by using our methods (MOS of 7.8–9.2) instead of the standard one (MOS of 0.9–2.2). We then extend our content-based prioritizing scheme to consider FEC protection, and also find that content-based FEC can provide noticeable improvements compared to frame-based FEC.

Chapter 4 shifts the focus from packet prioritization and FEC protection to scheduling of prioritized packets. While highest-priority-first scheduling seems to be a natural way to stream prioritized packets, it only works best under severe network conditions, but with mediocrity in other scenarios. If the network condition is good (e.g., high bandwidth, low loss rate), earliest-deadline-first scheduling often provides significantly better quality. In most situations, good performance could be achieved by considering both highest-priority packet and earliest-deadline packet within a set of high-priority packets.

Surprisingly, although RTT is expected to have substantial influence on scheduling time, considering RTT in making schedule decisions is not that beneficial. Under our real-time streaming scenarios, we find that scheduling performance is not significantly changed with or without RTT consideration.

List of Figures

List of Acronyms

2D	Two-dimensional
3D	Three-dimensional
3G	Third Generation
3GPP	Third Generation Partnership Project
4G	Forth Generation
ACK	Acknowledgement
ADPCM	Adaptive Differential Pulse Code Modulation
ADSL	Asymmetric Digital Subscriber Line
AL	Adaptation Layer
AP	Access Point
ARQ	Automatic Repeat Request
AVC	Advanced Video Coding
BAM	Bandwidth Allocation Mechanism
BER	Bit Error Rate
BoD	Bandwidth on Demand
bps	bit per second
Bps	Bytes per second
CBR	Constant Bit Rate
CDMA	Code Division Multiple Access
CIF	Common Intermediate Format

CRC	Cyclic Redundancy Check
CSD	Circuit Switched Data
DCT	Discrete Cosine Transform
DMOS	Difference Mean Opinion Score
DWT	Discrete Wavelet Transform
DPCM	Differential Pulse Code Modulation
EDF	Earliest Deadline First
EDGE	Enhanced Data rates for Global Evolution
EoH	Earliest or Heaviest
EZW	Embedded Zero-tree Wavelet
FEC	Forward Error Correction
FGS	Fine Granularity Scalability
FOMA	Freedom of Mobile Multimedia Access
fps	frame per second
GOB	Group of Blocks
GOP	Group of Pictures
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HSDPA	High Speed Downlink Packet Access
HVS	Human Visual System
IP	Internet Protocol
IPTV	Internet Protocol Television
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ISP	Internet Service Provider
kb	kilobit (1000 bits)
kB	kilobyte (1000 bytes, not 1024 bytes - kibibyte)

kbps	kilobit per second
kBps	kilobyte per second
LC	Layered Coding
LLC	Logical Link Control
MAC	Medium Access Control
MDC	Multiple Description Coding
MOS	Mean Opinion Score
MPEG	Motion Pictures Experts Group
MSB	Most Significant Bit
MSE	Mean Square Error
MTU	Maximum Transmission Unit
NACK	Negative Acknowledgement
NTP	Network Time Protocol
OS	Operating System
P2P	Peer-to-Peer
PCM	Pulse Code Modulation
PDA	Personal Digital Assistant
PFGS	Progressive Fine Granularity Scalability
PPP	Point-to-point Protocol
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Intermediate Format
QoS	Quality of Service
QP	Quantization Parameters
R-D	Rate-Distortion
RLC	Radio Link Control
RLP	Radio Link Protocol
RS	Read-Solomon

RSVP	Resource ReSerVation Protocol
RTCP	Real Time Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
RTT	Round-Trip Time
RVLC	Reversible Variable Length Code
SCTP	Stream Control Transport Protocol
SIF	Source Input Format
SIP	Session Initiation Protocol
SNR	Signal-to-Noise Ratio
SSIM	Structural SIMilarity
SSNR	Segmental Signal-to-Noise Ratio
UDP	User Datagram Protocol
UEP	Unequal Error Protection
UMTS	Universal Mobile Telecommunications System
VBR	Variable Bit Rate
VLC	Variable Length Coding
VOD	Video On Demand
VoIP	Voice over Internet Protocol
VRC	Video Redundancy Coding
W-CDMA	Wideband CDMA

Chapter 1

Introduction

Study the science of art. Study the art of science.
Develop your senses – especially learn how to see.
Realize that everything connects to everything else.
—Leonardo Da Vinci

Digital multimedia has rapidly grown beyond personal, stand-alone entertainment applications to multi-users, network-based communication applications. When the first two audio and video standards MPEG-1 [125] and MPEG-2/H.262 [126, 130] were introduced, their main applications were for stand-alone entertainment such as Video-CD and digital TV. However in new multimedia standards such as MPEG-4 and MPEG-4 AVC/H.264, many efforts have been focused on communication and delivery over error-prone networks such as the Internet and wireless networks [127, 129, 139, 236, 262, 275]. Video conference, distance learning, Web TV and video phone over mobile networks are just a few examples of how multimedia, by connecting everything to everything else, could help to connect everyone to everyone else.

For the last few years, we have witnessed an exponential growth in the amount of multimedia data transferring over networks. At the initial stage of the Internet, most information is in the textual format; but nowadays, multimedia types such as image, audio and video are becoming increasingly important [39,287,296]. Voice over Internet Protocol (VoIP) is widely used not only by home users (e.g., via Skype, Yahoo! Messenger on PC, broadband cable) but also by corporates and telecommunication carriers for international phone calls. For example, in 2003, 11 percent of international calls (22 billion minutes) was carried using VoIP [285]. In 2005, VoIP traffic reached 19.4 percent (around 52.8 billion minutes) and just a year later, it already reached 24.2 percent (around 75.8 billion minutes) [249]. Meanwhile, various video services are offered by an increasing number of content providers and cable companies (e.g., BBC, CNN, Reuters, CNET, MTV Networks, CinemaNow, Comcast, etc). Akamai, the largest content distribution network, reports that the video streaming traffic of a normal media site doubles every six or eight months [27].

At the same time, ones can also observe the enormous development of wireless communication and portable devices (laptop, smart mobile phone, tablet PC, etc.). In 1997, it was expected that the wireless cellular networks would support IP-based multimedia applications such as mobile internet access, mobile video conference, streaming video/audio, distance learning [335]. At the end of 2001, this expectation partly became true when the third-generation wireless systems (3G networks) – with high-speed data and Internet access, multimedia data transmission and packet-switched core network – began the service in Japan [150]. By April 2006, 3G services have been served over 84 countries to 266 millions subscribers (among 2.16 billion mobile customers) [284]. It is an inevitable fact that wireless and mobile communications will be an essential part of our life.

Along with the convergence of communications, computing and entertainment, we can expect that an increasing number of multimedia services will be streamed over networks. Some services like IP Television (IPTV), Video On Demand (VOD) are normally carried through dedicated cables or satellite links with small loss ratio and high bandwidth. However, many would be delivered over the Internet and wireless networks with time-varying, unpredictable characteristics and often high packet loss ratio (due to congestion, delay, fading, etc.).

In order to find out how to maintain and maximize the streaming quality in these lossy and changing environments, numerous approaches have been proposed, e.g., increasing the error-resilience of data bit streams [236, 304], assuring a guaranteed resource [17, 42, 101, 190, 329], and concealing the effects of loss at receivers [224, 298, 305]. In this thesis, we focus on packet transmission, particularly to find (i) how to optimize packet allocation over path diversity, (ii) how to prioritize packet – based on its semantic content, syntax data, or both, and (iii) how to schedule prioritized packets to maximize the output quality.

To have a better understanding how our works fit in the overall picture, we will briefly describe a general multimedia streaming system in Section 1.1. This also helps Section 1.2 to explain why bit error, network fluctuations may lead to packet loss, and in turn how packet loss could severely affect the received quality. In Section 1.3, some common approaches to minimize the effects of packet loss are shortly described and discussed. Our research problems are presented in Section 1.4, together with the thesis organization and its contributions.

1.1 Overview of a general multimedia streaming system

Figure 1.1 presents a general multimedia streaming system. Interested readers could refer to [8, 208, 237, 287] for detailed information.

At the sender side, original data (audio, video, image) are either captured directly from sources or read from storage devices. To reduce the data rate, data are then encoded (for raw data) or transcoded (for stored data) if necessary by corresponding source encoder. The compressed bit streams are then divided into packets of fixed

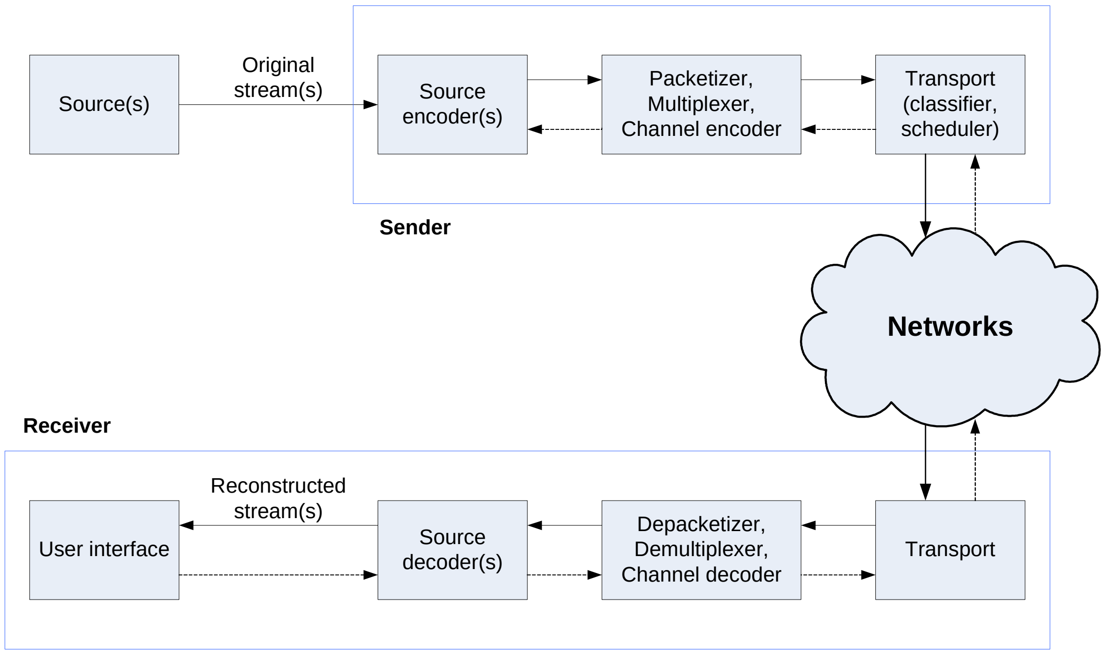


Figure 1.1: A general multimedia streaming system (adapted from [8, 208, 237, 287]).

or variable length. Packets of different types (audio, video) could be multiplexed to form one or several transport streams. After that, channel-encoding or error-resilient tools such as Forward Error Correction (FEC) could be applied to protect packets from transmission errors or losses [73].

Packets could also be classified and assigned different priorities so that appropriate level of protection could be allocated, or a packet scheduler could decide their sending order. They are then transmitted to network using transport protocols such as User Datagram Protocol (UDP), Transmission Control Protocol (TCP) or Real-time Transport Protocol (RTP) over UDP [226, 227, 251]. Note that at transport or lower layers, FEC could also be used while Cyclic-Redundancy Check (CRC) is normally utilized – optionally in UDP or by default in Ethernet frame, TCP, IPv4, etc. – for error checking.

At the receiver side, packets are received by corresponding transport protocols. Error and loss detection techniques could be applied to check whether a packet is corrupted or lost. The corrupted/lost packet could be recovered by error and erasure correction methods, or be requested for retransmission. The receiver can also decide to ignore erroneous/lost packets and jump to the next re-synchronization point. After this channel-decoding stage, packets are demultiplexed if necessary, and unpacked to reform the original compressed data stream(s). Error-concealment methods could be applied before or during source decoding process to reconstruct the original data.

The dash line in Figure 1.1 indicates that feedback could be used during the streaming process. For example, receiver's transport layer could send feedback such as retransmission requests, link measurement parameters, to the sender's counterpart. Users could send feedback on which data stream is more important to them so that the sender's classifier and scheduler may act accordingly. Packetizer, channel-encoder may feedback to the source encoder to better adapt with network conditions, and in many cases they could be built in the source coder for network adaptation. That is, the boundaries between different stages (components) of the streaming process are not always rigid, and in fact, they are increasingly designed to cooperate with and blend into each other [63, 157, 257, 295, 321, 326].

1.2 Packet loss

Multimedia, especially video, data in the raw format contain high redundancies and have to be compressed before transmission. In order to achieve high compression ratio, most encoding schemes reduce spatial similarity within a frame (e.g., DCT or DWT for video) and temporal redundancy between consecutive frames (e.g., by DPCM, ADPCM for audio, by motion estimation for video). The redundancy between data symbols is then further reduced by Variable Length Coding (VLC) methods such as Huffman and arithmetic coding [85, 111, 207]. Consequently, we have a pervasive dependency structure within encoded bit streams. That is the reason why if a packet is lost, its subsequent dependent packets could be useless and the quality of video signals may be severely affected [40, 199, 289].

At the bit level, a packet may be corrupted by some errornous/lost bits caused by link impairment. Consequently, the VLC codewords containing these bits and the following codewords (until the next synchronization marker) would be unable to be decoded. Therefore synchronization markers are periodically inserted into the bit streams, normally at the beginning of every packet. Video standards like H.263 and MPEG-4 even incorporate Reversible VLC to decode the bits before the synchronization markers in backward direction [304, 309]. Bit errors could be detected by CRC and then corrected by FEC, but only when sufficiently strong codes are used. If the error/loss is unrecoverable, the packet is still considered lost and retransmission could be required.

Beside packet loss due to bit errors, packets may be dropped by senders, network nodes or be late. Since the characteristics of network links (especially Internet and wireless networks) are time-varied, unpredictable and often lossy, it is inevitable that some multimedia packets will be lost during transmission. For example, if bandwidth is suddenly decreased and no longer enough to send all data packets, some packets will be dropped or even not be sent. Congestion at network bottle-necks also creates buffer overflow at routers and forces the routers to drop packets. Besides, network congestion may prevent packets from arriving before their deadline, thus make these late packets useless for the receivers.

To receivers, all of these irrecoverably corrupted, dropped, or late packets are useless. Henceforth in this thesis, what we mean by "a lost packet" – except when it is stated otherwise – is a packet unavailable or useless for decoding, regardless of its causes. Because of the harsh quality degradation created by packet loss, minimizing its effects is an important issue in multimedia streaming.

1.3 Approaches to minimize packet-loss effects

Packet loss may occur due to various reasons; therefore, its effects could be minimized by using various techniques. For example, to reduce packet loss due to bit errors, we could apply strong error correction to protect the packet, or send it over a better link if path diversity is employed [12, 175]. To prevent a packet from being late, senders could transmit the packet much earlier than its deadline so that if it is lost, there would be enough time for retransmission. Senders could also monitor network conditions and adjust their sending rates accordingly to reduce the probability of packet drop. On the other hand, receivers could reserve and be guaranteed a sufficient bandwidth for their streams by using Resource ReSerVation Protocol (RSVP) [42] or other bandwidth allocation mechanisms [108, 117]. Furthermore, error-concealment could be used at receivers to minimize loss effects, for example, by replacing the lost packet by its preceding one or using spatial interpolation. Some common approaches to minimize the effects of packet loss for the Internet and wireless communications are summarized in Figure 1.2 (partly adapted from [92, 250], with substantial additions).

From Figure 1.1, we could roughly categorize these techniques based on their focus areas, as follows: (i) encoding-based methods, (ii) transmission-based methods, and (iii) decoding-based methods. By "encoding-based method", we mean those error-resilient coding schemes that are mainly employed at the encoder [304]. Transmission-based methods are those closely involved with packet transmission such as transport protocols, error-resilient techniques at low layers, loss prevention and recovery methods. Decoding-based methods at receivers comprise of loss recovery and error-concealment methods at the receiver side [224, 305].

1.3.1 Encoding-based methods

An effective strategy to counter with packet loss is making encoded bit streams more error-resilient during the source-encoding process and/or channel-encoding process.

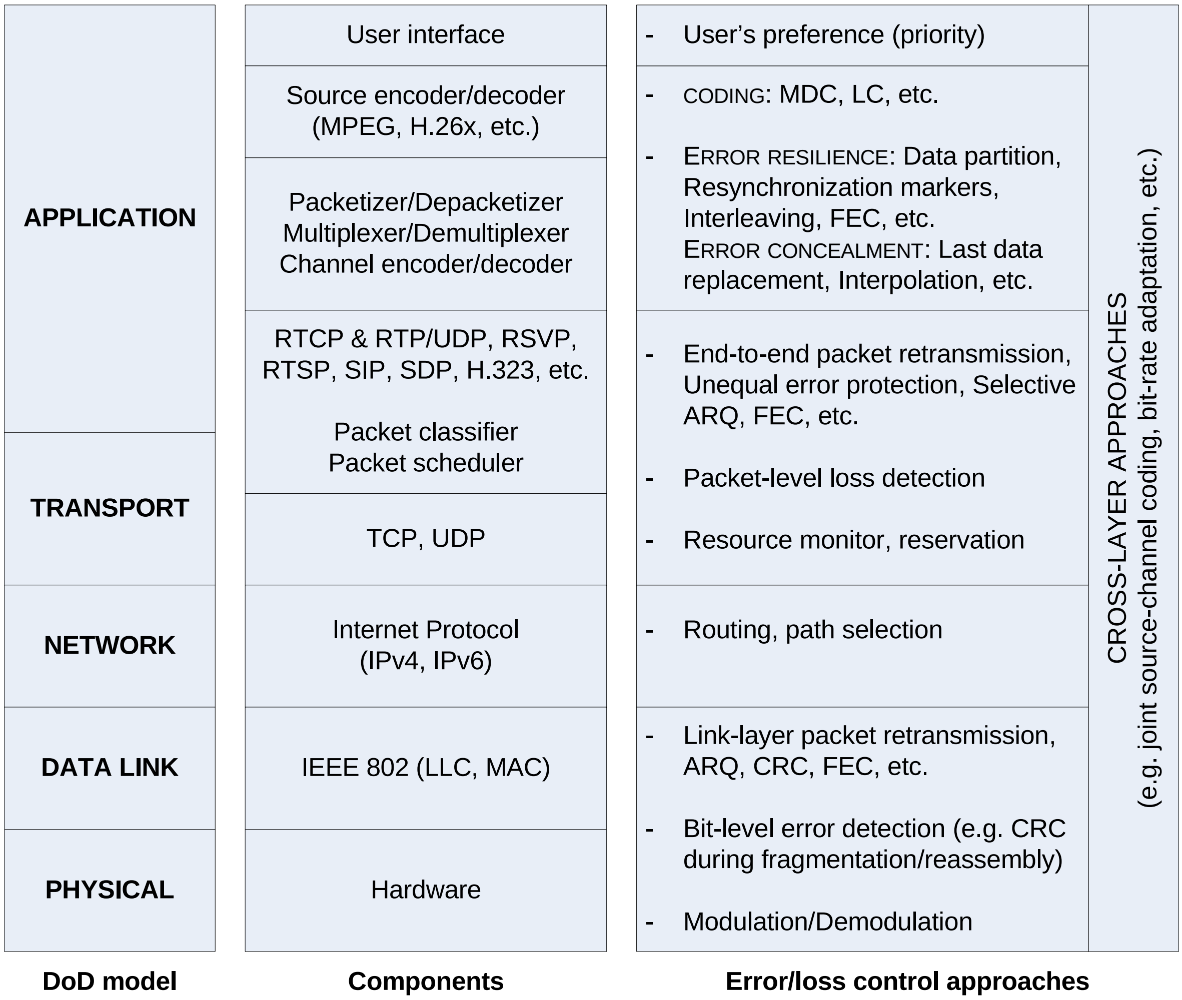


Figure 1.2: Approaches to minimize packet-loss effects.

While some methods solely work with source encoder, others such as Layered Coding (LC), Multiple Description Coding (MDC) and 3D subband coding require joint cooperations of source and channel encoders.

In video coding, the simplest approach is using more independent coding of each frame, for example, using all I-frames, re-initializing the prediction loop periodically by inserting one I-frame after certain number of frames (MPEG GOP), or partially intra-encoding each frame. Although these methods are effective in error control, they are expensive to apply due to their low compression ratio and substantial overhead.

A popular approach is *Layered coding (LC)*, which is firstly proposed by Ghanbari [201]. It is further developed and adopted in MPEG-2/H.262, JPEG2000, MPEG-4, MPEG-4 AVC/H.264 standards and bears various names such as scalable/multi-resolution/embedded/progressing coding [8,85,111,128,236]. In this approach, source data are partitioned into a base layer and a few enhancement layers with different priorities. The base layer contains the most important data and decoding only this layer can provide an acceptable perception quality. The enhancement layers deliver complementary information to combine with the base layer for offering higher-quality output. These low-priority layers could be lost or cleverly discarded without losing the core information. However, an error in the base layer may severely affects the successful reconstruction of the original data. Therefore, if networks are lossy and have no priority support, strong protection should be applied to the base layer, e.g., by using stronger FEC or more number of retransmissions.

There are several ways to realize layered coding, e.g., data partition, temporal scalability, spatial scalability, SNR scalability or hybrid form. Example of SNR scalability could be found in the works by Liang et al. [175] and Wang et al. [302], where audio data could be encoded either at a coarse quantized level to form base layer or at a finer quantized level to form enhancement layer. In the simplest form of temporal scalability, I-frame and some P-frames in MPEG video could form the base layer, while other B-frames become the enhancement layer [111]. Fine Granularity Scalability (FGS), Progressive FGS tools in MPEG-4 video standards allow to create two-layer structure by bit-plane DCT-based coding or wavelet, in which the base layer is encoded with a bit rate Rb and the enhancement layer is fine-granularly coded to a maximum bit rate Re [85, 234, 237].

While LC uses layers with different priority, *Multi Description Coding (MDC)* divides source data into multiple equally-important streams [303]. Any subset of these streams can be independently decoded into a baseline signal and provide a reconstructed output in a certain desired fidelity. The more descriptions are received, the better reconstruction quality is achieved. Because LC stream is sensitive to the position of loss in the bitstream, a certain lost packet could make its subsequent packets useless, therefore, may severely affect the received quality. Meanwhile, MDC quality depends only on the fraction of packets received, thus is considered to be more suitable for noisy and unreliable channels. To assure that the probability of losing all the descriptions is small, MDC streams are normally transmitted over multiple paths [13, 14, 179, 192, 193, 209].

To create MDC bit streams, several ways could be applied such as sub-sampling in spatial/temporal/frequency domain [21, 159, 299, 309], using multiple-description (MD) quantization [286, 291, 292], MD transform coding [115, 240, 300, 301] or MD-FEC [9, 202, 203, 231, 232]. An example of temporal frame interleaving technique called Video Redundancy Coding (VRC) based on Reference Picture Selection is proposed by Wenger et al. [309]. While MD quantization methods assign a pair of indexes to quantizer's output to produce two descriptions, MD transform coding divides coefficients into two descriptions with some redundancy between them using a correlating transform. A more popular method is MD-FEC, in which a scalable bit stream is divided into different parts and FEC is applied across these parts to create multiple equal-quality descriptions. Interested readers could refer to [231] for more information.

While most video standards are using motion-compensated hybrid with DCT transform, *3D subband coding* with wavelet transform has attracted numerous researches recently [37, 57, 91, 98, 158, 253, 282, 316], and has been partly adopted in JPEG2000 and MPEG-4 AVC/H.164 standards [262, 275]. In this approach, signals are divided into a number of subbands spatially and temporally, then encoded using wavelet techniques such as embedded zero tree wavelet (EZW) [258], set partitioning in hierarchical trees (SPIHT) [245] or embedded block coding with optimized truncation (EBCOT) [281]. While motion-compensated hybrid video codecs may cause "drift problem", motion-compensated spatiotemporal wavelet coding schemes do not employ any recursive prediction loop and may create more efficient scalable bit streams [211]. For example, LC structure could be realized by incorporating with Unequal Error Protection (UEP) with 3D subband encoders to provide different protection levels to different subbands as well as bit-rate scalability in a more natural way than traditional encoders [255]. Similarly, MDC structure can also be obtained by appropriately interleaving 3D subbands among packets, so that every packet can be independently decoded and has approximately equal expected visual importance [278,279]. On the other hand, 3D subband techniques require larger memory and additional computational complexity at receivers for decomposing temporal subbands, which are undesirable for those receivers with limited power and computing capability.

1.3.2 Transmission-based methods

In order to achieve good performance, it is important for all approaches, especially transmission-based methods, to understand and adapt to their working environment as well as to user requirements. We firstly describe main characteristics, e.g., bandwidth, round-trip time, packet loss ratio (or loss rate), of some common network connections. User requirements for different application classes – specified in standards of the ITU Telecommunication Standardization Sector (ITU-T), the Internet Engineering Task Force (IETF), 3rd Generation Partnership Project (3GPP) – are also briefly presented.

Transmission-based methods to decrease the effects of packet loss could be roughly divided into three sub-categories: (i) supporting, (ii) prevention, and (iii) recovery. Supporting methods are network tools, protocols or architectures which, for example, could monitor (e.g., bandwidth estimation) or guarantee (e.g., bandwidth reservation) network conditions to support other methods. Prevention methods are normally employed at the senders and aim to reduce packet-loss's effects, either by reducing

Type		Maximum downlink data rate	Typical downlink data rate
Wired connection	Dial-up access via telephone line	56 Kbps	33 – 53 Kbps
	ISDN	64 – 2048 Kbps	
	ADSL	128 Kbps – 24 Mbps	115 Kbps – 8 Mbps
	Cable television	2 – 25 Mbps	
Wireless connection	2G cellular – CSD on GSM (1990s)	9.6 Kbps	
	2G cellular – cdmaOne/IS-95 (IP-based)	76.8 Kbps	
	2.5G cellular – GPRS on GSM (packet)	171 Kbps	4 – 80 Kbps
	2.75G cellular – CDMA2000 1x (UMTS)	144 Kbps	
	2.75G cellular – EDGE (Enhanced GPRS)	384 Kbps	160 – 238.6 Kbps
	2.75G cellular – EDGE ph 2 (Real-time IP)	473 Kbps	
	2.75G cellular – Enhanced EDGE	2 Mbps	
	3G cellular – FOMA (W-CDMA, 2001)	384 Kbps	200 Kbps
	3G cellular – UMTS (W-CDMA & GSM)	1920 Kbps	384 Kbps
	3G cellular – CDMA2000 1xEV-DO	2.4 – 3.1 Mbps	600 Kbps
	3G cellular – CDMA2000 1xEV-DV	3 – 4.8 Mbps	
	3.5G cellular – HSDPA on UTMS (2006)	14.4 Mbps	
	4G cellular (testing in Japan)	100 Mbps – 1 Gbps	varied
	WLAN IEEE 802.11b	11 Mbps	negotiable & varied
	WLAN IEEE 802.11a/g	54 Mbps	22 – 26 Mbps
	High data rate WLAN	100 Mbps	

Figure 1.3: Typical data rate of different types of link (combined from [83, 110, 238, 276]).

the probability of loss (path diversity) or by minimizing the damages (FEC, packet scheduler). Meanwhile, recovery methods often reside at the receivers to recover lost packets, e.g., by requesting for retransmissions.

1.3.2.1 Network characteristics and user requirements

It is well known that characteristics of best-efforts networks like the Internet and wireless networks are time-varied and unpredictable. For the Internet, there are no guarantees on bandwidth, transmission delay, delay jitter and loss ratio. For wireless networks, the variations in bandwidth, delay and bit-error rate are even higher [330]. It is because for wired networks like the Internet, the main reasons for packet loss are network congestion and delay; however for wireless networks, bit corruption due to multi-path fading, interference, and attenuation are also important factors [4].

Typical data rate of some wired and wireless connections are shown in Figure 1.3 [83,110,238,276]. Compared to the data rate required by normal video signals (56 Kbps–2 Mbps for QCIF and CIF, 1.5 Mbps or higher for MPEG-2/MPEG-4 videos [107,112]), it is clear that some types of links are inadequate for video streaming. To cope with the problem of bandwidth insufficiency and fluctuations, approaches like rate allocation, buffer management, resource reservation are normally employed.

For packet loss ratio, many measurements have been taken and published with different numerical results, since measured networks are different in link quality, network load, number of nodes, and distance between nodes, etc. For example, one-hop-link loss ratio for an in-building wireless LAN (AT&T WaveLAN) was reported to be around 0.01–0.14% [86]. When streaming MP3 music over a IEEE 802.11b based indoor wireless ad hoc networks with 4 nodes, [187] (2002) reported a loss ratio of 0.3–9.1%, which varied depending on the routing protocols and node locations. For multi-hop 802.11b indoor and outdoor wireless networks with 29–32 nodes, MIT's researches (2004–2005) [4, 29] reported an average packet loss ratio of 50% (at link layer, without ACK and retransmission). For the Internet, the average packet loss rate is reported to be around 3–9% in 1994–1995 by Paxson [220, 223]). Analysis of the connections between 31-49 hosts (most are universities, research institutes in USA) during two winters 1999–2001 [332] reported an average loss ratio of about 0.6–0.87%. It also found that the packet loss ratio of most links was less than 1%, 12–15% links had loss ratio of 1–10% and less than 1% links had a loss ratio higher than 10%. During 2006–2007, the average packet loss rate reported by Internet Traffic Report [241, 242] is around 8–12%.

Transmission delay, reflected in RTT value, is also varied. For 3G wireless networks like W-CDMA or high-speed cable connections, RTT is typically less than 100ms [92]. For the Internet, the average RTT is reported to be about 134–160ms [74, 241]. Dial-up connections normally have higher latency, about 200–400ms or even up to 600ms, while GPRS connections could have RTT from 600 to more than 1000ms.

One important issue is the constancy of Internet behavior, i.e., for how long we could reasonably assume that the network properties are unchanged. Various researches on this problem have been published [34, 36, 220, 221, 317] and an excellent study is presented by Zhang et al. [332], in which the traffic between 31–49 hosts from different university/institutes is collected during two winters (1999–2001) and analyzed extensively (a similar study is carried out by Paxson during 1994–1995). End-to-end throughput is reported to behave quite stable (90% of the time, it is steady for 20 minutes or less) and not wildly fluctuate in a minute-by-minute manner. For packet loss, there is a high probability that a packet will be lost if the packet sending 500–1000ms earlier is lost, and vice versa. On the long range, the loss behavior of the Internet could be well modelled by Markov-Gilbert model, and the loss spikes are normally very short (95% of losses are shorter than or equal to 220ms). Besides, it is found that about half of the time, a constancy region, in which loss ratio is virtually unchanged, of 10 minutes or less could be found. Packet delays also experience spikes of highly evaluated RTT intervening between steady periods, however delay's behavior is often less steady than loss's behavior. Overall, the properties of Internet path could be generally expected to be steady at least on the time scale of minutes.

While different type of network connections create different bandwidth, packet loss ratio and delay conditions, users also have distinct requirements for different application classes. Various standards have been published by ITU-T, IETF and 3GPP to offer guidelines on such requirements [2, 3, 134, 137]. A summary of network performance objectives for some common multimedia applications, adapted from [134], are shown in Figure 1.4.

Class	Application	Degree of symmetry	Typical data rates	Key performance parameters and target values		
				One-way delay	Delay variation	Packet loss ratio
Interactive (delay << 1s)	Conversational voice and video	Two-way	Audio: 4-64 Kbps Video: 16-384 Kbps	< 150 ms (preferred) < 400 ms (limit)	< 1 ms	Audio: < 3% Video: < 1%
	Interactive games	Two-way	< 1 KB	< 200 ms	N.A	Zero
	Command/ control	Two-way	~ 1 KB	< 250 ms	N.A	Zero
Responsive (delay ~ 2s)	Voice/video messaging	Primarily one-way	Voice: 4-32 Kbps	< 1 s (playback) < 2 s (record)	< 1 ms	< 3%
	WWW browsing	Primarily one-way	~ 10 KB	< 2 s / page (preferred) < 4s / page (acceptable)	N.A	Zero
Timely (delay ~ 10s)	Streaming audio and video	Primarily one-way	Audio: 16-128 Kbps Video: 16-384 Kbps	< 10 s	<< 1 ms	< 1%
	Still image	One-way	< 100 KB	< 15 s (preferred) < 60 s (acceptable)	N.A	Zero

Figure 1.4: End-user QoS classification and requirements (ITU recommendation G.1010 [134]).

1.3.2.2 Supporting methods

To support multimedia transmission over networks, there are two general directions: network-centric and end-to-end approaches. Both could be used to monitor the network characteristics – e.g., bandwidth, packet loss ratio, round-trip time (RTT) – so that senders and receivers may adapt their sending/receiving policy. Network-centric approach requires the participation and support of intermediate network routers/switches along the transmission path. These not only could monitor network properties but also may guarantee the network conditions if required, e.g., by employing QoS architectures like Integrated Services [41] and Differentiated Services [31]. For example, bandwidth could be reserved and allocated by Resource ReSerVation Protocol (RSVP) [42, 329] and other bandwidth allocation mechanisms (BAM) [16, 94]. One problem with this approach is that it requires enormous deployment of intelligent routers over the Internet. However, router manufactures and ISP companies seem to have both economical incentives and technical abilities to overcome such obstacle. A bigger problem is the strong opposition from customers, content producers and press, since such deployment will violate the Network Neutrality principle of the Internet and likely lead to network discrimination [71, 252].

Meanwhile, end-to-end approach is based on the cooperation between senders and receivers without altering the network architecture or heavily relying on the QoS support of intermediate network devices. Therefore, it may provide higher flexibility and adaptability, since applications know best what their requirements are, how packets are related to each other, and which packets are important [69, 70, 217]. For example, end-to-end transport protocols like Real-time Transport Protocol (RTP) and its companion Real Time Control Protocol (RTCP) [251], and Stream Control Transmission Protocol (SCTP) [87, 270] could monitor network conditions to adapt their transmission policy. However, network devices normally provide broader and more accurate information about network conditions, thus it would be desirable for senders and receivers to utilize such information in their decision making. Some excellent, fundamental arguments about end-to-end and network-centric approaches could be found in [28, 32, 56, 246].

To reduce the probability of network congestion, senders and receivers could adapt their sending/receiving rates to network conditions. For example, the sender may increase quantizer step (H.261, MPEG-2) or reduce the frame rates (H.263, MPEG-4) in the encoding process to decrease its sending rate [312]. It could also use rate shaping techniques such as selectively discarding frames or unimportant DCT coefficients, or employ other scalable rate control methods to ensure that its sending rate will not exceed the available bandwidth [90, 152, 169, 225]. Besides, the sending rate could also be determined by TCP-friendly formulas [99, 218, 290], which require information on Maximum Transmission Unit (MTU) [200], RTT and packet-loss ratio. On the other hand, receivers could control their receiving rate by deciding which layers they want to subscribe to if layered-coding data are streamed [44, 310].

No matter how sending/receiving rates are determined, they all should be upper-bounded by the available bandwidth of the transmission path. There are various tools to estimate this available bandwidth, and some general reviews of these tools have been published [124, 148, 229, 273]. The most popular method to estimate this value is based on packet-pair principle, which is proposed by Jacobson [146] and is further studied by others [171, 181, 264, 265]. Particularly, it could be achieved by sending sequences of probing packet trains, then observing the time interval between the head and tail packets of each train, which will increase if the available bandwidth is less than the transmission rate or remain unchanged otherwise. Other tools such as Pathchar [147], Pathneck [123], Cartouche [120], BFind [7] could even locate position of the bottleneck link. Note that Pathchar, Cartouche and BFind estimate the *capacity* of the bottleneck link (which is determined by its physical layer), not the *available bandwidth* of the transmission path (which is the bandwidth that could be used without affecting other data flows on the link) [148, 167]. Further information about how to avoid mistakes and conduct a sound measurement could be found in the works by Jain et al. and Paxson [149, 222].

Packet loss ratio could be estimated either by network routers or by senders and receivers. For example, routers could employ Simple Network Management Protocol

(SNMP) [47] to passively monitor packet loss ratio within their domains. On the other hand, senders and receivers may estimate packet loss ratio by counting packet ACK or NACK at senders, or by monitoring packet sequence numbers at receivers. They may also use `ping` utility [206], utilize RTCP feedback messages which are normally reported every several seconds (e.g., 5s), or use average loss intervals to estimate the packet loss ratio like in TCP-friendly Rate Control (TFRC) [100, 119]. Interested readers may refer to the works by Paxson and Sommers et al. [222, 268] for more information on how to improve the measurement accuracy.

RTT could also be estimated using the tools mentioned above, normally by sending probe packets such as IMCP echo request packets (`ping`), UDP [34] or TCP packets [323] and observing the timestamps of the feedback messages. Receivers in multicast sessions could employ a scalable approach proposed by Sisalem and Wolisz [266] to estimate the round-trip time to the sender.

1.3.2.3 Prevention methods

Transmission-based prevention methods, normally implemented at the sender's side, aim to reduce the effects of packet loss during transmission. This could be achieved mainly by two ways: (i) reducing the probability of loss – e.g., by routing, transmitting packets via multiple paths, and (ii) reducing the damage extent of packet loss, for example, by adding channel protection, joint source-channel coding, interleaving or careful packet scheduling.

To reduce the loss probability of a packet, one way is to send it over the highest-quality path, which is either pre-determined by the sender and stored in the packet's header (e.g., IPv6) or decided by intermediate routers. A measurement-based study shows that "in 30–80% of the cases, there is an alternate path with significantly superior quality" [248]. However, applying the first option to the Internet is difficult, since (i) network conditions are normally unpredictable so the chosen path may become worse, thus (ii) an overlay network may be required, but (iii) storing all routers' addresses in packet header may create large overhead and security problems. The second option also suffers from the unpredictable nature of networks and requires a differentiated support from routers.

One simple way to reduce the probability of loss is to send multiple copies of that packet. However, it is expensive and often inefficient to send all these packets over one path, since they could easily be lost if the path is congested. Another idea is *path diversity*, which was first studied by Dolev in 1982 [81,216,235] and then was extended to multimedia streaming by Apostolopoulos [12], in which different (or same) subsets of packets are sent to the receiver over different paths. Since the probability of all channels being congested at a given instant is much less than that of a single channel, sending through multiple ways can provide an average path behavior and improve the transmission quality.

Several questions have to be addressed in order to successfully employ path diversity. If same packets will be sent over all paths, two main questions are (i) how to select different and disjoint paths, and (ii) how to assure packets will travel via the selected paths? If different packets will be sent over different paths, an additional question is (iii) how to decide which packet will travel through which path? The path selection question has been extensively studied in various works [11, 24, 25, 261, 283]. The second problem could be solved by overlay, application-level routing, or source routing in IPv6, etc., [10, 18, 19, 68, 77, 185, 197, 247]. One part of this thesis focuses on the third question, which will be further described in Section 1.4 and Chapter 2.

To reduce possible damages due to packet losses, one strategy is letting packets go through a channel coding process, in which FEC codes are added [1]. For example, one could use parity codes to protect every n packets by a redundant packet, or add redundant information of previous packets into the current one [35]. FEC codes such

[1]Note that channel coding is normally performed at application or transport layer, and FEC is applied for block of packets to prevent packet loss. Meanwhile, at link layer (e.g., of satellite systems), FEC is often applied within packets to detect and correct bit errors.

as Reed-Solomon and Tornado codes [30, 239] could be used to create n packets from $(n-k)$ original packets so that the original data can be recovered if less than k packets are lost. If the loss is greater than k, only a portion or none of the lost data may be recovered, depending on the type of FEC being used. In mobile communications where the raw loss ratio is normally around 5–10%, typical value of code rate is from 1/6 to 1/2 information bit/signal [73].

Since the original data could only be decoded until sufficient number of packets $(n-k)$ have been received, a delay would be introduced. Besides, FEC operations also require a certain computational power and sufficient memory buffer. Therefore, the capability of FEC would be restricted not only by the application's delay constraints [326], but also by the capability of senders and receivers.

The main problem with FEC-based strategy is that it is designed with a predetermined channel-loss threshold, i.e., to overcome a specific amount of loss. If the channel condition is better than the predicted condition, it will become inefficient since the redundancy is more than actual need. Inversely, it is ineffective (cannot recover the lost data) if the channel loss is larger than the expected level. Hence, FEC-based strategy is optimized only when it can adapt to channel loss ratio, which is normally time-varying and highly dynamic.

How to determine an optimal bit rate allocation between channel coding (e.g., FEC) and source coding, given a constrained bit budget and changing network conditions, normally requires a joint source-channel coding approach [84, 93, 109, 161, 327]. In fact, joint source-channel coding approach could be considered a special case of a more general direction: *cross-layer design* [269, 321]. In this direction, information is allowed to be exchanged between various layers of the protocol stack to optimize the system performance, e.g., in multimedia quality and energy adaptation [75, 88, 260, 320], modulation and demodulation at radio link [63] or packet classifying and scheduling [182, 186, 328]. Numerous researches on joint source-channel coding and cross-layer approaches have been published in recent years and excellent reviews could be found in [157, 257, 269, 295, 326]. In this thesis, the FEC allocation problem is addressed in Chapter 2 and Chapter 3.

The third way to reduce the damage extent of packet loss is through packet scheduling, i.e., purposely choosing the sending time of packets to reduce the possible effects of loss. It has been observed that instead of sending two copies of a packet over multiple paths, similar result may be achieved by sending both over the same path with a 10–20ms delay between them [11]. Similarly, interleaving has been proved to be an effective way in reducing loss effects in various applications, from multimedia streaming [173, 196, 306] to wireless and mobile transmission [15, 62, 73]. It is because network losses often occur in burst, interleaving could spread packets to avoid the loss of several consecutive packets, which creates more severe effects than what could be done by the loss of several separated packets [40, 174].

However, an effective packet scheduling method should be much more than simply sending copies of original packets or interleaving them, which often treat all packets in the same manner. Because different multimedia packets normally have different deadlines and values (which are also changed over time, e.g., become null if the packets' deadlines are over), it is necessary to schedule packets individually based on their characteristics. Furthermore, interleaving can only cope with a low packet loss ratio; and for sending copies of packets over a same path, the number of copies and the delay between them should be decided based on current network conditions. That is, intuitively, packet schedulers should know not just about characteristics of packets, but also about network characteristics at their sending times. We will talk more about this in Section 1.4 and Chapter 4.

1.3.2.4 Recovery methods

By definition, transmission-based prevention methods like FEC, path diversity delivery, interleaving are performed before the transmission of packets[2], whereas transmission-based recovery methods like retransmission, ARQ are carried out after knowing that packets were corrupted or missed (lost in transmission).

To detect a *corrupted* packet, checksum such as parity, Cyclic-Redundancy Check (CRC) [30, 228, 271], Fletcher's checksum [97, 336] or Adler-32 [78, 272] are calculated and added to the packet at various protocol layers. At the link layer, CRC is applied to MAC frames in wired LAN (Ethernet or IEEE 802.3) and wireless LAN (Wi-Fi or IEEE 802.11), from Personal Area Networks like Bluetooth (IEEE 802.15) to Wide Area Networks like WiMAX (IEEE 802.16), or mobile networks like GSM and Wideband CDMA (W-CDMA). At the network layer, IPv4 header is validated by a Header Checksum field of 16 bit one's complement, which is checked (packet with invalid IP header will be discarded) and updated wherever the packet's header is modified (e.g., inside routers where the packet is not protected by link layer's checksum)[3]. At the transport layer, UDP and TCP segments also use 16-bit one's complement checksum (optional in UDP) to check UDP/TCP header, IP header, addresses (in TCP), and data.

At receiver's low layers, erroneous packets would be detected by integrity checking and if they are unrecoverable, retransmissions are automatically requested [92]. For example, Radio Link Control (RLC) frames are allowed to be retransmitted in CDMA2000, and MAC frames retransmissions are widely used in 3G, 4G systems as well as wireless LAN standard [55, 64, 178, 188, 293]. TCP segments with invalid header checksum could also be automatically retransmitted after a timeout.

Detecting a *missing* packet – a packet has been sent but lost in transmission

[2]For convenience, the term "packet" would be used to describe network-layer packets, transport-layer segments, or data link frames.

[3]The growing protocol IPv6, which will be adopted by US government in 2008 [96], omits this field and relies on error checking from other layers instead.

and could not reach the receiver – requires different techniques. Fast checking and retransmission at MAC layer are only useful when the packet, though corrupted, is received. However, detecting the missing packet can not be performed at MAC layer but have to rely on higher layers, for example, by checking packet sequence number of several consecutive packets. Let's consider W-CDMA systems, in which TCP segments and IP packets are passed from transport, network layers to Radio Link Control (RLC) layer. These packets may be divided into several RLC frames, before passing to MAC layer. The receiver's MAC layer can detect a corrupted RLC frame, but not a missing RLC frame. The receiver's RLC can detect the missing RLC frame if a missing number is found after monitoring several RLC frames of the same TCP segment [54]. However, if the whole TCP segment does not arrive, then the RLC layer cannot be aware of that. Only TCP layer, by checking the byte sequence number, can detect the missing TCP segment. If UDP is used instead of TCP, then only the upper layer RTP can know about a missing RTP packet by monitoring packet sequence number. Therefore, checking and recovery missing packets may incur larger delay, but on the other hand, can bring in better results.

Once a missing packet has been detected, requests for retransmission may be sent automatically or optionally. TCP has a built-in mechanism to request and retransmit lost packets after a timeout or after receiving a triple ACKs. However, applications using UDP and RTP have to handle losses by themselves. At application level, receiver can also inform sender which data frames are correctly received, or notify the encoder to re-initialize the prediction loop if reference frames are lost, for example, by using NewPred in MPEG-4 version 2 or Reference Picture Selection in H.263 version 2 [85, 106, 113, 309]. On the other hand, applications/transport protocols can inform only which data frames/packets are not successfully received (e.g., by NACK) to reduce the feedback traffic, since the number of lost packets is generally smaller than the number of successfully received packets.

Another way is using Selective Acknowledgment (SACK), which is proposed in 1996 as an additional option for TCP. Since within a RTT, a TCP sender can only learn about one lost packet, its performance may decrease when multiple packets are lost within that time. “An aggressive sender could choose to retransmit packets early, but such retransmitted segments may have already been successfully received” [195]. The SACK option allows a receiver to inform the sender about all TCP segments (even non-contiguous) have been successfully received, so the sender only has to retransmit the segments that have actually been lost. This mechanism has been implemented in various operating systems like Windows 98, Linux 2.x and later, Sun Solaris [48].

Common concerns: One concern with retransmission is that it may not be scalable for applications involving a large number of participants like multicast, broadcast, due to a possible acknowledgement implosion. However, feedback should exist to ensure a service that is reliable, and more importantly adaptable to different requirements, capabilities of various receivers and to time-varied conditions of networks. Therefore, it would be beneficial by retransmitting packets selectively or limiting the number of retransmissions for each packet [331]. If SACK or NACK is utilized, the number of feedback data can be substantially reduced, and so will the possibility of an acknowledgement implosion.

Another concern is that a reliable back channel should exist, preferably with short RTT. Otherwise, they may not be suitable for real-time or interactive applications whose one-way delay should be, on average, less than 200ms. However, stringent delay is not required for common applications like news broadcasting or video streaming, which could tolerate a one-way delay up to 10 seconds (see Figure 1.4). Moreover, substantial investments have been put into upgrading network infrastructure, and RTT values of common networks are becoming relatively small. For example, 3G wireless networks like W-CDMA typically have a RTT value less than 100ms [92], while Internet has an average RTT around 134–160ms and a maximum RTT normally less than 200ms [74, 241]. For networks with such RTT, link-layer retransmission is recommended if a moderate delay, e.g., 1 second is allowed. If a longer delay (say 2–3 seconds) is acceptable, application-layer retransmission could be considered [92].

Note that since low layers could have faster response towards network changes than higher layers, retransmission delay at low layers is much smaller than that at higher layer. For example, in W-CDMA systems, MAC retransmission is performed very fast with 2ms delay [54] compared to the ARQ scheme (Radio Link Protocol or RLP) at logical link control layer, whose delay is around 80–100ms [55]. In turn, RLP delay is much smaller than transport-layer retransmission's delay (e.g., by TCP), which is smaller than application-layer's delay. It has been shown that MAC layer retransmission could significantly improve the TCP performance over 3G CDMA networks and various researches have been studied in this direction [55, 61, 89, 177, 178, 230]. However, it also should be noted that a certain layer could only detect and recover a certain type of loss (corrupted packet or missing packet). Therefore, it is always a necessity to use several layers, as well as different approaches, to cope with all possible types of loss.

1.3.2.5 Prevention vs. Recovery

There is a tendency in various papers to compare the performance of retransmission methods like ARQ versus that of FEC-based methods. The pros and cons of both approach have been discussed in Section 1.3.2.3 and Section 1.3.2.4, and are summarized in Figure 1.5.

In our view, the most important point is flexibility toward loss. Effectiveness and efficiency are in fact dependent on the flexibility and adaptability to operating conditions (e.g., applications requirement, network conditions), which in turn depend on whether a feedback channel is employed. Meanwhile, feedback channel exists in most networks and could be efficiently employed in numerous applications, even

	FEC	Retransmission
Feedback	- No feedback channel is required. - However it would be better if feedbacks about network conditions (loss ratio, bandwidth) are considered.	- A feedback channel is needed so that receivers can notice senders about lost packets.
Delay	- Delay in channel coding process, depending on the number of protected packets (*k*). - Receivers wait until sufficient packets have been received (say *k* out of *n*).	- No delay in channel coding process. - Receivers wait about one round trip time to receive lost packets.
Flexibility towards loss	- Less flexible. - Designed for a specific loss ratio.	- More flexible. - Adapt with network conditions.
Effectiveness	- Ineffective if the designed loss ratio is less than actual network loss ratio.	- Only send what is lost.
Efficiency	- Inefficient if the designed loss ratio is larger than actual network loss ratio.	
Complexity	- More complex. - Requires more computational and electric power at senders and receivers.	- Simple (just duplicate and send).
Suitable for...	- Applications require real-time or small delay, e.g., interactive applications. - Network without feedback channel. - Transmission link with large transmission delay, e.g., satellite.	- Applications can tolerate larger delay (> 1-2s), e.g., multimedia streaming. - Network with feedback channel. - Transmission link with fluctuated packet loss ratio.
Unsuitable for...	- Devices with limited computational ability and power, e.g., mobile phone, sensors.	

Figure 1.5: Comparison between FEC and Retransmission approaches.

for multicast communication via satellites [154]. Delay has been constantly reduced because of significant investments in network infrastructure, e.g., about one trillion US dollars to lay fiber-optic cables in the latter half of 1990s [103]. Similarly, rapid and substantial improvements on hardware technologies may quickly overcome the limitations in power, memory ability of devices, and the problem of computational complexity.

It is widely agreed that retransmission is more flexible than FEC since it can automatically adapt to network characteristics [326], thus more robust, bandwidth-efficient and reliable [154]. In exchange, a longer delay is normally required, mainly due to feedback's delay. FEC is more advantageous since it does not have to wait for feedback, but its efficiency would be greatly enhanced if it could receive information about network conditions.

These observations are also valid when we compare prevention approach with recovery approach in general. Prevention is often more complex, expensive and only effective if the real operating conditions are within designed limits. Recovery is normally simpler, cheaper, more adaptive to changing conditions and thus more reliable. On the other hand, prevention is *pre-event* operation while recovery is *post-event* operation, thus prevention always has a delay advantage over the latter. Therefore, each approach will obviously be more useful than the other in some particular conditions. However, it would always be better to incorporate both approaches in hybrid manner rather than focusing on any single approach.

1.3.3 Decoding-based methods

While transmission-based methods are designed to prevent and recover packet loss, decoding-based methods mainly aim to limit the loss's effects during decoding process at receiver. Particularly, when a decoder realizes the existence of data loss, it has several choices: (i) asking for retransmission of the loss data, (ii) concealing the effects of loss by itself (loss concealment). The first option is more closely related to transmission process and has been reviewed in Section 1.3.2.4. The last option is closer to decoding process and will be briefly described in this section. More comprehensive reviews on these methods for audio, video transmission could be found in [224, 244, 298, 305].

Note that in talking about concealment, the commonly used term is "error concealment" to cover a wide range of data corruption or loss, which could be as small as a pixel or as large as several data frames. As mentioned in Section 1.2, we will restrict ourselves to talk mostly about the errors created by packet loss, which could affect one to several pixels blocks or frames. To differentiate, the term "loss concealment" is used instead of "error concealment". In general, there are three common ways to conceal and reduce the loss effects: (i) ignoring the loss, (ii) replacing the lost data by approximated data, and (iii) concealing the effects using inherent characteristics of source signal.

Ignoring the loss means that the decoder may simply skip the lost data to display the next successfully received data. For video, if the frame rate is sufficiently high, e.g., 20-30fps, dropping a lost frame may not create noticeable effects. However for audio, dropping (deleting) the lost segment then contiguously playing its preceding and succeeding segments (splice method) is not that effective [224].

A more popular approach is to approximate the lost data from what have been received. The simplest way is replacing it by the nearest data, temporally and/or spatially. For audio and video, a missing frame could be replaced by the latest successfully received frame. This repetition method works well with audio [224] but may annoy some video viewers since the video is freezed. For video, a lost pixel block could be replaced by the nearest block within its frame, or the corresponding block from previous frame. The position of "corresponding" block within previous frame could be exactly the same as the position of the lost block within this frame, or could be calculated from motion vector [233]. If the lost pixel block is small, it could be approximated by interpolating from the surrounding pixels.

Besides, some approaches could employ inherent redundancies of video data such as edge orientations, geometric structures or foveation feature of the human visual system [164, 324, 325]. Some reviews and techniques could be found in [80, 233, 298, 305].

1.4 Motivations, problems and thesis organizations

In Section 1.3, we have reviewed and discussed popular encoding-based, transmission-based and decoding-based approaches to prevent the problem of packet loss and reduce its negative effects. The essential goal of all these methods is to provide a good perception quality, which could be either objectively approximated by video frame rate, MSE, SNR, PSNR, SSIM [307], etc. or subjectively measured by MOS [131,136, 143]. Due to the non-linear characteristics of human visual and auditory systems, a good objective result of MSE, PSNR may not indicate a good perception quality. This problem does not exist in subjective measurements, which let a number of users rate the perceived quality.

Since *users* would be the ultimate judges of any system, it is essential to know and deliver what are important to users, not to the system. Knowing which content are more important for users (their priorities) would certainly help the system to know what to give higher protection during encoding process, where to allocate more resources during transmission process, and where to focus concealment efforts. For example, by tracking viewers eye movements to determine their regions of interest, we could improve the perceived quality if "perceptually relevant regions are played at higher frame rate than the surrounding area" [118, 254]. Similarly, in some applications such as video surveillance, video conferencing, telemedicine, certain regions within the frames are more important to users. Consequently, these relevant regions should be given high priority.

However in most works, priority is assigned to packets based on the syntax data they carried, or their importance to the *decoder*. For instance, priority may be assigned based on frame type. Within each frame, packets are assumed to have the same quality contribution, thus the same priority. Obviously, such syntax-based prioritization may not reflect users' content-based needs. This is one of the main motivations for this thesis.

Another motivation comes from the debate between network-centric and end-to-end approaches. Delivery prioritized packets over networks is often mistakenly associated with QoS architectures such as Differentiated Services, Integrated Services, in which packets with different priorities are treated differently by network devices. However, such QoS support is not a prerequisite, since senders can simply protect their high-priority packets with more FEC/duplications, or schedule those packets adaptively to network conditions (e.g., congestion) in order to increase their receiving probability. It may even ineffective to let networks, instead of users and applications, decide what to do, since the latter normally know best what their requirements are.

In this thesis, we focus on the problems of prioritizing and delivering packets in multimedia streaming. By differentiating packets based on their quality contributions – either implicitly prioritizing packets based on their coding interdependencies or explicitly prioritizing packets based on their semantic contents – senders can decide what to protect, what and when to send. Specifically, we address the following main questions:

- A sender wants to send a set of packets to a receiver via multiple paths. Would it be better if packets are differentiated, for example, by encoding original data using LC instead of MDC? Given the dependencies between LC packets, the bandwidth constraint and the average network loss rate, how to decide which packet should be sent over which path, with possibility of retransmission, to maximize the expected quality?

- Given a video in which users' regions of interest are defined, how to prioritize packets based on such semantic contents? Will content-based prioritization offer better perceptual results than traditional frame-based prioritization? If so, by how much? How to measure such improvement? Should we consider syntax data, such as sequence header, picture headers, in content-based prioritization? If so, which syntax data should be used?

  If FEC is used instead of retransmission, how to classify and select which packets to protect? How to optimize the level of FEC protection? Will content-based FEC also provide better quality than frame-based FEC?

- Given a set of prioritized packets, how should we send them over a limited bandwidth and lossy link? Which packet characteristic should be used? Should we send the highest-priority packets first, or the earliest-deadline packet first? Or should we consider both priority and deadline? Will including RTT in making schedule decision offer better results? What is the best way to schedule packets?

The organization of this thesis is as follows. We start by briefly describing a general multimedia streaming and the effects of packet loss on streaming quality. Then, we spend most of Chapter 1 to review and discuss popular approaches to minimize the effects of packet loss, based on common network characteristics and user requirements. By conducting a comprehensive review from users' perspective, we find that some common assumptions about the behavior of networks, users' requirements, retransmission constraints may be vague and not very updated. For example, user requirements are not very stringent as they are traditionally believed, e.g., conversational video can stand a one-way delay up to 400ms, while streaming audio and video can tolerate a one-way delay up to 10s [2, 3, 134, 137]. Similarly, although network behaviors are unpredictable, research shows that their constancy could be safely assumed in scale of minutes [332]. Meanwhile, a common RTT value on the current Internet is normally around 134–160ms, or at most 200ms [74, 241]. For such RTT values, link-layer retransmission could be used if required delay is around 1s, and application-layer retransmission could be used if the required delay is 2–3s [92]. Such observations motivate us to investigate and compare FEC-based and retransmission-based delivery methods in better light, as well as lay the foundation for subsequent chapters.

In Chapter 2, we investigate the path diversity approach, in which a sender sends layered coding packets over different paths to a receiver without retransmission. In this scenario, packets are implicitly prioritized based on their inter-dependencies.

That is, no priority is assigned to packets but they are treated differently based on their roles in coding process.

We start with a general optimization framework to optimize the packet allocation over multiple paths. We apply our framework to layered coding (LC) data, whose packets have the same size and the relationship between them could be represented by a simple model. Using dynamic programming technique, we design a polynomial algorithm to find the best allocation scheme, with the assumption that each packet could be protected by a number of duplications, which would be decided before transmission. We find that with a good allocation scheme, LC could achieve much better quality than MDC, especially when bandwidth is limited or there is a high bandwidth disparity between paths. This conclusion clears the common belief that MDC is better than LC in multimedia streaming. Essentially, it means that by treating packets differently based on their quality contributions, we can obtain better results than by equalizing all packets.

In Chapter 3, we focus on how to prioritize packets. While in Chapter 2, packets are implicitly prioritized based on their coding dependencies (i.e., based on syntax like many other works), Chapter 3 departs from that conventional approach, and shows that prioritizing packets based on semantic *regions of interest* within frames can provide dramatic improvements.

Using blob tracking in video surveillance to track moving pedestrians, we can assume that such blobs are the natural regions of interest for users. We then prioritized each macroblock within each frame based on its direct relationship to the blobs, and its coding dependency with other macroblocks. Slices, packets are created and prioritized based on the semantic content and optionally the syntax data they carry. We find that content-based prioritization can provide much better perceptual quality than frame-based prioritization, for example, about 6–11dB improvement in quality of the tracked object. Furthermore, an over-protection prioritizing scheme, e.g., considering many syntax data while prioritizing, may even reduce output quality. We also propose a simple metric (F-PSNR) to measure the quality of blobs, and find that it can strongly indicate the quality perceived by actual viewers.

We then consider another option to protect packets: Reed-Solomon Forward Error Correction (RS FEC) instead of retransmission. Determining which and how many original packets are protected, by how many RS-coded packets is non-trivial. As in the case of prioritization, packets are usually protected at the frame level, e.g., based on the type of frames to which they belong. However, we propose an content-based FEC scheme, in which packets are classified, selected and protected based on their content. Our experiments show that the quality of tracked objects obtained by the content-based FEC scheme could be 10–17% higher than that of the frame-based FEC when videos are transmitted at their average data rate. Under severer bandwidth constraint, content-based FEC could achieve an improvement up to 36% compared to frame-based FEC.

In Chapter 4, we go back to the transmission stage, and focus on packet scheduling process. Though for prioritized packets, sending them in the order of their priority seems to be a natural way, we want to study which and how information about packets and network should be used in scheduling prioritized packets. For example, what would happen if we first schedule packets based on its deadline, instead of priority? Would it be even better if additional information about network, such as RTT, is used in making schedule decisions? Arguing that using either packet's priority or deadline is not enough, we consider both priority and deadline in making scheduling decision. By investigating the performance of different scheduling algorithms, we find that how deadline and priority are used has a great effect on the scheduler's sensitivity to RTT and loss rate, thus the received quality. Particularly, sending highest-priority first provides a relatively stable output quality. Therefore, this scheduling policy works best under bad network conditions, e.g., high loss rates, low bandwidth. On the other hand, sending the earliest-deadline packet first is better in good conditions. Meanwhile, considering the highest-priority and earliest-deadline packet within a set of high-priority packets usually provides good performance in most situations. Surprisingly, although RTT is expected to have substantial affects on packet scheduling, we find that in our content-based video streaming scenarios, the output quality is not significantly changed, with or without RTT consideration.

Finally, Chapter 5 concludes the thesis with a summary of our results and provides some suggestions for future developments.

Chapter 2

Packet allocation over multiple paths

Not all bits have equal value.

—Carl Sagan

2.1 Introduction

Chapter 1 has explained how packet loss could severely affects the quality of multimedia streaming over error-prone networks and discussed various counter approaches. One way is to transmit packets over different network paths (path diversity), so that the variability of packet loss and delay are reduced. In this chapter, we argue that the decision of which packets to send over which paths can greatly affect quality of the received streams.

Of course, if all packets are equally important and all paths are similar, then such decisions are not important. However, when network paths are different (for example, in loss behavior, bandwidth), it would be beneficial to send packets over the better paths. Furthermore, if packets are not equally important, the effects could be even more significant. Given the commonly observed disparity in network paths' characteristics and in packets' priority, therefore, it is important to find a good way to allocate packets over the paths.

To illustrate how a good packet allocation could help to improve the received quality, we compare the performance difference between streaming Multiple Description Coding (MDC) and streaming Layered Coding (LC) data over multiple paths. The idea of MDC is to represent an original data stream with a set of k equally-important packets, so that the original stream can be approximately recovered from any subset of packets. The more packets received, the better the quality of the stream [114, 231, 232]. Meanwhile, LC (or multi-resolution, scalable, embedded, progressing coding) like JPEG-2000 and MPEG-4 partitions the video source data into a base layer and a few enhancement layers with different priorities. The base layer contains the most important video data and decoding only this layer can provide an acceptable perception quality. The enhancement layers deliver complementary information to combine with the base layer for offering higher-quality video output. Since packets from different LC layers are not equally important, they could be benefitted from a good allocation scheme. Furthermore, MDC could be obtained by adding redundancies to an existing LC. Therefore intuitively, working directly on LC may be more effective than using MDC.

Traditionally, a single routing path between a sender and a receiver is used for point-to-point, real-time video and audio communication over the Internet. The quality of service of the communication, therefore is subjected to the properties of the path. Events such as bursty loss and occasional congestion can have negative effects on the quality of the communication. A new model for communication, called *packet path diversity* has been proposed recently by Apostolopoulos [12]. This model proposes using multiple paths between the sender and the receiver for data transmission. By routing data through multiple disjoint paths, we can achieve an "average" channel for communication with reduced fluctuations in loss rate and delay, as the probability that losses or congestions occur simultaneously in all paths is smaller.

Three major issues have to be solved in path diversity implementation: (i) how to select the disjoint paths? (ii) how to enforce the packets to travel through the selected paths? and (iii) how to allocate the packets among the paths? The path selection problem has been extensively studied due to its relevance in telephony and wireless network [261]. Specifying the path for the packets to travel can be done either at the application-level using overlay routing [10, 185, 247], multiple path routing [11, 18, 19, 68, 197] or at the network-level using IPv6 loose source routing.

Methods for allocating packets have been proposed in the literature, mostly based on Multiple Descriptive Coding (MDC) [12,151,175]. In his work [12], Apostolopoulos proposed sending two independently decodable streams, consisting of even and odd frames respectively, over two different paths. Liang, Steinback and Girod proposed similar system for voice communication [175] by using encoding schemes proposed by Jiang and Ortega [151]. These earlier schemes did not consider network conditions. Liang et al. later proposed a scheme that chooses the path to send the next packet based on last packet ACK feedback, which was further developed by Chakareski and Girod [49]. However, a reliable back channel with sufficient short round-trip-delay (RTT) is not always available, and hence may not be suitable for real-time communications. Moreover, back channel is also not applicable in broadcasting or multicast video applications.

Several researches on performance comparisons between MDC and LC over multiple paths have been published [170, 263, 302], and there is a common belief that LC is worse than MDC when the application requires short delay but networks has long RTT or no feedback channel is available. Another conclusion is that MDC is better than LC at high packet loss rate. These conclusions are drawn based on current LC-packet allocation methods, in which ACK feedback is always required and/or protecting base layer will lead to significant delay. Our work is based on the following observations:

- MDC incurs high bandwidth and CPU overhead, and may not be suitable for all situations, such as streaming to low-power devices (e.g., PDA, mobile phone). For the same original data, LC normally requires less bits to encode than MDC does. Therefore, under the same bandwidth constraint, using LC allows more original data to be sent than using MDC.

- Distributing packets encoded with MDC over multiple paths is easy – since all packets are equally important, we can send *any packets along any of the chosen path*. On the other hand, the application may not be able to exploit other information (e.g., network characteristics, coding structures) to further improve output quality. Meanwhile for LC, the priority difference between layers allows us to *choose which and how packets are sent*. This suggests that performance improvement can be achieved if better allocation algorithm is used. Furthermore, for certain MDC that is obtained by adding redundancies in an existing LC, allocating the MDC packets equally over multiple paths is just a special case of allocating the LC packets. In such cases, working directly on the underlying LC could yield higher performance.

- Packet ACK is not a prerequisite in packet distribution. If senders can cleverly decide in advance how to send packets based on a limited knowledge of the network conditions and does not have to wait for acknowledgements from receivers, the problem of delay disadvantage no longer exists. Moreover, if senders can find ways to assign the important level of packets and send them based on their priority without causing any delay or network modification, such solutions would be totally applicable.

The goal of this chapter is to illustrate that with good allocation algorithm to allocate packets among the paths, LC can give better performance than MDC. Our claim, which is in contrast to the common belief that MDC is better, is supported by experiments on ns-2 with data coded by well-known MDC and LC methods.

The rest of this chapter is organized as follows. In Section 2.2, we present the general model for packet allocation and optimization problem [210], as well as a dependency model for layered coding data and its allocation algorithm. The result of experimental comparisons between MDC and LC are shown in Section 2.3. Finally, we conclude in Section 2.4.

2.2 Framework and formulation

In this section, we describe how we model and compute the allocation of LC packets. To send a sequences of MDC packets over multiple paths is quite straightforward, because MDC packets are supposed to be equally important.

2.2.1 General optimization framework

We present the generalized mathematical model for maximizing the gain (thus minimizing expected distortion). In this model, the media data are divided into *chunks*. A chunk consists of a set of packets, with some interdependencies between them. The interdependencies could be due to layered coding (e.g., between base and enhancement layer) or due to motion estimation (e.g., between I frames and P frames in MPEG). There are no dependencies among chunks. A chunk is also assumed to be of reasonable length in time (for example, less than acceptable buffering delay). For simplicity, each packet is assumed to be of the same size. Our results can be easily generalized into a model in which packets have unequal sizes.

A chunk is modeled as a graph $G = (V, E)$ where V is the set of packets, and E represents the dependencies between packets – there is an edge (u, v) from packet u to packet v if u needs to be received for v to be decoded.

Additional FEC packets may be sent to protect packets against bit errors and packet drops. Each FEC packet may protect some number of data packets, with the possibility that a data packet is protected by more than one FEC packets. For practical reasons, we can assume that an FEC packet cannot protect packets that belong to more than one chunks. We model each FEC packet as a subset of V. The set of FEC packets F to protect V is therefore a subset of the powerset 2^V. We also define d to be the *tolerable overhead*. The total number of packets sent (including FEC packets and duplication packets) must not be greater than $(1 + d)$ times the number of packets in V.

We consider a network model without feedback channels, in which a single receiver is connected with a single sender by a set of M disjoint paths $P = \{P_1, P_2, .., P_M\}$ between them. Each path P_m is associated with a bandwidth capacity, and an average packet loss rate, which are denoted as B_m and p_m respectively. Since the time of a data chunk is usually small, we assume that the network conditions remain constant over the time period of a chunk (see Section 1.3.2.1, on the constancy of Internet path) , thus, a single probability value p_m is sufficient to model the loss behavior of path P_m. The unit of B_m is taken to be the number of packets that can be sent over the time period of a data chunk. Note that B_m is *the bandwidth capacity allocated for the data*, and could be constrained by a combination of link capacity, effective TCP-friendly bandwidth and tolerable overhead.

Define an *allocation* to be a function $N : V \cup F \times P \rightarrow \mathbb{Z}^*$. The number $N_{u,m}$ indicates how many times a packet u is sent onto path P_m. A *gain function* of a graph G is a function $g_G : 2^V \rightarrow \mathbb{R}$. $g_G(W)$ measures the gain when exactly a subset of packets $W \subseteq V$ is received or recovered by the receiver. The expected gain of a particular allocation N for graph G can be calculated as:

$$E(g) = \sum_{W \subseteq V} g_G(W) * \gamma(W, F, N) \tag{2.1}$$

where $\gamma(W, F, N)$ is the probability that the receiver receives or recovers all packets in W given a particular allocation N and FEC protection scheme F.

The goal now is to maximize $E(g)$ over all possible N and F, subjected to the bandwidth constraint in Equation 2.2.

$$\sum_{u \in V \cup F} N_{u,m} \leq B_m \text{ for all } m = 1..M \tag{2.2}$$

and the overhead constraint in Equation 2.3.

$$\sum_{m:P_m \in P} \sum_{u \in V \cup F} N_{u,m} \leq |V|(1 + d) \tag{2.3}$$

The optimization problem is infeasible to solve in general due to the large search space. However, we can consider specific dependency graphs that model common media encoding formats to simplify the problem. For example, a group of pictures in MPEG-4 with Fine Granular Scalability could be modelled by a chain of K layers $L_i, i = 0..K$, in which the base layer L_0 is coding-independent, while every other layer L_i is decodable only when all layers L_0, L_1,..., L_{i-1} are successfully received. Considering FEC protection is also difficult, since we have to decide, for example, which set of layers is to be protected by which FEC packet, how to allocate FEC protection over different sets (see Section 3.3). In the next section, we focus on a simple dependency graph called Pairs model (see Section 2.2.2) and do not consider FEC packets.

The network model could be simplified by considering only two disjoint paths (see Figure 2.1), which has been shown to be sufficient for significant improvements in the quality [12]. It can be shown that in the case of two paths, the optimal packet allocation always allocates as many packets as possible onto the more reliable path. Without loss of generality, we assume $p_1 \leq p_2$. Thus, the bandwidth B_1 will always be exhausted.

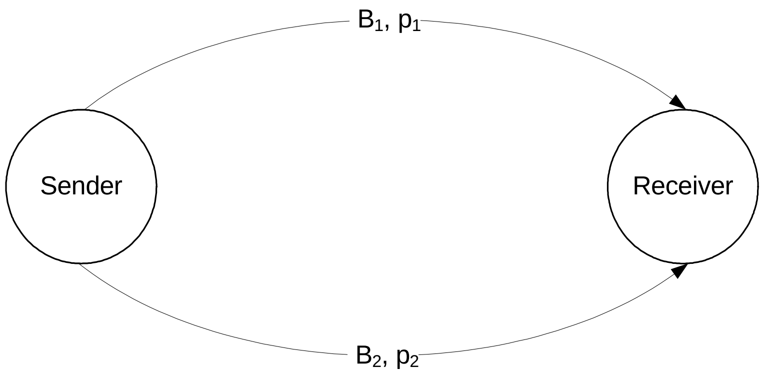


Figure 2.1: Network model.

Without FEC protection, the probability of successfully receiving a packet u given an allocation N is given by Equation 2.4.

$$\gamma(u, \phi, N) \;\; = \;\; 1 - p_1^{N_{u,1}} p_2^{N_{u,2}} \tag{2.4}$$

For brevity, we will use the notation γ_u to denote $\gamma(u, \phi, N)$ when the context of N is clear.

Note that our framework finds the optimal packet *allocation*, not packet transmission *schedule* over multiple paths. In other words, it tells us which packets to send on which paths, but not *when* to send a particular packet. In this chapter, we assumed that after allocation, all packets are sorted based on their captured time and then transmitted over the paths in a round-robin manner.

2.2.2 Optimal allocation for layered coding data

Our dependency graph for layered coding data (Pairs model) is shown in Figure 2.2. In this model, a chunk consists of K packet pairs. Data are divided into two layers. We label the packets in the base layer as L_i and packets in the enhancement layer as H_i, where $i = 1, 2, ...K$.

We define the gain function for each pair of packets as follows. If neither packets are received, or only the enhancement packet is received, the gain is 0. If only the

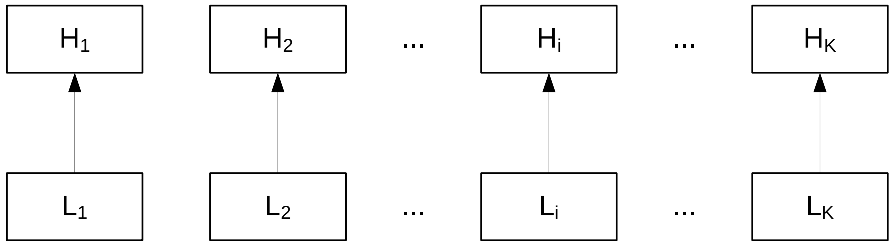


Figure 2.2: Packet pairs model.

base packet is received, then the gain is Δ. If both the base packet and enhancement packet are received, we let the gain value to be 1.

The expected gain $E(g)$ at the receiver can be expressed in Equation 2.5.

$$E(g) = \sum_{i=1}^{K} (\Delta \gamma_{L_i}(1 - \gamma_{H_i}) + \gamma_{L_i}\gamma_{H_i}) \tag{2.5}$$

We now briefly describe a dynamic programming algorithm for finding optimal allocation N that maximizes $E(g)$ subjected to the bandwidth constraints B_1 and B_2. The algorithm works by filling up a 3-dimensional table A, where each entry $A_i^{b_1,b_2}$ stores the optimal expected gain for i pairs of packets given bandwidth constraint b_1 (for P_1) and b_2 (for P_2). Hence the table entry $A_K^{B_1,B_2}$ gives us the maximum expected gain we seek.

We keep another 2-dimensional table N_{opt} of size $(B_1 + 1) \times (B_2 + 1)$, where each entry $N_{opt}^{b_1,b_2}$ keeps the maximum expected gain for a *single* pair of packets given bandwidth constraints b_1 (for P_1) and b_2 (for P_2). To initialize each entry in the table, we exhaustively search for all possible allocations. This takes $O(b_1 b_2)$ time for each entry, thus the total running time for initializing the table N_{opt} is $O(B_1^2 B_2^2)$.

By exploiting the recursive nature of Equation 2.5, the table $A_i^{b_1,b_2}$ can be filled as in Equation 2.6.

$$A_i^{b_1,b_2} = \begin{cases} N_{opt}^{b_1,b_2} & \text{if } i = 1 \\ \max_{j,k}(A_{i-1}^{b_1-j,b_2-k} + N_{opt}^{j,k}) & \text{if } i = 2..K \end{cases} \tag{2.6}$$

The algorithm described above only returns the maximum expected gain. To find the best allocation for pair i, we keep the values of j and k that maximize the gain in each table entry. We can obtain the optimal allocation by tracing back the entries for j and k, starting from $A_K^{B_1,B_2}$. This is a common dynamic programming technique and we omit the details here. Interested readers are referred to [72].

The recursive step searches through all possible allocations for pair i such that the sum of expected gain for pair $1, 2, ...i-1$ and pair i is maximum. Therefore, filling in each table entry for $A(i, b_1, b_2)$ takes $O(b_1 b_2)$ time, giving the total running time of this dynamic programming algorithm $O(KB_1^2 B_2^2)$. In general, this algorithm is pseudo-polynomial as it depends on the input parameters B_1 and B_2. In our case, since $B_1 + B_2$ are bounded by $2K(1+d)$ and d is bounded by a constant in practice, the time complexity of our algorithm is polynomial.

An important note about our allocation algorithm is that it is not meant to be run realtime during streaming. Instead, it can be performed off-line, with its results stored in a table. During streaming, table lookups can be performed based on the perceived network conditions to determine the best packet allocations over the paths.

2.3 Experiments and results

2.3.1 Test data

To make comparison with the allocation method proposed by Liang et al. [175], we use their MDC scheme, which is described by Jiang et al. [151]. From each 800 original signal samples (800 bytes), two MDC packets are constructed. For the first packet, the even original samples are quantized in a finer resolution (PCM, 8 bits per sample) and the differences between adjacent odd and even samples are encoded in coarser resolution (ADPCM, 2 bits per sample). Inversely for the second packet, we encode odd samples in fine resolution and the difference between even and odd samples in coarse resolution. Therefore, each MDC packet has the size of 500 bytes, which means a redundancy of 25%, excluding packet headers.

As mentioned in Section 2.2.2, the two LC streams are generated as follows. From each 800 original signal samples (800 bytes), the 4 most significant bits (MSB) are extracted and packetized in a base layer packet. The enhancement layer packet consists of 4 least significant bits of 800 samples. By this way, each packet has a the size of 400 bytes, but the base packet contributes most important information while the enhancement packet is only useful if its corresponding base packet is received.

To find the packet gain, we first compare the quality of each encoded packet with its original 800 samples in terms of Segmented Signal-to-Noise ratio (SSNR). From that, we find the importance-level ratio between two packets of a pair (LC base layer packet and enhancement layer packet, MDC odd and even packets). From the assumption that the gain of a pair is 1, we can calculate the gain of each packet. We found that most LC base layer packets have higher gain than its enhancement layer packets while the gain of each MDC packet is around 0.5 ± 0.01, which is agreeable with their coding nature.

In our experiments, we use two files encoded by PCM at the sampling rate of 8 kHz (mono, 8 bits/sample). The first file, called `f116`, consists of 21600 samples (2.7 seconds), and the second file, named `CuckooWaltz`, consists of 32000 samples (4.0 seconds). Using either MDC or LC schemes, we obtain the same number of packets: 27 pairs of packets for `f116` and 40 pairs of packets for `CuckooWaltz`. Since the size of each MDC packet is 500 bytes, the size of `f116` in MDC format is 27000 bytes, and that of `CuckooWaltz` is 40000 bytes. The total bandwidth required to send each file in MDC format is 10KB/s. Under this bandwidth, we can send 20 MDC packets or 25 LC packets per second.

2.3.2 Packet allocation schemes

We use the network model shown in Section 2.2 in our experiment, as it is used by Wang [302] and Liang [175] in their studies. In these works, the authors use the same method to send MDC streams, which is simply transmitting one description over one path, and the other description on the other path.

Liang assumes both paths having the same bandwidth, therefore for the cases where paths' bandwidths are different, we extend their original scheme to *LiangExt allocation*. Similar to Liang allocation, this scheme sends (sequentially, from left to right) packets of MDC even stream over one path, and those of MDC odd stream over the other path. However, if all packets of a stream have been already sent but its allocated bandwidth has not been used up, packets will be sent again sequentially until all bandwidths are fully used.

The *LiangExt allocation* scheme is reasonable when the bandwidth of each path is higher than the required bandwidth of each stream. However if a path's bandwidth is much lower than the required bandwidth, it is no longer a logical choice. To have a fairer comparison in these cases, we develop a simple *Greedy allocation* scheme, which alternatively sends packets from two MDC streams (the first packets from two streams, then the second packets from two streams, and so on) over each path until all allocated bandwidth are used up.

For LC streams, instead of interleaving the base packets and enhancement packets over two paths as proposed by Wang et al. [302], we employ our scheme described in Section 2.2.2.

2.3.3 Experiment settings and results

To evaluate, we compare the quality of reconstructed files produced by each scheme under the same network conditions (bandwidth, average loss rate). The average loss rate p_1 is kept at 1%, while p_2 varies from 1% to 40% with different increments:

1% increment in the range of [1%, 10%] and 10% increment in the range of [10%, 40%]. The bandwidth of each path also varies from 0% to 80% of the required bandwidth, depending on the experiment scenarios. The network and streaming process are simulated using Network Simulator ns-2. For each network configuration, the experiment is repeated 100 times, and the average value of these runs are reported.

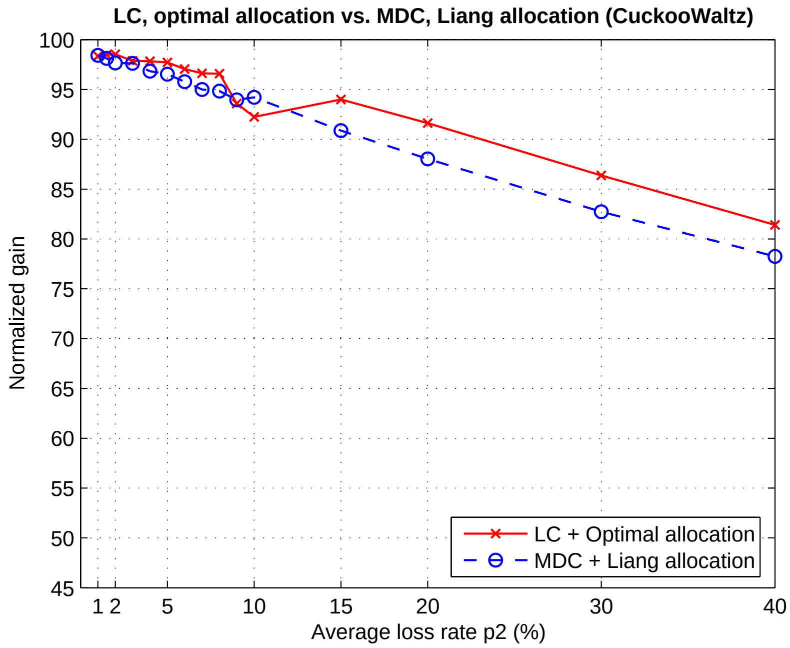
Ideally, the quality of reconstructed files should be measured perceptually in terms of Mean Opinion Score (MOS). This could be achieved by either conducting formal subjective tests with users according to ITU recommendations [132, 142] or using objective methods using psychoacoustic models like Perceptual Evaluation of Speech Quality (PESQ) [135,140,144], Perceptual Evaluation of Audio Quality (PEAQ) [133] in combination with ITU E-model [121, 138, 141]. However, such tests are expensive and time-consuming to conduct, especially with a larger number of files. Therefore, we opt to measure the quality of reconstructed files by a simple metric called *normalized gain*, which is defined in Equation 2.7.

$$\text{Normalized gain} = \frac{\textit{Reconstructed gain}}{K} 100 \tag{2.7}$$

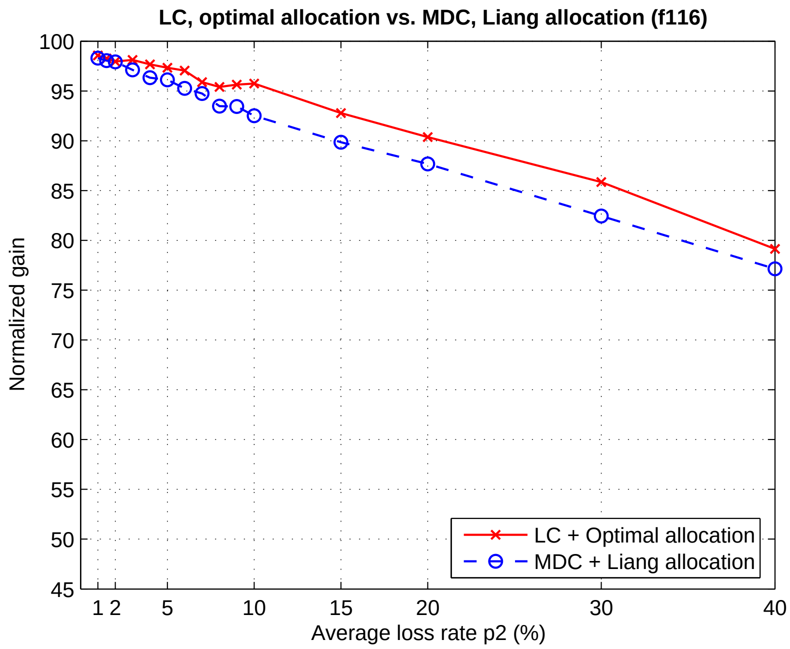
where K is the total gain of the original file (27 for `f116` and 40 for `CuckooWaltz`).

The first experiment compares the quality obtained by sending LC streams using our proposed method (`[LC + Optimal allocation]`) with the quality obtained by sending MDC streams using Liang's method [175] (`[MDC + Liang allocation]`). The bandwidth of each path is equal to 50 percent of the total bandwidth required, which is precisely enough for send one MDC stream over each path. The average loss rate of path 1 is 1% and the average loss rate of path 2 changes from 1% to 40%.

Figure 2.3 shows that the quality produced by `[LC + Optimal allocation]` scheme are better than that of `[MDC + Liang allocation]` scheme in most of the time. Particularly, for `CuckooWaltz`, the average normalized gain obtained from `[LC + Optimal allocation]` scheme is in the range of 81.4–98.5 (standard devi-



(a) CuckooWaltz



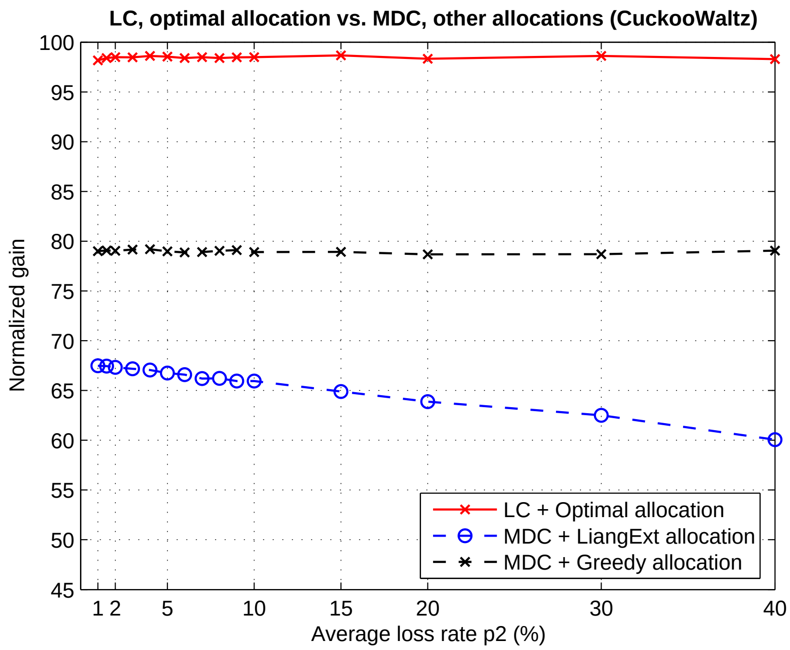
(b) f116

Figure 2.3: LC with optimal allocation vs. MDC with Liang allocation ($B_1 = B_2 =$ 50% of total bandwidth required, $p_1 = 1\%$, p_2 varies): (a) CuckooWaltz, (b) f116.

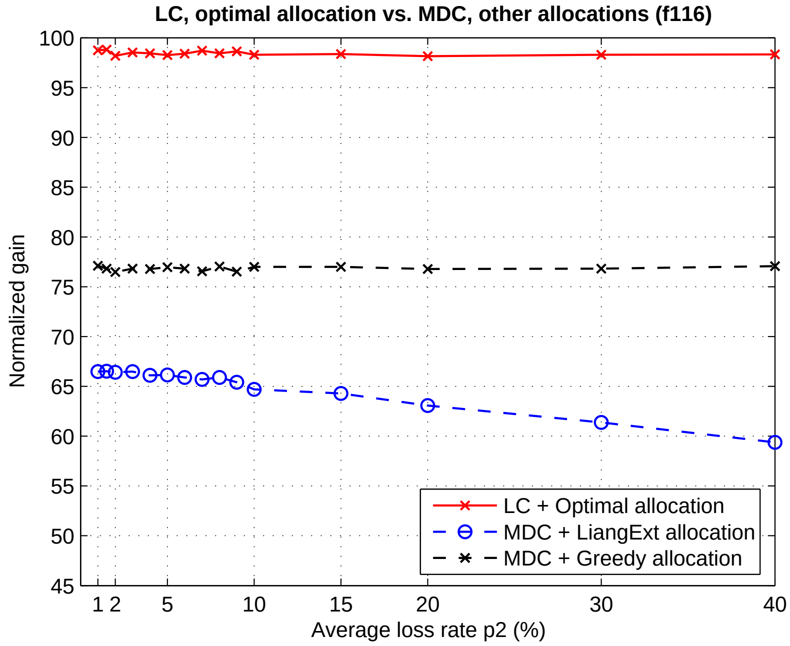
ation σ of 1.8–5.5), while that obtained from `[MDC + Liang allocation]` scheme is 78.4–98.4 (σ of 1.5–4.3). The corresponding numbers for `f116` are 79.2–98.6 (σ of 0.6–1.9) with `[LC + Optimal allocation]` scheme, and 77.2–98.3 (σ of 0.6–1.5) with `[MDC + Liang allocation]` scheme. Because LC packet (400 bytes) is smaller than MDC packet (500 bytes), the first scheme has more room to choose and duplicate important packets while the latter can send each MDC packet only one time. However, the quality difference is not significant in this case, when there is not much difference between the paths (they have the same bandwidth, and only differ in loss rate).

In the second set of experiments, the bandwidths and packet loss rates of two paths are varied. The total bandwidth is still exactly equal to the total bandwidth required, but each path has a different bandwidth: $B_1 = 80\%$, $B_2 = 20\%$. The average loss rates of two paths are the same as in the first set of experiments. In these cases, we compare the quality received from streaming LC packets with the proposed optimal allocation (`[LC + Optimal allocation]`) with the quality obtained from streaming MDC packets with LiangExt allocation (`[MDC + LiangExt allocation]`) and Greedy allocation (`[MDC + Greedy allocation]`) schemes.

Figure 2.4 shows that our proposed scheme `[LC + Optimal allocation]` produces significant higher quality than streaming MDC with other allocation schemes. For `CuckooWaltz`, `[LC + Optimal allocation]` scheme produces an average normalized gain of 98.2–98.7 (standard deviation σ of 1.7–2.2); `[MDC + Greedy allocation]` scheme, 78.7–79.2 (σ of 1.2–1.5); and `[MDC + LiangExt allocation]` scheme, 60.1–65.7 (σ of 0.3–2.6). It means the quality obtained from the first scheme is 24–25% higher than that of the second scheme, and 46–64% higher than that of the last scheme. For `f116`, its average normalized gain is 98.2–98.8 (σ of 1.9–2.5) with `[LC + Optimal allocation]`; 76.5–77.1 (σ of 1.4–1.9) with `[MDC + Greedy allocation]`; and 59.4–66.5 (σ of 0.7–3.7) with `[MDC + LiangExt allocation]` scheme. It means



(a) CuckooWaltz



(b) f116

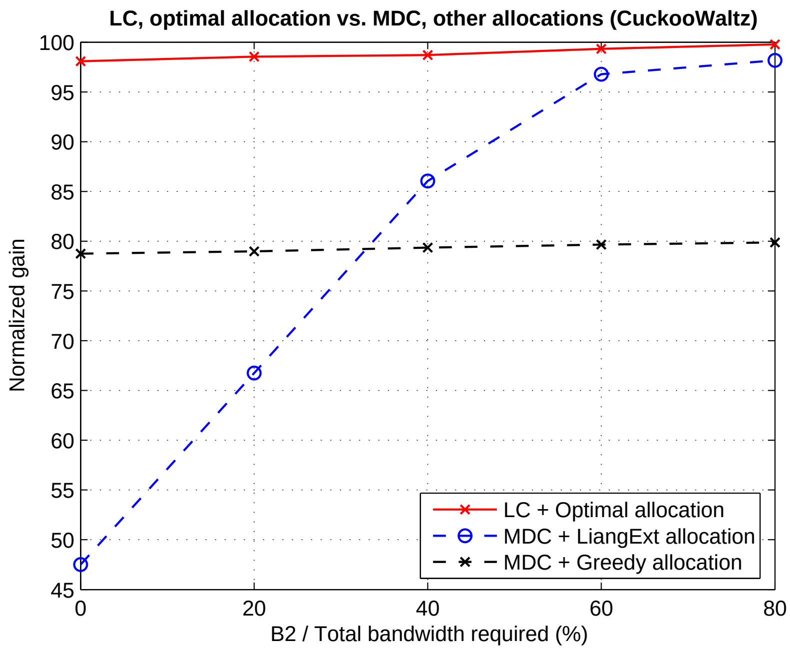
Figure 2.4: LC with optimal allocation vs. MDC with LiangExt and Greedy allocation schemes ($B_1 = 80\%, B_2 = 20\%$ of total bandwidth required, $p_1 = 1\%$, p_2 varies): (a) CuckooWaltz, (b) f116.

the quality obtained from `[LC + Optimal allocation]` scheme is 28–29% higher than that of `[MDC + Greedy allocation]`, and 48–66% higher the quality produced by `[MDC + LiangExt allocation]` scheme.

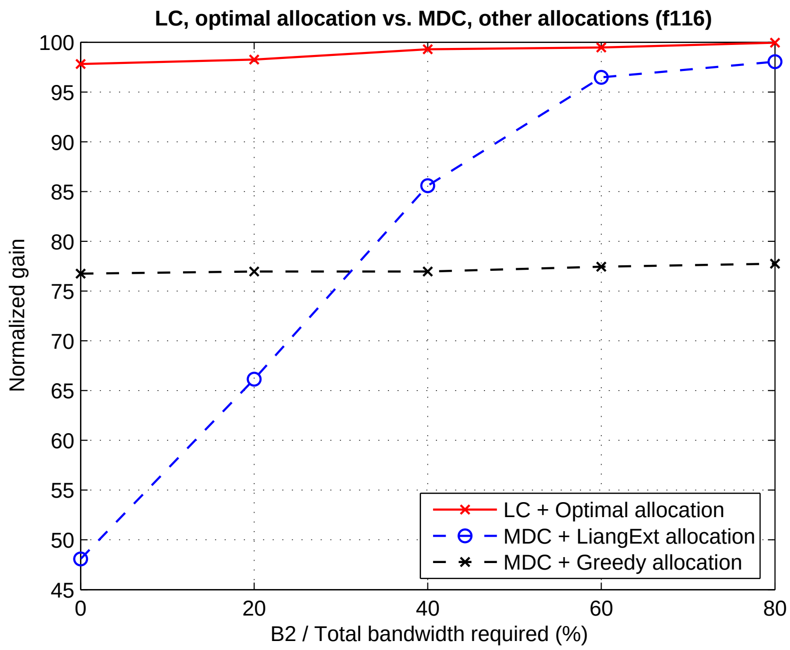
It is interesting to note that under the same total bandwidth and loss rates, the bandwidth difference between two paths has a substantial effect on MDC's and LC's performance. When two paths have the same bandwidth (50% of the total bandwidth required), the qualities received from `[MDC + Liang allocation]` and `[LC + Optimal allocation]` schemes sharply decrease with the increase of p_2 (see Figure 2.3). When the bandwidth B_1 of path 1 (loss rate of 1%) increases to 80% and B_2 decreases to 20% of the total bandwidth required, `[LC + Optimal allocation]` scheme can allocate more important packets to the better path (path 1), thus significantly improves its performance (see Figure 2.4). Meanwhile, the other two schemes cannot effectively use any spare bandwidth in the better path, since for them all MDC packets are equally important.

Figure 2.5 shows the results when the average loss rates of two paths are unchanged ($p_1 = 1\%$ and $p_2 = 5\%$) but the bandwidth of path 2 (B_2) varies from 0% to 80% of the total bandwidth required ($B_1 = 80\%$). While the reconstructed quality in the case of `[MDC + Greedy allocation]` scheme is almost unchanged (around 79.3 ± 0.4 for `CuckooWaltz` and 77.2 ± 0.3 for `f116`), the quality obtained from `[MDC + LiangExt allocation]` scheme is heavily affected by the bandwidth change. When B_2 is lower than or equal to 20% of the total bandwidth required, the performance of `[MDC + LiangExt allocation]` scheme is always lower than that of `[MDC + Greedy allocation]` scheme. However, when B_2 increases to 60–80%, the first scheme outperforms the latter by at least 20%.

Meanwhile, `[LC + Optimal allocation]` scheme provides a stable and significant higher quality than other schemes in all cases (around 98.9 ± 0.7 for `CuckooWaltz` and 99 ± 0.4 for `f116`). Compared to `[MDC + LiangExt allocation]`, it provides



(a) CuckooWaltz



(b) f116

Figure 2.5: LC with optimal allocation vs. MDC with LiangExt, Greedy allocation schemes ($B_1 = 80\%$ of total bandwidth required, B_2 varies, $p_1 = 1\%$, $p_2 = 5\%$): (a) CuckooWaltz, (b) f116.

an average normalized gain of 1.6–50.6 point higher for `CuckooWaltz`, and 1.9–49.7 point higher for `f116`. Even with a bandwidth of only 80% of the total required bandwidth ($B_1 = 80\%$, $B_2 = 0\%$), it still offers a similar or better quality compared to `[MDC + LiangExt allocation]` scheme produces with a total bandwidth two times higher ($B_1 = B_2 = 80\%$).

2.4 Summary

In this chapter, we studied the packet allocations of layered coding media streams over multiple paths. We proposed an analytical framework to solve the following problem: Given a set of disjoint paths without feedback channels, their effective bandwidth, their probability of loss, the dependencies between packets, and the rate-distortion function, which packet (with possibility of duplication) should be sent through which path to maximize the expected received quality? This framework was applied for a simple simple relationship model for layered coding data, in which all layers could be assumed to have the same size. We presented a polynomial algorithm to find the best allocation, with the assumption that each packet could be protected by a number of duplications, which would be decided before transmission.

We found that with a good allocation scheme, layered coding (LC) could achieve much better quality than multiple-description coding (MDC), especially when bandwidth is limited or there is a high bandwidth disparity between paths. These finding cleared the common belief that MDC is better than LC in multimedia streaming. It also illustrated that by treating packets differently (LC) rather than equalizing them (MDC), significantly better quality could be achieved.

Chapter 3

Content-based priority streaming in video surveillance

We don't see things as they are.

We see them as we are.

—Anais Nin

3.1 Overview

In Chapter 2, we have studied how to allocate, protect data layers over multiple paths based on the dependency relationship between layers, in order to maximize the expected quality. Although no priority value is assigned to any layer, the base layer always receives higher allocation and protection than other enhancement layers. In effects, we implicitly give higher priority to the base layer.

Various works also show that prioritizing video packets for streaming over lossy networks can improve quality and effective frame rate at the receiver. Typically, such prioritization is done at frame granularity based on syntax, or their importance to the *decoder*. For instance, priority may be assigned based on frame type. Within each frame, packets are assumed to have the same quality contribution, thus the same priority. However, in some applications, such as video surveillance and video conferencing, certain regions within the frames are more important to the *users*. This chapter will study the effect of prioritizing, protecting and streaming such *regions of interest* within the frames for streaming.

In Section 3.2, we propose a simple and effective scheme to packetize and prioritize packets based on semantic *regions of interest* within frames. Packets are then streamed, with possible retransmission, according to their priority. Using blob tracking in video surveillance as an example application, we demonstrate that compared to frame-based priority streaming scheme, content-based scheme can achieve 6–11dB improvement in quality of the tracked object. Subjective measurements with 19 users also confirm the objective results.

In Section 3.3, we consider another option to protect packets: Reed-Solomon Forward Error Correction (RS FEC) instead of retransmission. Determining which and how many original packets are protected, by how many RS-coded packets is non-trivial. As in the case of prioritization, packets are usually protected at the frame level, e.g., based on the type of frames to which they belong. However, we propose a content-based FEC scheme, in which packets are protected based on their content. Our experiments show that the quality of tracked objects obtained by the content-based FEC scheme could be 10–17% higher than that of the frame-based FEC when videos are transmitted at their average data rate. Under severer bandwidth constraint, content-based FEC could achieve an improvement of up to 36% compared to frame-based FEC.

3.2 Content-based priority streaming

3.2.1 Introduction

Priority video streaming — prioritizing video data and streaming them over lossy networks according to their priorities — is proved to be useful in various applications. Usually, video is prioritized at frame level based on video syntax, such as sequence header, slice header, and frame type. Such syntax-based prioritizing and streaming could help to improve effective frame rate and overall frame quality of received videos [9, 20, 163, 219, 277].

These quality measurements, however, may not be sufficient, and that of importance, to applications such as video surveillance, video conferencing, and telemedicine. Consider a surveillance video, which usually contains many idle and non-event segments interleaved with some short bursts of events. Within each eventful video segment, some frames may be more essential to users than others. In each frame, users may only focus on certain *regions of interest*, such as human, cars, colorful objects, or moving objects, and do not care about the rest. Ultimately, the quality of what users focus on would be their yardstick to judge quality of the whole video. Therefore, it would be better to protect the actual *content* that users are interested in, rather than the important syntactic *data*.

To avoid lengthy expressions, we will use the terms "blob" and "interest region" interchangeably with "region of interest" from this point onwards.

Identifying the regions of interest of a viewer is a non-trivial problem. In the literature, two approaches exist. The first approach explicitly obtains feedback from the viewer (e.g., using eye-tracking devices [118, 162, 254]). The second approach mathematically models human's attention and predicts where a viewer is focusing on [145]. These approaches aim to identify the region of interest for video in general. In certain specific applications, however, region of interest can be easily identified with high accuracy. One such application is video surveillance.

One common operation in video surveillance is blob tracking – highlighting moving objects in a scene and tracking the objects over time. For users monitoring such videos, these blobs naturally become their region of interest. In this section, we investigate the effectiveness of prioritizing these blobs in a distributed video surveillance system.

A distributed video surveillance system consists of multiple network-capable surveillance cameras (Figure 3.1). These surveillance cameras are capable of capturing video and transmitting them over the network to a processing server for analysis. The processing server may optionally archive the video on disk and send analysis results to users. For instance, the processing server may detect motion in a scene and starts tracking the moving object in the video. The moving objects are highlighted and sent to the users. For mission-critical surveillance applications (e.g., military), one would build the video surveillance system over a dedicated or over-provisioned network, to ensure that the surveillance videos received are of high quality. Many other surveillance applications (home monitoring, corporate security), however, is layered over the commodity Internet, which is often lossy and has limited bandwidth.

Given the resource constraint of the underlying network, it is necessary to know which part of a video is more important in order to prioritize and send them accordingly. In our video surveillance application, the blobs in the video being tracked are sent with higher priority from the processing servers to the users. Specifically, their macroblocks are assigned higher priority than non-blob ones.

One arising question is whether we should, and if so, how to combine both syntax-based and semantic-based prioritization. Surprisingly we found that if the blobs in each frame are prioritized, then frame-based prioritization is unnecessary. This finding leads us to a simple prioritizing scheme, in which a packet priority is determined based on (i) whether it contains blob-related macroblocks or not, and (ii) whether it contains

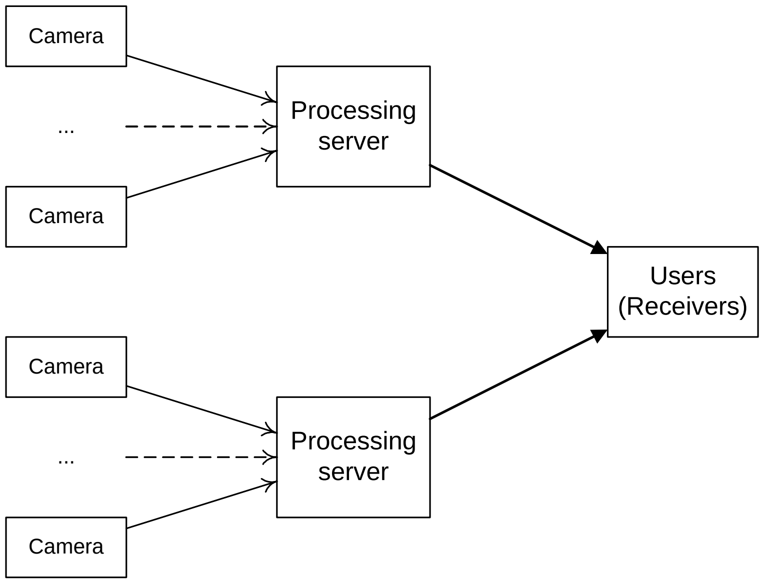


Figure 3.1: A distributed video surveillance system.

sequence or GOP header or not. Note that blob-related macroblock could be either a blob's macroblock (i.e., inside the blob) or a macroblock (from different frame) on which the coding of a blob's macroblock depends.

Our experiment results show that by prioritizing packets based on semantic content rather than based on syntactic information like frame type, the perceptual quality could be significantly enhanced. Objective comparison of blobs' PSNR shows a 6–11dB improvement by our scheme, compared to frame-based priority streaming. Subjective comparison with 19 users, using MSU Perceptual Video Quality tool [204] to measure Mean Opinion Score, confirms the advantages of using content-based prioritization.

The rest of this section is organized as follows. Section 3.2.2 gives an overview of related works. Our content-based prioritizing scheme is presented in Section 3.2.3, and Section 3.2.4 briefly describes how packets are scheduled based on their priority. Section 3.2.5 talks about our prototype's implementation, evaluation metrics, and experiment results. Discussions about the results, as well as some interesting observations are also presented. Finally, we conclude in Section 3.2.6.

3.2.2 Related works

In this section, we will briefly present some related research, and discuss the difference between our work and the existing literature.

Content-based protection could be carried out at different phases: encoding or transmission. Both borrow content-analyzing techniques from computer vision, image processing, and visual information retrieval [5, 194]. Content-based encoding has been studied and applied extensively, e.g., for objects coding in MPEG-4 [53, 294] and Region of Interest (ROI) coding in JPEG 2000 [43]. In content-based transmission, source and channel coding (e.g., FEC [104], interleaving [62]) are usually combined with content's information to determine level of protection [104], to allocate appropriate bandwidth [33, 318] or bit rate [166, 333, 334]. For example, Yang and Nahrstedt [318] allocate more bandwidth to the camera capturing high motion activity, which is calculated by averaging motion activity every second. In this section, we are concerned not about coding, be it source or channel coding, but more about packet's prioritization and delivery.

Note that *content* could mean different things in different works. It could be scene-level [33,52,104], frame-level [62,277], region-level [53,165,294] or packet-level [62,82]. It could referred to either syntactic or semantic information. For example, AMISP scheme [104] focuses on structuring MPEG-2 video by modulating the number of slice headers and intra-coded macroblocks. Then FEC packets are inserted to protect packets whose loss may create high spatial distortion (e.g., those contains sequence/picture/slice headers). In streaming lecture videos [183, 184], semantic content (text) is used to determine which frames or regions are important. Our work focuses on semantic content at region level.

There are various ways to prioritize packets based on content. For example, in the work by Shin et al. [259], packet's priority is calculated from motion vector size and the number of intra-coded macroblocks contained in the packet. In the work by

Shih-Fu Chang et al. [52], sport videos are segmented and prioritized based on what is showing (e.g., pitching shots in baseball, serving shots in tennis) or user's preference. To determine the priority of what is showing, domain knowledge about event structure is used in this case. In other work [277], the first frame in each video shot is given highest priority, and the second highest priority is assigned to representative frames, which are chosen based on motion and color analysis.

Essentially, for every content-based prioritizing scheme, two questions need to be answered: (i) which content is important? and (ii) how to derive packet's priority from that content? The important content could be determined by users or detected automatically by various algorithms [102, 145, 160, 172]. For example, Itti and Koch, based on the feature integration theory on visual attention [288], compute a saliency map for each frame to determine the focus of attention [145]. Komogortsev and Khan use an eye-tracking device to predict the region at which viewer is currently looking, to encode it with higher quality [162]. These approaches are designed for general video applications.

In our work, we focus on video surveillance and we use blob tracking algorithm proposed by Li et al. [172]. Using their algorithm, objects and background are automatically detected, classified (based on Bayes decision rules and general feature vector), segmented and adaptively updated. Therefore, foreground objects could be well detected from complex videos with both stationary and moving background objects. The second question, which is a focus of this section, will be discussed in Section 3.2.3.

3.2.3 Content-based prioritizing scheme

This section describes and discusses our content-based prioritizing scheme for MPEG-encoded video. In particular, we will show how to prioritize packets for an MPEG Group of Picture (GOP) based on information about interest regions within each frame and the dependencies between frames. The GOP is assumed to have one I-frame, followed by a number of P-frames or B-frames.

To find the priority of a packet (or slice, macroblock), two stages are usually required. The first stage, based on visual content, finds its *content priority*. The second stage finds its *effective priority*, based on the content priority and syntactic data/relation. This priority would be used to represent the importance of the packet (slice, or macroblock). The word "priority" refers to "effective priority" when the context is clear. We assume all priorities are integer numbers.

Our content-based priority streaming consists of 4 major steps (see Figure 3.2). Firstly, given the information about interest regions — if exist — within each frame, each macroblock m of frames in a GOP will be assigned a content priority $\mathtt{wc}_m$. Then, based on the coding dependencies among the macroblocks within the GOP, the effective priority w_m of each macroblock is computed. Secondly, consecutive macroblocks are grouped into slices according to their priority w_m. No syntactic information is used in this step, thus the effective priority w_s of a slice s is equal to its content priority $\mathtt{wc}_s$. The third step is packetization, in which slices are sequentially grouped together until the packet size reaches the MTU limit. The effective priority w_u of a packet u is determined based on its slices' priorities and, optionally, based on the syntactic data it carries. Finally, packets are transmitted and retransmitted according to their priority. Details of each step will be described in the following sections.

3.2.3.1 Priority map and effective priority

Assuming that for each frame, the interest regions (blobs) are already defined, either manually or automatically by an algorithm mentioned in Section 3.2.2. This step finds the priority of each macroblock for each frame.

To store the priority of all macroblocks within a frame, we use a *priority map*. It

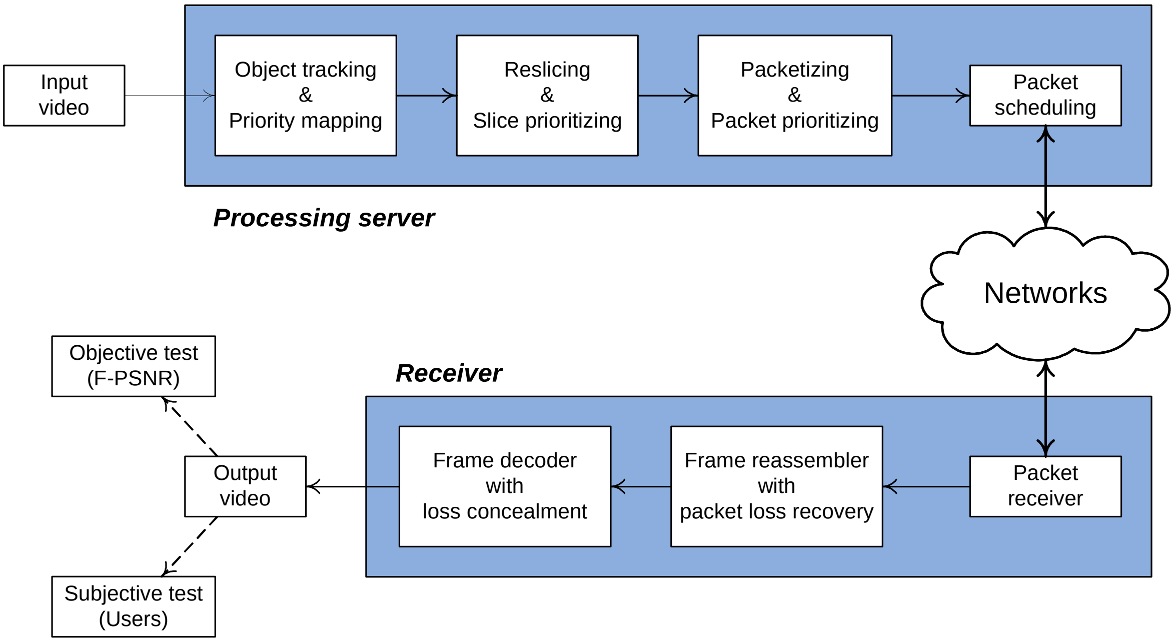


Figure 3.2: The content-based priority streaming prototype.

could be understood as an 2D array, where each element corresponds to a macroblock in the frame. The value of each element is the priority of its corresponding macroblock. Figure 3.3 shows a priority map superimposing on its frame

For each macroblock m within a frame, the first stage is to find its content priority $\mathtt{wc}_m$ based on its relation to the blobs within the frame. There are several ways to define this relationship, e.g., depending on whether it is inside or outside the blobs, how far it is from the blobs. Therefore, the content priority could be assigned in different ways. The simplest way is to assign a high value to macroblocks inside the blobs, and a low value to those outside. A more complex way is assigning the highest value to the firsts, then gradually decreasing value as we go further from the blobs.

In our scheme, for each blob, we define a protected region, which is k percent larger than the actual blob. (For our experiments, we chose k to be 20% based on the size of the blobs in the two test videos. A different value could be chosen depending on the nature of the surveillance video.) A macroblock m that belongs to this protected region will have its content priority $\mathtt{wc}_m = w_{\mathtt{max}}$, where $w_{\mathtt{max}}$ is an integer number

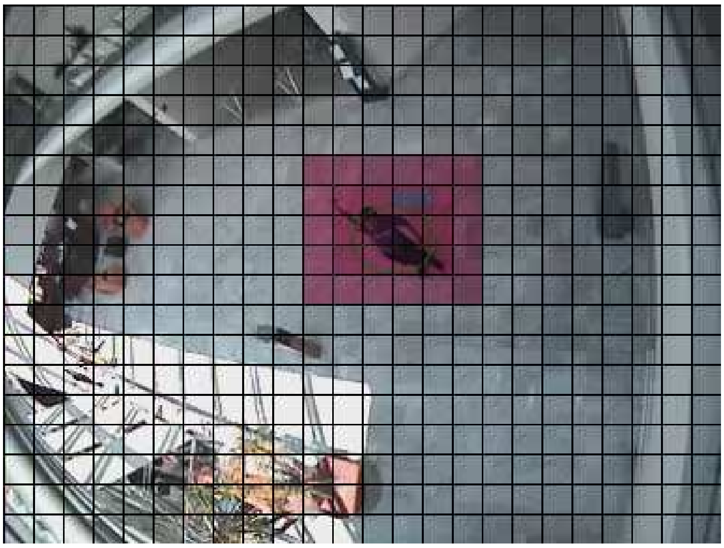


Figure 3.3: Priority map for a video frame.

greater than 1. If m is outside this protected region, we assign its content priority $\mathtt{wc}_m = 1$.

The next stage is to find the *effective priority* w_m of each macroblock m, based on its content priority $\mathtt{wc}_m$ and its encoding relationships with other macroblocks within its GOP. We first present our method using a formal graph model, followed by a description of the computational procedure.

For any MPEG GOP, we can construct a directed, acyclic graph G, whose each vertex is a macroblock. There exists an edge (m_i, m_j) if macroblock m_j is predictively coded based on macroblock m_i. An I-frame macroblock m_j will have no in-coming edge since it is intra-coded. On the other hand, there will be no out-going edge from a B-frame macroblock m_i, since it has no dependent macroblock.

Let $P_m \subseteq G$ be the set of all nodes reachable from m (i.e., dependent on m). Let $\mathtt{wc}_m$ be the content priority of macroblock m, its effective priority w_m is computed recursively as follows:

$$w_m = \begin{cases} max_{m' \in P_m} w_{m'} & \text{if } P_m \neq \{m\} \\ \mathtt{wc}_m & \text{otherwise} \end{cases} \tag{3.1}$$

In practice, it is not necessary to construct a graph for the whole GOP to compute effective priority, since macroblock's motion vector(s) could be used for the same purpose. For each GOP, the whole procedure to find the effective priority for all macroblocks consists of two passes, as follows.

1. In the first pass, starting from the I-frame at the beginning of the GOP, the content priority $\mathtt{wc}_m$ of each macroblock m is assigned, and its motion vectors (if exist) are extracted. Since B-frame macroblocks are not used as reference for any other macroblocks, their effective priority w_m will always equal to its content priority $\mathtt{wc}_m$. Furthermore, according to Equation 3.1, if the macroblock m is coding dependent on the macroblock m', the effective priority $w_{m'}$ will always be greater than or equal to w_m. Therefore, if a B-frame is encountered, the effective priority w_m of each macroblock m in the frame will be propagated (assigned) to all its forward and backward reference macroblocks.

2. In the second pass, we go from the end of the GOP towards the I-frame. If a P-frame is encountered, the effective priority w_m of each macroblock m in this P-frame will be propagated (assigned) to all its forward reference macroblocks.

Note that by this way, the effective priority w_m of a macroblock m will be equal to either 1 or $w_{\mathtt{max}}$.

3.2.3.2 Re-slicing and slice prioritizing

In this step, macroblocks are grouped into slices. Each slice is assigned a priority, which is calculated from its macroblocks' priorities.

Because effective priority w_s of a slice s is dependent on the priority of its macroblocks, grouping macroblocks with different priorities into a slice will make it difficult to determine the slice's priority. Let S be the set of all macroblocks m belong to the slice s. If $w_s = max_{m \in S} w_m$, we will unnecessarily increase the importance of low-priority macroblocks. If $w_s = \sum_{m \in S} w_m$ or $w_s = \lceil \frac{\sum_{m \in S} w_m}{|S|} \rceil$, the importance of high-priority macroblocks will be diluted. Therefore, it would be better to group macroblocks with same priority into a slice.

To achieve that, we scan one pass through macroblocks in each frame, and insert a new slice header whenever (i) the macroblocks' priority changes value, or (ii) when adding one more macroblock will make the slice's size exceed 1400 bytes (so that after adding headers, packet size could be less than Maximum Transmission Unit). Therefore, each slice will contain a number of consecutive and equal-priority macroblocks, and its priority is set equal to the priority of any macroblock within it (either 1 or $w_{\mathtt{max}}$).

3.2.3.3 Packetizing and packet prioritizing

In this part, slices are grouped into network packets. The priority of each packet is calculated based on its slices' priority and the syntactic data contained in the packet.

Our packetization scheme simply follows the recommendations in RFC2250 [122]. MPEG sequence header, GOP header, picture header are recommended to be at the beginning of RTP payload, therefore, a packet will always contain slices from a single frame. For each frame, slices are added into a packet until the packet's size reaches the MTU limit [40]. We try to avoid slice fragmentation as much as possible, so that most of the time each packet contains an integral number of slices.

The priority of a packet u is calculated in two stages. First, its content priority $\mathtt{wc}_u$ is found by averaging its slices' priorities and rounding up to the nearest integer. Since the priority of a macroblock/slice is either 1 or $w_{\mathtt{max}}$, the content priority of a packet lies within the range $[1, w_{\mathtt{max}}]$. Let x, y the number of high-priority slices and low-priority slices in packet u, respectively. The content priority $\mathtt{wc}_u$ is as follows.

$$\begin{aligned} \mathtt{wc}_u &= \lceil \tfrac{w_{\mathtt{max}} \times x + 1 \times y}{x+y} \rceil \\ &= \begin{cases} \lceil w_{\mathtt{max}} - \frac{w_{\mathtt{max}}-1}{1+x/y} \rceil & \text{if } y \neq 0 \\ w_{\mathtt{max}} & \text{otherwise} \end{cases} \end{aligned} \tag{3.2}$$

Thus, a packet consisting of many low-priority slices will always has lower content priority than a packet containing only a few high-priority slices. In other words, the value of $\mathtt{wc}_u$ is substantially affected by the ratio between x and y, but not by the total number of slices $(x + y)$ it contains.

The next stage is to include syntactic data carried by the packet in deciding its priority. Let $\mathtt{ws}_u$ be the *syntactic priority* of packet u. The effective priority w_u of a packet u is the sum of its content priority $\mathtt{wc}_u$ and its syntactic priority $\mathtt{ws}_u$.

Again, we have to consider several questions, for example, (i) which syntactic data should be prioritized: sequence header, GOP header, picture header, all of them, or none of them, (ii) should syntactic priority $\mathtt{ws}_u$ have higher value than content priority $\mathtt{wc}_u$, and (iii) how to assign the syntactic priority $\mathtt{ws}_u$. Our experiments show that there are no obvious answers for these questions. More details will be discussed in Section 3.2.5.6.

In our approach, we decide that if a packet contains sequence header or GOP header, its syntactic priority $\mathtt{ws}_u$ will be equal to $w_{\mathtt{max}}$. The syntactic priority of other packets is zero. This is because if these headers are lost, the corresponding sequence or GOP will be completely lost if no loss concealment is applied at receiver.

We choose $w_{\mathtt{max}}$ to be 4. As mentioned above, if macroblock m belongs to a protected region, its content priority $\mathtt{wc}_m$ is set to $w_{\mathtt{max}}$; otherwise $\mathtt{wc}_m = 1$. Thus, $w_{\mathtt{max}}$ could be any integer larger than 1, for example, 2. However, if we want to consider frame type in assigning syntactic priority, $w_{\mathtt{max}}$ needs to be larger than the syntactic priority of I-, P-, and B-frames, which could be set to 2, 1, and 0 respectively.

Therefore with $w_{\mathtt{max}} = 4$, high-content low-syntax packet (e.g., $\mathtt{wc}_u = 4$, $\mathtt{ws}_u = 0$ – on B-frame) would always have higher priority than a low-content high-syntax packet (e.g., $\mathtt{wc}_u = 1$, $\mathtt{ws}_u = 2$ – on I-frame). On the other hand, the maximum priority $w_{\mathtt{max}}$ could be larger than 4, but this would mean a larger range of values for w_u. It is not clear, however, whether having more different possible values for w_u is useful.

3.2.4 Priority-based scheduling

There are many ways to determine the sending order of packets, e.g., based on its priority, playout deadline, size, or combination of these. For scheduling based on priority, the most natural way is to send and retransmit packet with highest priority first [89, 95, 163]. Consequently, lowest-priority packet will be dropped first when there is not enough resource.

Since our main purpose is to study the effects of content-based prioritization, we opt to a simple scheduling algorithm, which is modified from FirstFit algorithm proposed by Chang [51]. Particularly, packets with highest priority will be sent first. If two packets have the same priority, the packet with earliest playout deadline will be sent first. If both packets have the same priority and deadline, the one with larger size will be sent first. The intuition behind is that given the same coding parameters, larger packet size means more data, thus probably more important.

If a packet is sent but its acknowledgment is not received after a window time W, the packet is considered lost and will be put in the scheduler's pending buffer again for possible retransmission. Packet whose playout deadline is over will be discarded from the pending buffer. Note that in our work, we do not require a QoS-enabled network to achieve priority-based streaming. When a packet is released into the network is solely decided by the scheduler, which resides at the sender. Once the scheduler sends out the packet, it does not want intermediate network nodes to reconsider its decision, but expect the network to simply transmit the packet to the other end.

3.2.5 Experiments and results

3.2.5.1 Prototype implementation

Figure 3.2 shows the main components of our simulation prototype. Since we want to study how different prioritizing schemes affects the streaming quality of a video, not how to schedule packets from different videos, in our experiments only one video is processed and streamed at a time. The processing server will automatically detect and track blobs in each video frame of the input video. Each frame is expected to be played at the receiver after a playout delay D. Macroblocks are prioritized, grouped into slices, which are in turn grouped into packets. Packets are prioritized according the content and syntax data they carry.

After a GOP is processed, its prioritized packets are put into the scheduler's pending buffer and transmitted to remote users (receivers) over a lossy link. As mentioned above, if a packet is sent but its acknowledgment is not received after a window time W, it will be put in the scheduler's pending buffer again. We assume that W is equal to RTT to study the performance in different network conditions. Packet is considered totally lost if it is not successfully received before its playout deadline.

At the receiver side, packet are received and acknowledged by packet receiver component. Received packets are then passed to frame reassembler to reassemble frame data. If sequence or GOP or picture header is lost, it will be replaced by the corresponding last-successfully-received header. Data are then passed to the frame decoder component, which will decode frames to produce the output video, and conceal any loss that occurs. A lost macroblock (or frame) will be replaced by its corresponding last-successfully-received macroblock (or frame).

The object tracking component is realized by modifying the *blobtrack* module in OpenCV version 1.0 (Open Source Computer Vision Library) [1]. Other components are implemented in C++ using Dalí Multimedia Library [215] and mnt (Multimedia

Network Toys) [214].

3.2.5.2 Test data and experiment settings

Our experiments use two video surveillance videos from PETS benchmark datasets. The first video (`pets2002-set1.mpg`) consists of 142 frames (640x240 pixels) extracted from the video `people_test_dataset1.mpg` [212]. The second video, named `Walk1-man.mpg`, consists of 200 frames (384x288 pixels) extracted from the video `Walk1.mpg` in CAVIAR test case scenarios [213]. Both videos are encoded in MPEG-1 IPPP format (one I-frame followed by 11 P-frames in one GOP) with frame rate of 25 fps. This frame pattern (without B-frames) is common among the networked video cameras to reduce latency in capturing and encoding.

In prioritization process, the maximum content priority, $w_{\texttt{max}}$ is set to 4. The value k, size of protected region around a blob, is set to 20%. Detailed explanation why these values are chosen is presented in Section 3.2.3.

A Markov 2-state model is used to simulate the *network* loss. If the network state is G (Good) then the packet is considered to be successfully received; if it is B (Bad) then the packet is dropped. The successful arrival of a packet is generated by the Markov model with the transition matrix $[1 - p_{GB}, p_{GB}; p_{BG}, 1 - p_{BG}]$, where $p_{GB} = 0.05$, $p_{BG} = p_{GB}(1-p)/p$ where p is the average loss rate. In our experiments, p is varied from 0% (zero percent) to 10%.

An important note is that the *network* loss rate does not cover the loss due to queue drop. When a packet is lost during transmission (Bad network's state), it will be put back into the scheduler's queue after a window time $W = RTT$. If the packet has the highest priority within the current queue, it will be retransmitted immediately. Otherwise, it will have to wait for its retransmission chance. However, if its playout deadline is over, the packet will be discarded from the pending buffer. This *queue* loss happens even when the *network* loss rate is equal to zero.

To cover different network conditions, RTT value is changed from 100ms to 300ms. An RTT of 100ms is typical for good connections such as cable or DSL, an RTT of 200-400ms normally occurs when connecting to a remote site. The playout delay D for each frame in both videos is 1000ms. The Maximum Transmission Unit (MTU) value is 1500 bytes. RTP packet is assumed to be sent over IPv4 network, with IP header of 20 bytes. All videos are streamed at their average data rate. We summarize the experiment parameters in Table 3.1.

Symbol	**Meaning**	**Value(s)**
k	Size of protected region (larger than blob size)	20%
MTU	Maximum Transmission Unit	1500 bytes
p	Average network loss rate	0%, 2%, 5%, 10%
p_{GB}, p_{BG}	Markov transitional probabilities	$p_{GB} = 0.05$
$W = RTT$	Round Trip Time	100ms, 200ms, 300ms
D	Playout delay	1000 ms
$w_{\mathtt{max}}$	Content priority for macroblocks within protected regions	4
	Content priority for macroblocks outside protected regions	1
	Syntactic priority for packets carrying sequence/GOP headers	4
	Syntactic priority for packets of I-frames	2
	Syntactic priority for packets of P-frames	1
	Syntactic priority for packets of B-frames	0

Table 3.1: Experiment parameters.

3.2.5.3 Frame-based prioritizing scheme

In order to study the effects of our content-based prioritizing scheme, we compare it with a frame-based prioritizing scheme. For the sake of brevity, our content-based prioritizing scheme is called `[Blob + SEQ]` scheme, and the frame-based one is named as `[PIC + SEQ]` scheme.

In `[PIC + SEQ]` scheme, the effective priority w_u of a packet u is equal to its syntax priority $\mathtt{ws}_u$, which is determined by both frame type, as well as sequence/GOP header. In our implementation, the effective priority is 2, 1 and 0 (zero) for a normal packet u belongs to an I-, P- and B-frame, correspondingly. If packet u contains sequence header or GOP header, its priority is increased by a value of $w_{\mathtt{max}}$ in the same way as our scheme.

The main difference between content-based and frame-based schemes is that information about interest regions is not used by the latter. While `[Blob + SEQ]` scheme reslices based on two criteria: macroblock's priority and slice's size; frame-based scheme does not care about macroblock's priority and just groups consecutive macroblocks into slices until the slice's size is reaching MTU value. Consequently, the overhead created by slicing and packetizing in frame-based scheme could be smaller than content-based scheme, thus its videos require a smaller sending rate than content-based prioritized video. Numeric measurements of the number of RTP packets (n_{RTP}) and the required sending rate r (in bytes/s, including RTP and IPv4 headers) for the tested videos are shown in Table 3.2.

		PIC + SEQ	Blob + SEQ	Overhead
pets2002-set1	n_{RTP}	541	588	8.7%
	r	122,333	123,850	1.2%
Walk1-man	n_{RTP}	1058	1125	6.3%
	r	169,370	171,082	1.1%

Table 3.2: Overhead of content-based prioritizing scheme compared to frame-based prioritizing scheme.

3.2.5.4 Evaluation metrics

To evaluate, we compare output videos produced by the two schemes under the same network conditions (average loss rate, bandwidth, RTT). We are mainly concerned with the interest regions in each frame in both subjective and objective tests.

Objectively, for each frame, we compare PSNR of the interest regions, called F-PSNR (Focused-PSNR) instead of comparing PSNR of the whole frame. The average PSNR (or F-PSNR) of each video is found by averaging MSE for all frames and then applying the PSNR equation [205]. The reported values are the average values of 15 runs.

Subjectively, 19 users are invited to evaluate the output videos using MSU Continuous Quality Evaluation method, provided by MSU Perceptual Video Quality tool [204]. For each network configuration, we compare three output videos (out of 15 videos) obtained from our content-based scheme with three corresponding output videos (out of 15 videos) obtained from frame-based scheme. Each of these three pairs is played side-by-side five times. The positions of our scheme's video, and frame-based scheme's video are randomly changed after each playing time, and users are not aware which is which. Users are asked to compare the quality of interest regions (each is bounded by a green ellipse), between 2 videos. The report Mean Opinion Score (MOS) [131, 143] for each pair is the average of MOS values from 19 users, calculated by ITU-R BT.500-11 recommendation [136].

3.2.5.5 Results and discussion

Figure 3.4 shows frame 119^{th} of video `pets2002-set1.mpg` obtained by our content-based scheme and frame-based scheme, when the video is streamed under an average loss rate of 5%, and RTT of 100ms. While the blob from our prioritizing scheme is intact, it is heavily damaged if frame-based scheme is used. Similar result for video `Walk1-man.mpg` is shown in Figure 3.5, when the video is streamed under an average loss rate of 5%, and RTT of 100ms.

Quantitatively, Figure 3.6 shows the average PSNR and F-PSNR for all frames of video `Walk1-man.mpg` obtained by our content-based scheme and frame-based scheme, when the video is streamed under an average loss rate of 2% and RTT of 100ms. The lower sub-figure compares F-PSNR results between the frame-based scheme and our scheme. Note that blobs only appear from frame 41 to frame 186, therefore F-PSNR values only exist within that range.

Since packets containing GOP header are highly protected in both schemes, the quality of blobs in I-frames are equal in all schemes. However for other frames, blobs from our scheme have much better quality while they rapidly degraded in frame-based scheme. The reason for this could be observed from the upper sub-figure, which shows

(a) Original video

(b) Content-based prioritizing scheme

(c) Frame-based prioritizing scheme

Figure 3.4: Video pets2002-set1 is streamed with a bandwidth equals to its average data rate, under an average network loss rate of 5% and an RTT of 100ms. Frame 119^{th} obtained from (a) original video, (b) content-based prioritizing scheme [Blob + SEQ], (c) frame-based prioritizing scheme [PIC + SEQ].

(a) Original video

(b) Content-based prioritizing scheme

(c) Frame-based prioritizing scheme

Figure 3.5: Video Walk1-man is streamed with a bandwidth equals to its average data rate, under an average network loss rate of 5% and an RTT of 100ms. Frame 83^{th} obtained from (a) original video, (b) content-based prioritizing scheme [Blob + SEQ], (c) frame-based prioritizing scheme [PIC + SEQ].

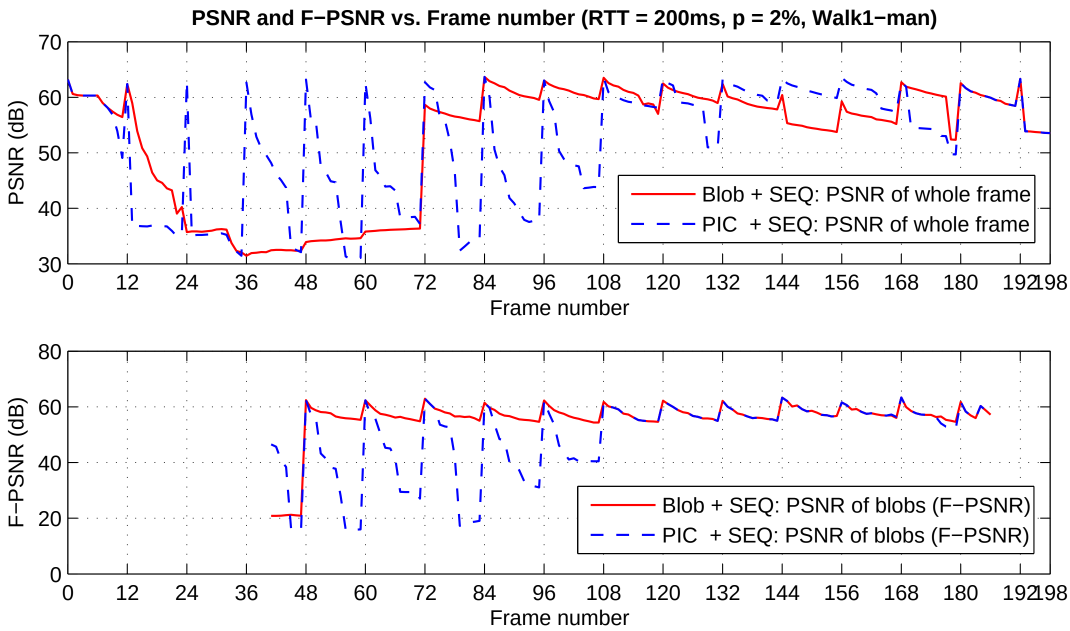
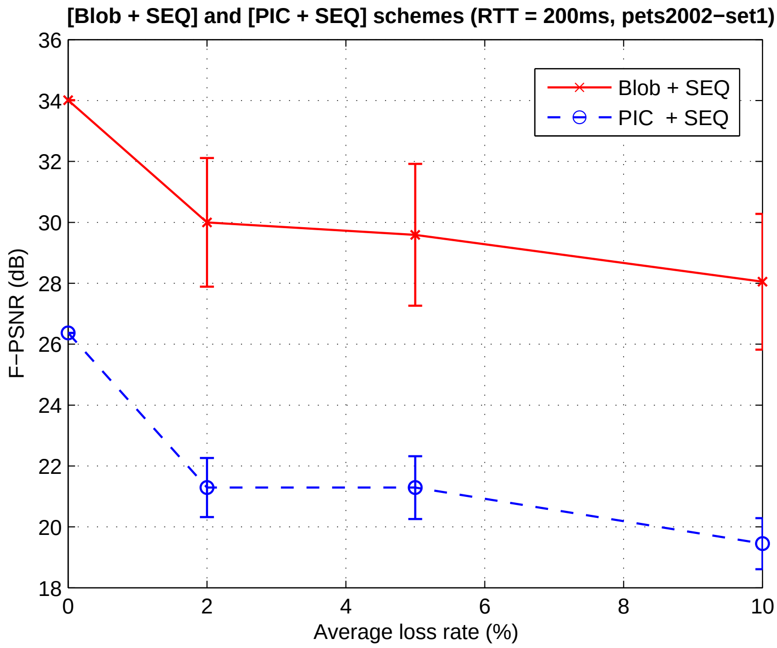


Figure 3.6: Average PSNR and F-PSNR vs. Frame number. Video Walk1-man is streamed with a bandwidth equals to its average data rate, under an average network loss rate of 2% and an RTT of 100ms.

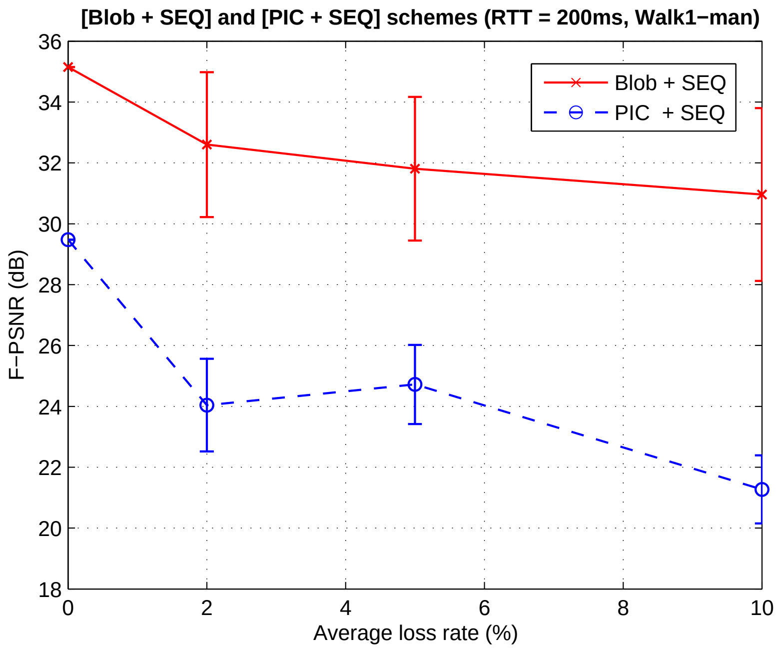
PSNR results of two schemes. While frame-based scheme gives equal protection for all GOPs, even for GOPs containing no blobs, the content-based scheme focuses more on GOPs related to blobs and on the blobs within each frame. In that way, the quality of non-event GOPs and frames are sacrificed, in order to achieve better quality for the regions of interest.

Figure 3.7 shows the average F-PSNR (with its standard deviation) of two prioritizing schemes for two videos `pets2002-set1.mpg` and `Walk1-man.mpg` under different network loss rates. The F-PSNR improvement by our scheme compared to the latter, indicated by the F-PSNR difference between two schemes, is shown in Table 3.3. Under different network conditions, the videos obtained from the content-based scheme always have a significantly higher blob's quality than what received from the latter, particularly around 8–11dB for `pets2002-set1.mpg` and 6–10dB for `Walk1-man.mpg`.

As mentioned in Section 3.2.5.2, a packet may be lost during transmission – determined by the *network* loss rate – or be dropped at scheduler's queue if its playout



(a) pets2002-set1



(b) Walk1-man

Figure 3.7: Average F-PSNR vs. Average network loss rate (streaming with average data rate, RTT = 200ms): (a) video pets2002-set1, (b) video Walk1-man.

Video	Loss rate	F-PSNR improvement		
		100ms	200ms	300ms
pets2002-set1	0%	8.09	7.64	10.71
	2%	8.77	8.71	7.67
	5%	7.94	8.30	7.91
	10%	7.99	8.60	8.13
Walk1-man	0%	5.67	5.67	5.67
	2%	8.80	8.56	7.21
	5%	7.94	7.09	8.25
	10%	9.00	9.69	9.42

Table 3.3: F-PSNR improvement (difference) of our content-based scheme vs. the frame-based scheme.

deadline is over. The *queue* loss happens because of various constraints such as play-out deadline, transmission bandwidth – these parameters cannot have infinite values to assure all packets are transmitted on time. Therefore, even when the *network* loss rate is equal to zero, we may still have packet loss at scheduler's queue.

We can observe the effects of these types of loss from Figure 3.7 and Table 3.3, as follows. When the average *network* loss rate increases, F-PSNR values of both videos decrease as expected. The F-PSNR decrement of videos from `[Blob + SEQ]` scheme is gradual while `[PIC + SEQ]` scheme creates steeper declination. That is, videos prioritized and packetized by `[PIC + SEQ]` scheme are more affected by network loss than videos processed by the content-based scheme. Even so, the decline is not very sharp, indicates the limited effects of network loss rate upon blob's quality in this case.

On the other hand, the effects of *queue* loss is rather substantial and observable from the two following facts. First, when the only loss is due to queuing (average *network* loss rate is zero), two prioritizing schemes show noticeable F-PSNR difference: 7.64 dB for `pets2002-set1.mpg` and 5.67 dB for `Walk1-man.mpg`. Second, the F-PSNR improvements at other network loss rates are not much different from that at zero network loss rate. It means that queue loss is the main factor in making the difference here. This is not surprising, since blob-related packets are highly prioritized

		Blob + SEQ		PIC + SEQ	
		MOS_a	σ	MOS_a	σ
pets2002-set1	Pair 1	**9.16**	1.20	**0.84**	1.20
(loss rate = 5%,	Pair 2	**9.05**	1.09	**0.95**	1.09
RTT = 100ms)	Pair 3	**8.39**	1.53	**1.61**	1.53
Walk1-man	Pair 1	**7.78**	1.88	**2.22**	1.88
(loss rate = 5%,	Pair 2	**9.13**	1.28	**0.87**	1.28
RTT = 100ms)	Pair 3	**8.90**	1.81	**1.10**	1.81

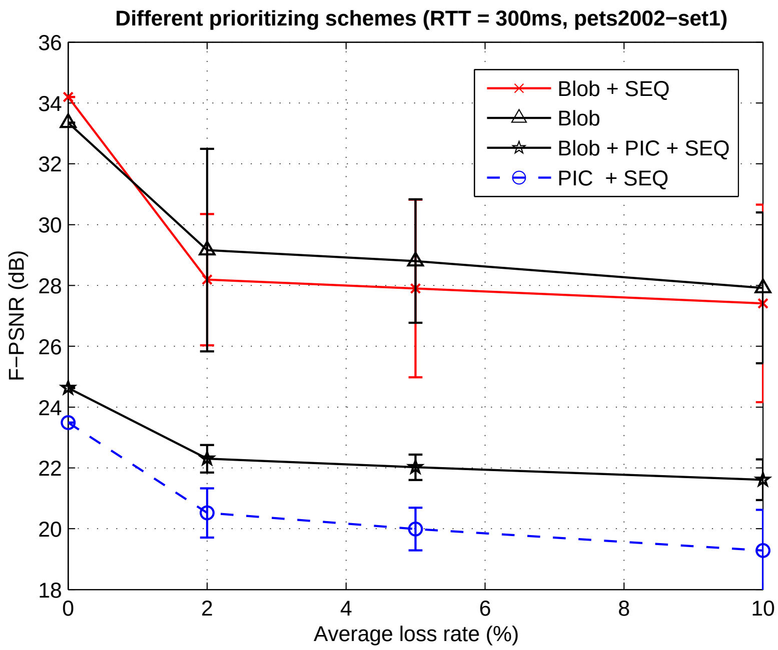
Table 3.4: Mean Opinion Score average (MOS_a) and standard deviation σ calculated by ITU-R BT.500-11 recommendation.

by `[Blob + SEQ]` scheme, thus they will not likely to be dropped from scheduler's queue. Meanwhile, they are not differentiated from non-blob packets by `[PIC + SEQ]` scheme, and may lose their transmission chances to non-blob important-syntax packets.

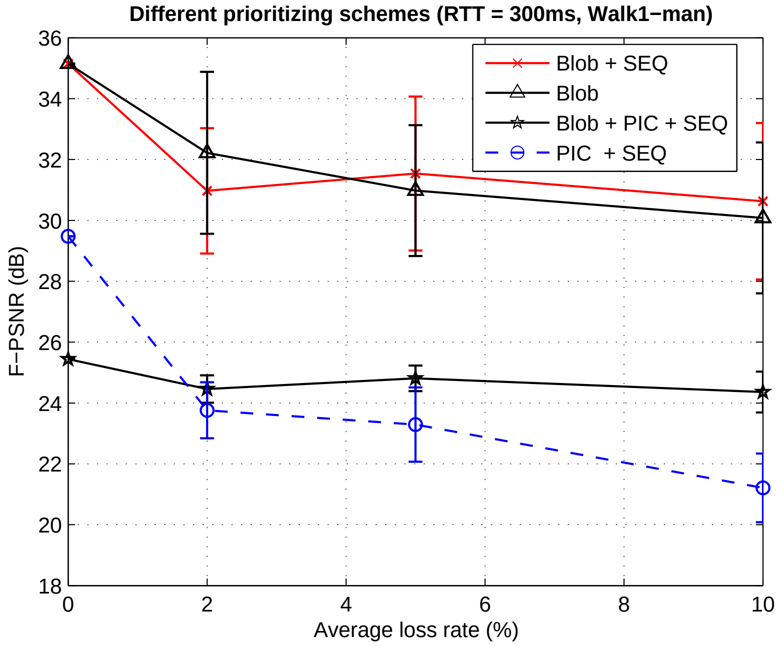
Since PSNR may not well reflect the perceptual quality of video [311], subjective tests with 19 users are carried out. The user set includes 5 women and 14 men, 2 video experts and 17 non-experts. For each prioritizing scheme, we test two videos under the following network: average network loss rate of 5%, and RTT of 100ms. The average Mean Opinion Score results, summarized in Table 3.4, strongly confirm the subjective improvements. Using the proposed content-based scheme, the perceptual quality of interest regions is significantly better compared to the quality obtained by using frame-based scheme.

3.2.5.6 Further discussion

One arising question is that what will happen if we only prioritize packets based on its content and not include syntax data such as sequence/GOP header into the consideration. Conversely, what will happen if we consider all sequence/GOP header and picture header in the prioritizing process? Denote the first scheme `[Blob]`, the latter `[Blob + SEQ + PIC]`. Figure 3.8 shows some results of these and `[Blob + SEQ]`, `[PIC + SEQ]` schemes.



(a) pets2002-set1



(b) Walk1-man

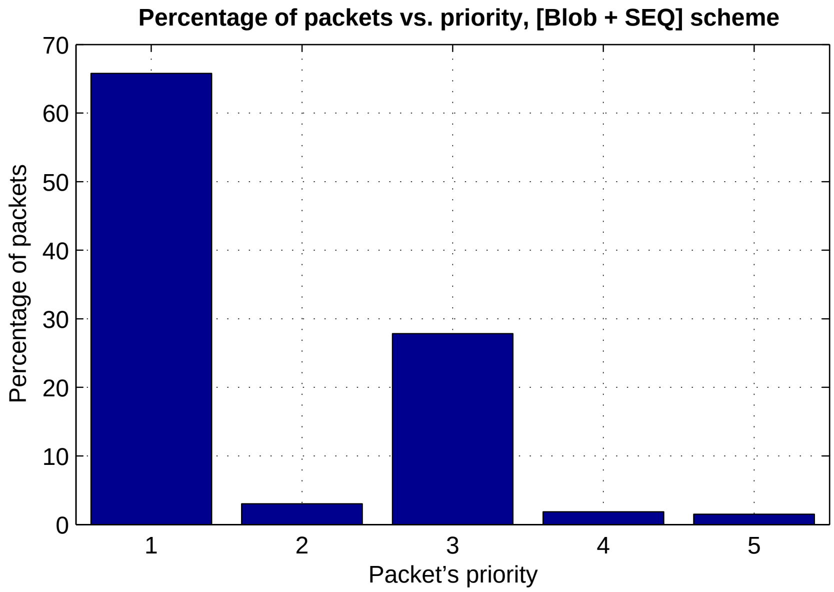
Figure 3.8: Average F-PSNR of different prioritizing schemes vs. Average network loss rate (streaming with average date rate, RTT = 300ms): (a) video pets2002-set1, (b) video Walk1-man.

Comparing between `[Blob]` and `[Blob + SEQ]` schemes, we can see that excluding sequence header from prioritization may even slightly increase the quality of blobs. It happens despite the fact that the loss of a sequence/GOP header may make that whole sequence/GOP undecodable. This is mainly due to our implementation, particularly (i) in the video encoding process, sequence header is inserted at the beginning of every GOP, and (ii) at the receiver, any lost sequence/GOP header is replaced by the last successfully received header. Furthermore, while `[Blob + SEQ]` scheme gives high protection for sequence/GOP header of every GOP, `[Blob]` scheme only protects macroblocks/slice/packets related to blobs, whose decoding dependencies are already taken into account during our prioritizing process. Therefore, the latter scheme may be more efficient than the first one.

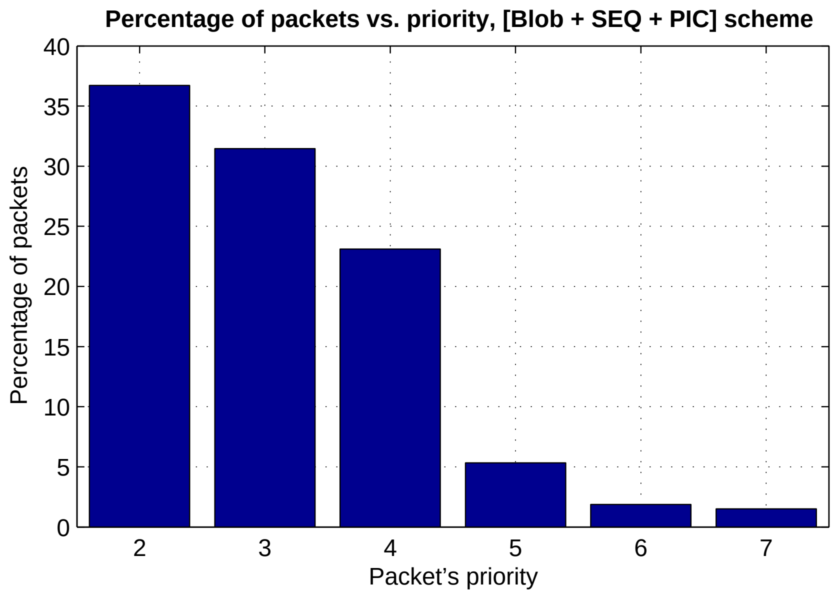
For example, in `Walk1-man.mpg`, blobs only appear in the lower right corner of frame. Therefore, packets with GOP header (the first packet in I-frame) and packets with PIC header (the first packet in all frames) usually do not contain blob-related data. That explains the big gap between `[Blob + SEQ]` and `[Blob]`, as well as between `[Blob + SEQ]` and `[PIC + SEQ]` scheme (see Figure 3.8). Giving high priority to packets with GOP/PIC header in this case is not only unnecessary, but also may hinder blob-related packets from being sent.

Surprisingly, considering frame type in prioritization (changing from `[Blob + SEQ]` to `[Blob + SEQ + PIC]` scheme) does not improve the blobs' quality, but significantly decreases it. This is, at first, counter-intuitive since with more protection, ones may expect better results. However, adding frame-based priority may in fact reduce our emphasis on content and confuse the scheduler, because the priority of a high-content low-syntax packet may equal to the priority of a non-content high-syntax packet. In that case, there is no difference between them, and the high-content packet may lose its slot to the non-content packet.

For instance, in `[Blob + SEQ + PIC]` scheme, a packet having a priority of 4 could



(a) Blob + SEQ



(b) Blob + SEQ + PIC

Figure 3.9: Percentage of packets vs. Packet's priority prioritized by different schemes: (a) [Blob + SEQ], (b) [Blob + SEQ + PIC].

be either a packet with content priority of 3 and syntactic priority of 1 (on P-frame) or a packet with content priority of 2 and syntactic priority of 2 (on I-frame). For such reasons, the priority distribution of packets obtained from `[Blob + SEQ + PIC]` scheme is quite different from what is produced by `[Blob + SEQ]` scheme (see Figure 3.9), thus could significantly affect packet scheduling and the received quality.

3.2.6 Remarks

In this section, a simple and efficient *content-based priority streaming* scheme for video surveillance is proposed. In this scheme, videos are analyzed, prioritized, resliced, packetized and streamed based on its visual content (regions of interest – blobs) as well as its syntax data. To measure the quality of blobs within a frame, we propose a new metric, named Focused-PSNR (F-PSNR), which is simply the PSNR of blobs. Experiments show that by focusing on content instead of syntax data, the quality of blobs could be significantly improved. Compared with frame-based priority streaming, objective results shows a 6–11dB F-PSNR improvement by our scheme, and subjective measurement with 19 users strongly confirms the usefulness of our approach. Experiments with other prioritizing schemes produce some counter-intuition results and show that a simple prioritization approach may be more effective than an over-protective prioritizing scheme.

3.3 FEC for content-based priority streaming

3.3.1 Introduction

Reed-Solomon (RS) code is one of the most popular Forward Error Correction (FEC) methods to reduce the effects of packet loss in multimedia transmission [65, 224, 274, 304, 315]. The beauty of RS code is that given K original packets, if $(N - K)$ RS-coded packets are generated and sent together with these original packets, the original packets could be recovered if at least K out of these N packets are received. Nevertheless, how to determine K, N, and how to choose which K original packets to protect are normally dependent on application requirements and network conditions. Larger K means longer waiting time and larger buffer at the receiver. Higher code rate K/N means less overhead in exchange of lower protection ability. Since in this section, we only use RS-code FEC and not any other method of FEC (e.g., parity code, media-specific FEC [224]), the shorter term "FEC" is be used instead of the term "RS-code FEC" hereafter.

Various works have shown that compared to fixed FEC, better results could be achieved if the values of K and K/N are adjusted according to packets' importance, network loss rate, Round Trip Time (RTT), bandwidth, etc., [202, 274, 314]. In these Unequal Error Protection (UEP) schemes, original packets are usually classified at the frame level and protected based on their importance to the *decoder*. For example, packets from I-frame, P-frames, and B-frames could be divided into 3 classes [314,315]. Cai et al. [45,46] classify MPEG-4 video data into two classes: one includes important data such as I-frames, Video Object Plane header, Video Packet header; the other contains texture data. Accordingly, a syntax-based FEC scheme will try to maximize a syntax-based quality metric, such as frame rate [314] or video frame's PSNR [46].

These quality measurements, however, may not be sufficient, and of importance, to applications such as video surveillance, video conferencing, and telemedicine. In these applications, certain regions within frames could be more important to the *users* than the others. Let's consider a surveillance video, which usually contains many idle and non-event segments interleaved with some short bursts of events. Within each eventful video segment, some frames may be more essential to the users than the others. In each frame, users may only focus on certain *regions of interest*, such as human, cars, colorful objects, or moving objects, and do not care about the rest. Ultimately, the quality of what users focus on would be their yardstick to judge quality of the whole video. Therefore, it would be better to protect the actual *content* that users are interested in, rather than the important syntactic *data*.

In a distributed video surveillance system, video from multiple surveillance cameras could be captured and sent to a processing server. The processing server may detect motion in a scene, highlight and track moving objects (blobs) in the video, then transmits the video to remote users over network links, which are often lossy and have limited bandwidth. Naturally, these blobs would be users' regions of interest and they should be prioritized, protected and transmitted with higher priority.

Section 3.2 deals extensively with content-based prioritization. For protection during streaming, it relies on a simple priority retransmission scheme, in which packet with higher priority will be sent first. It demonstrated that prioritizing then streaming packets (with retransmission) based on semantic *regions of interest* could bring about substantial quality improvements. For example, content-based priority streaming could produce blobs' quality 6-11dB higher than what obtained by frame-based priority streaming.

While this section take advantages of the content-based prioritizing scheme from Section 3.2, we shift our focus to FEC protection (see Figure 3.10). Particularly, we investigate the following questions: (i) How to allocate FEC for a set of prioritized packets to maximize its expected total priority? (ii) Would it be better to use content-based priority, rather than frame type, in classifying packets and allocating FEC protection?

We propose a general *content-based FEC* scheme in which packets are selected and protected based on their content-based priority. Our experiments show that *content-based FEC* (FEC with packets' content-based priority) always produces better quality than *frame-based FEC* (FEC with frame-based information).

The rest of this section is organized as follows. Section 3.3.2 gives an overview of related works. In Section 3.3.3, the content-based FEC and frame-based FEC schemes

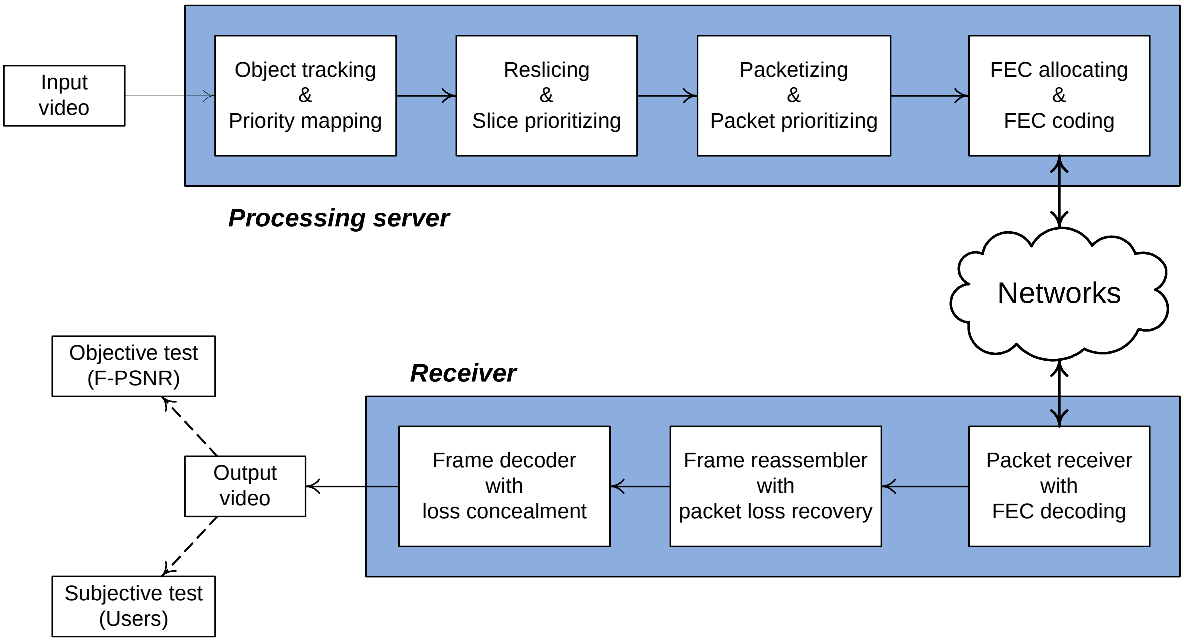


Figure 3.10: The content-based priority streaming prototype with FEC.

are presented. Section 3.3.4 talks about our prototype's implementation, evaluation metrics, and experiment results. Finally, we conclude in Section 3.3.5.

3.3.2 Related works

In Section 3.2.2, we have already talked about works on content-based prioritization. This section will briefly present some related research on content-based FEC.

Content-based protection could be carried out at different phases: encoding or transmission. Content-based encoding has been studied and applied extensively, e.g., for objects coding in MPEG-4 [53, 294] and Region of Interest (ROI) coding in JPEG 2000 [43]. In content-based transmission, source and channel coding (e.g., FEC [104], interleaving [62]) are usually combined with content's information to determine level of protection [104], to allocate appropriate bandwidth [33, 318] or bit rate [166, 333, 334].

While Section 3.2 focuses on content-based prioritization and retransmission-based loss recovery, this section focuses on channel coding to protect packets with FEC.

Furthermore, we assume that packets have been already packetized and prioritized by our content-based prioritizing scheme, instead of considering both source and channel coding at the same time like Frossard's work [104].

The problem of how to optimize the FEC allocation is not new and has been widely studied in various works [45, 46, 198, 202, 274, 280, 313–315]. Generally, FEC is calculated for a small number of frames or packets, e.g., a GOP of 15–25 frames, to reduce the waiting time and buffer size at receivers [45, 280, 313–315]. Packets in each group could be classified into different data classes based on type of the frame they belong to, their contribution to output video in terms of visual quality or frame rate [198, 313–315], or their potential distortion level [45, 46], etc. After that, the FEC allocation problem is reduced to finding how many FEC packets to protect how many original packets in each data classes, so that our objective is optimized. This is usually achieved by exhaustive search or using Lagrange method.

Normally, the criteria using to divide packets into class are closely related the quality we want to optimize. If frame type is used in classifying, the objective is often to maximize the reconstructed frame rate. If quality distortion is used, the objective is either maximizing the video quality or minimizing total distortion.

While most works focus on frame level and classify packets based on their frame type, we focus at packet level – where FEC actually happens – and classify them based on their priority. Such priority value could be assigned based on their semantic content, their frame type, etc. Besides, while most works assume that each frame/layer will be wholly protected by FEC, we allow partial protection, i.e., FEC protection for only a number of packets within a class. The details of our optimized FEC allocation schemes and more subtle differences will be presented and discussed in Section 3.3.3.

3.3.3 Content-based FEC scheme

In this section, we will describe how to apply our *content-priority FEC* scheme for a set of original packets O, say all packets from 1 GOP. Section 3.2 showed that by giving high priority for packet with content or carrying sequence/GOP header, the quality of blobs could be significantly improved compared to frame-based prioritization. Therefore, we assumed that packets are prioritized using our content-based scheme (`[Blob + SEQ]` prioritizing scheme) described in Section 3.2.3. By that way, only one packet carrying sequence/GOP header has a priority higher than $w_{\texttt{max}}$ in each GOP, all other packets will have priority within the range of $[1, w_{\texttt{max}}]$.

A common optimization objective is to minimize the overall distortion [59, 105, 173, 280]. While it is possible to optimize this objective in our framework in general, the application domain we are studying deals with live, real-time video streams. In such scenario, computing the distortion values for every packet is not feasible. We therefore opt to optimize based on total priority, which is easier to compute than distortion.

Our scheme consists of two main steps. First, all packets in the GOP are classified into different data classes based on their priority. Then, for each classes we choose how many and which original packets should be protected, and the number of FEC packets should be used in order to maximize the expected total priority of the GOP, given the transmission rate constraint and average network loss rate.

3.3.3.1 Packet classification

We divide packets into $v_{\texttt{max}} = w_{\texttt{max}} = 4$ data classes as follows.

- Data class O_4: contains packets with $w_u \geq 4$.
- Data class O_3: contains packets with $w_u = 3$.
- Data class O_2: contains packets with $w_u = 2$.

- Data class O_1: contains packets with $w_u = 1$.

Thus, data class O_v contains more important packets than data class $O_{v'}$, with $v > v'$. We then could protect each class with different number of FEC packets, depending on the class's importance. Note that if $w_{\texttt{max}}$ is larger than 4, the priority range $[1, w_{\texttt{max}}]$ could and should be divided into several sub-ranges, so that the maximum number of data class $v_{\texttt{max}}$ is small.

3.3.3.2 Packet selection and FEC allocation

To show how to allocate FEC for a set of original packets O from a GOP, we denote:

- n_G: The number of frames within the GOP.
- f: Frame rate of the video [fps].
- O_v: The set of original packets in data class v, $v = 1..v_{\texttt{max}}$.
- $n(O_v)$: Number of packets in O_v.
- $s(O_v)$: Size (in bytes) of the largest-size packet in O_v.
- S_v: The set of original packets which are chosen from O_v to be sent.
- $n(S_v)$: Number of packets in S_v.
- F_v: The set of FEC packets protecting S_v.
- $n(F_v)$: Number of packets in F_v.
- p: Average network packet loss rate.
- MTU: Maximum Transmission Unit (e.g., 1500 bytes).
- B: The actual sending budget [bytes] used to send S_v and F_v, $v = 1..v_{\texttt{max}}$.

- B_m: The maximum sending budget [bytes] could be allocated to send S_v and F_v, $v = 1..v_{\texttt{max}}$.
- R: The maximum network transmission rate [bytes/s] could be used.
- W: The expected total priority after sending S_v and F_v, $v = 1..v_{\texttt{max}}$

The value of B_m could be determined from the transmission rate R, the number of frames in the GOP n_G, and the video's frame rate f as in Equation 3.3.

$$B_m = R\frac{n_G}{f} \tag{3.3}$$

Packet selection: Since B_m may be limited compared to the video's requirement, not all packets of the set O could be sent. Therefore, we need to select which classes and which packets within each class to protect first.

From Section 3.2.3, we see that a macroblock is prioritized based on their semantic content and its reference relationship with macroblocks from other frames. Obviously, less important content (e.g., background packets) could be discarded to send more important packets (e.g., blob-related ones). Beside, a packet with higher priority not only carries more important macroblocks, but could also be referred by packets with lower priority during decoding process. Therefore, all packets in data class O_v should be selected first before packets in data class $O_{v'}$ if $v > v'$. That is, data classes O_v would be selected in the order of decreasing v (from $v_{\texttt{max}}$ to 1).

Within a data class O_v, if a packet u has earlier deadline than packet u' then u should be selected first. Because u could be a reference of u', losing u would be more devastated than losing u'. To be more accurate, we can further divide packets within each data class O_v into 3 sub-classes based on their frame type, then select packets within each class based on their frame type, e.g., I-frame sub-class first, then P-frame sub-class, then B-frame sub-class. However, it will substantially increase the computing time, since the number of classes (and parameters) is tripled.

FEC allocation: The probability that the set S_v, protected by F_v and sent over a link with loss rate p, will be successfully reconstructed at the receiver is the probability that at least $K = n(S_v)$ packets will be received out of $N = n(S_v) + n(F_v)$ packets. It is calculated in Equation 3.4 as follows.

$$q(N, K, p) = \sum_{i=K}^{N} \left[\begin{pmatrix} N \\ i \end{pmatrix} (1-p)^i p^{N-i} \right] \tag{3.4}$$

In short, the FEC allocation problem is to find $n(S_v)$, $n(F_v)$ for all data classes to maximize the expected total priority W, subject to the sending budget constraint. Assuming that the maximum protect ratio $n(F_v)/n(O_v) = 1/1$, mathematically, we want to find $n(S_v) \in [0, n(O_v)]$, $n(F_v) \in [0, n(S_v)]$ for $v = 1..v_{\mathtt{max}}$, so that the sending budget B

$$B = \sum_{v=1}^{v_{\mathtt{max}}} \{[n(S_v) + n(F_v)] \times s(O_v)\} \tag{3.5}$$

satisfies the condition

$$B_m - MTU < B \leq B_m \tag{3.6}$$

and maximize the expected total priority W

$$W = \sum_{v=1}^{v_{\mathtt{max}}} [q\,(n(S_v) + n(F_v), n(S_v), p) \times n(S_v) \times v] \tag{3.7}$$

The expected total priority W is estimated in Equation 3.7, with the assumption that all packets in class O_v having priority of v. It is not accurate since in class O_4, the packet carrying sequence/GOP header has a priority higher than $v = v_{\mathtt{max}} = w_{\mathtt{max}} = 4$. This assumption, however, will not change the result of the FEC allocation.

Furthermore, the sending budget B is estimated by Equation 3.5 with the assumption that all packets in data class O_v and FEC class F_v have the size of $s(O_v)$. Since our packetizing process tries to put as much data as possible into a packet until reaching the MTU limit, the size of most packets would be similar (around and less than MTU).

Note that for the sending budget constraint in Equation 3.6, we require both lower bound and upper bound for the sending budget B. The upper bound is to prevent the sending budget exceeding the maximum sending budget B_m, like in other works. The lower bound means that if we still have room to send one more packet (size less than MTU), we will. This is to make sure that the whole maximum sending budget B_m would be used. Besides, when we do exhaustive search in the decreasing direction – $n(S_v)$ from $n(O_v)$ to 0, and $n(F_v)$ from $n(S_v)$ to 0 – this lower bound could be used as a break condition to substantially reduce the searching space and time.

The *content-based FEC* scheme could be easily modified to become a *frame-based FEC* scheme. The main modification is in packet classification step, in which packets are divided into classes solely based on the frame type they are belong to, instead of based on their priority. Therefore, we will have 3 classes if the GOP has all I-, P-, B-frames, or only 2 classes if the GOP has I-, P-frames. Packet selection and FEC allocation procedure could be applied without any changes.

3.3.4 Experiments and results

The purpose of our experiments is to study the effects of different FEC schemes on the streaming quality of surveillance videos. While *content-based FEC* scheme uses priority of packets – indicator of their content's importance – to classify and allocate FEC protection, *frame-based FEC* scheme uses frame type information. Both schemes optimize their FEC allocation for a given set of content-based prioritized packets, given a sending budget constraint and packet loss probability.

Similar adaptive frame-based FEC schemes, which optimize and adjust FEC allocation according to rate constraint and loss, have been compared with Equal-Error-Protection (EEP), fixed-FEC, or non-FEC schemes in various works [46,274,313,314]. All conclude that adaptive FEC schemes always has better performance than EEP, fixed-FEC and non-FEC schemes. Therefore, the latter schemes are not considered in our experiments.

3.3.4.1 Prototype implementation

Figure 3.10 shows the main components of our simulation prototype. The processing server will automatically detect and track blobs in each video frame of the input video. Each frame is expected to be played at the receiver after a playout delay. Macroblocks are prioritized, grouped into slices, which are in turn grouped into packets. Packets are prioritized according the content and syntax data they carries.

After a GOP is processed, its prioritized packets are divided into classes and protected by FEC, using one of the schemes described in Section 3.3.3. Original and FEC packets are then transmitted to remote users (receivers) over a lossy link, after a random network delay around $RTT/2$. A packet is considered totally lost if it is not successfully received before its playout deadline. In this section, we assume the playout delay is long enough so that no packet is late, thus all loss are due to network loss – determined by an average network loss rate.

At the receiver side, packet are received and recovered using FEC, without any retransmission. Original and recovered packets are then passed to frame reassembler to reassemble frame data. If sequence or GOP or picture header is lost, it will be replaced by the corresponding last-successfully-received header. Data are then passed to the frame decoder component, which will decode frames to produce the output video and conceal any loss if occurs. A lost macroblock (or frame) will be replaced by its corresponding last-successfully-received macroblock (or frame).

The object tracking component is realized by modifying the *blobtrack* module in OpenCV version 1.0 (Open Source Computer Vision Library) [1]. Other components are implemented in C++ using Dalí Multimedia Library [215] and mnt (Multimedia Network Toys) [214].

3.3.4.2 Test data and experiment settings

Our experiments use two video surveillance videos from PETS benchmark datasets. The first video (`pets2002-set1.mpg`) consists of 142 frames (640x240 pixels) extracted from the video `people_test_dataset1.mpg` [212]. The second video, named `Walk1-man.mpg`, consists of 200 frames (384x288 pixels) extracted from the video `Walk1.mpg` in CAVIAR test case scenarios [213]. Both videos are encoded in MPEG-1 IPPP format (one I-frame followed by 11 P-frames in one GOP) with frame rate of 25 fps. This frame pattern (without B-frames) is common among the networked video cameras to reduce latency in capturing and encoding. The average data rate of `pets2002-set1.mpg` is about 124000 bytes/s, and that of `Walk1-man.mpg` is about 172000 bytes/s.

A Markov 2-state model is used to simulate the network packet loss probability, which is the only reason for loss in this case since no delay is considered. If the network state is G (Good) then the packet is considered to be successfully received; if it is B (Bad) then the packet is either late or corrupted. The successful arrival of a packet is generated by the Markov model with the transition matrix $[1-p_{GB}, p_{GB}; p_{BG}, 1-p_{BG}]$, where $p_{GB} = 0.05$, $p_{BG} = p_{GB}(1-p)/p$ where p is the average loss rate. In our experiments, p is varied from 0% to 10%.

Since FEC is normally used when retransmission is not applicable, for example, when RTT is large or larger than the delay requirement, our experiments are carried out with the RTT value of 300ms. It means one-way delay of 150ms, which is around the upper-bound value of one-way delay for two-way videophone or interactive applications, according to ITU recommendation [134]. The Maximum Transmission Unit (MTU) value is 1500 bytes. RTP packet is assumed to be sent over IPv4 networks, with IP header of 20 bytes. Videos are streamed at different transmission rates, varied from 60% to 140% of their average data rates.

Since we want to study how different FEC schemes affects the streaming quality of a video, not how to schedule packets from different videos, in our experiments only one video is processed and streamed at a time.

3.3.4.3 Evaluation metrics and results

To evaluate, we compare output videos protected by the two FEC schemes under the same network conditions (average loss rate, bandwidth, RTT). For each frame, we compare F-PSNR, which is the PSNR of interest regions within the videos, instead of comparing PSNR of the whole frame. Our tests in Section 3.2 already showed that F-PSNR is highly correlated with the subjective quality perceived by users and measured by Mean Opinion Score (MOS) metric. The reported values are the average values of 15 runs.

Figure 3.11 shows the original frame 62^{th} of video `pets2002-set1.mpg` and the corresponding frame obtained by our content-based FEC scheme and frame-based FEC scheme, when the video is streamed with a bandwidth equals to its average data rate, under an average loss rate of 5%, and RTT of 300ms. Both frames obtained from these two schemes are damaged, however content-based FEC scheme leaves the blob intact and shift most damaging effects to less important areas. The inverse result is seen in the frame obtained by frame-based FEC scheme, since blobs are not put in high level protection here.

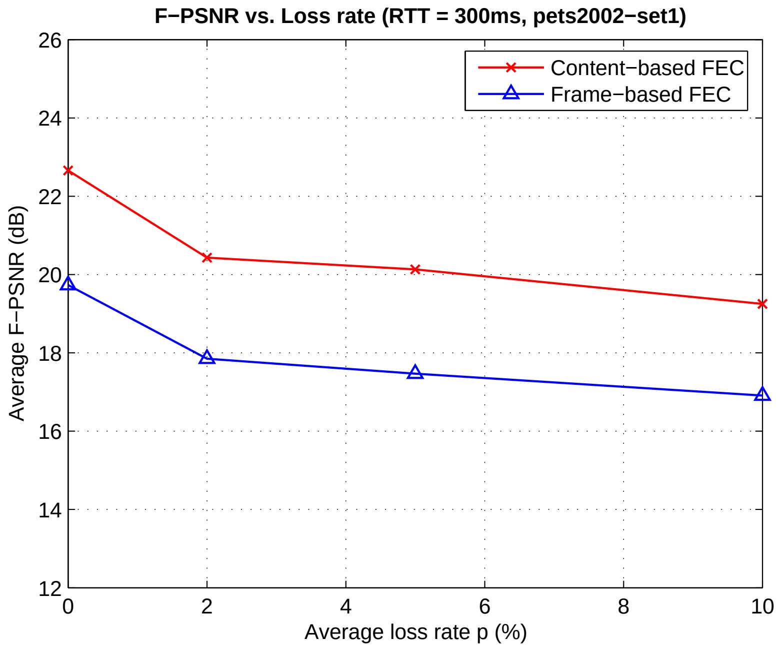
Figure 3.12 shows the average F-PSNR of `pets2002-set1.mpg` and `Walk1-man.mpg`, when they are protected by different FEC schemes, then streamed at transmission rates equal to their average data rate and under different packet loss rates. It is clear

(a) Original video

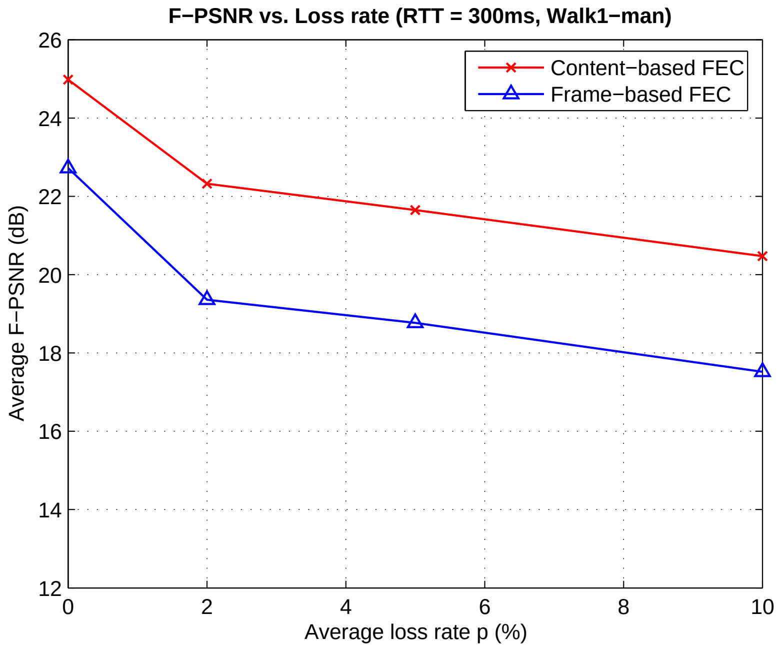
(b) Content-based FEC scheme

(c) Frame-based FEC scheme

Figure 3.11: Video Walk1-man is streamed with a bandwidth equals to its average data rate, under an average network loss rate of 5% and an RTT of 300ms. Frame 62^{th} (P-frame) obtained from (a) original video, (b) content-based FEC scheme, (c) frame-based FEC scheme.



(a) pets2002-set1



(b) Walk1-man

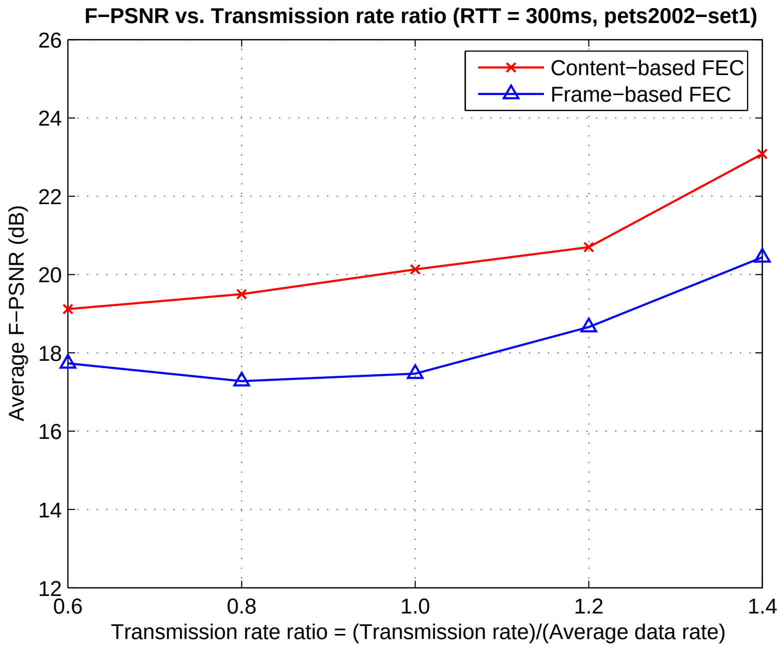
Figure 3.12: Average F-PSNR vs. Average network loss rate (streaming with average data rate, RTT = 300ms): (a) video pets2002-set1, (b) video Walk1-man.

that the blob's quality obtained from the content-based FEC scheme is always higher than that obtained from the latter scheme. The F-PSNR differences are around 2–3dB for both videos, which are equivalent with 10–17% improvement compared to frame-based FEC scheme.

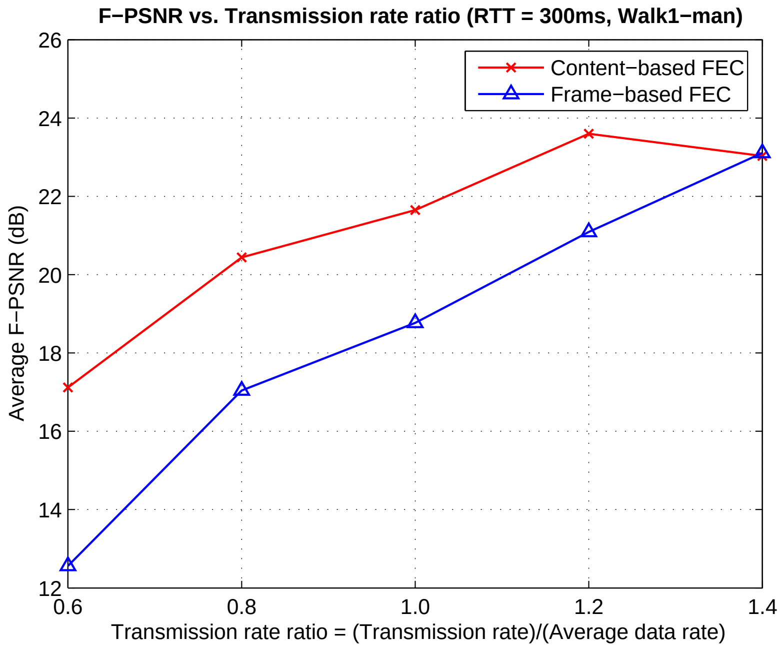
Figure 3.13 shows the average F-PSNR of the two videos when they are streamed under different network transmission rates, and an average packet loss of 5%. The x-axis is transmission rate ratio r, which is equal to the ratio between network transmission rate and video's average data rate. Let's consider the video `Walk1-man.mpg`. At the transmission rate ratio of 0.6, the quality of blob obtained by content-based FEC is 4.6dB or 36% higher than that from frame-based FEC scheme. This improvement decreases when the transmission rate is increased, from 20% (3.4dB) at $r = 0.8\%$ to 12% (2.5dB) at $r = 1.2$. The performances of two FEC schemes only converge when the transmission rate ratio r is 1.4 times higher than the average data rate of `Walk1-man.mpg`. Even at this high ratio, content-based FEC scheme still manages to obtain an improvement as high as 13% (2.64dB) in the case of `pets2002-set1.mpg`.

3.3.5 Remarks

We proposed a efficient scheme, named *content-based FEC*, to classify and optimize FEC allocation based on packet's content-based priority. Compared to *frame-based FEC* scheme where frame type information is used for packet classifying and FEC allocating, the proposed *content-based FEC* scheme could achieve an improvement of 10–17% under normal conditions. At higher packet loss rates or severer bandwidth constraint, our scheme could outperform frame-based FEC scheme from 17% to 36%. The results showed that by protecting what is important to users (e.g., blobs within frames), rather than what is important to the video itself (e.g., I-frames), the perceived quality could be significant improved.



(a) pets2002-set1



(b) Walk1-man

Figure 3.13: Average F-PSNR vs. Average network loss rate (streaming with average packet loss of 5%, RTT = 300ms): (a) video pets2002-set1,(b) video Walk1-man.

3.4 Summary

The main contributions of this chapter are summarized as follows. Firstly, we proposed prioritization of video packets for streaming based on the semantics of the video. Our approach is a departure from the conventional approach, where prioritization is based on syntax. Secondly, we proposed a simple but effective way to prioritize packets based on visual content of the video. Instead of focusing on coding, the proposed scheme focuses on prioritizing and scheduling of packets, which is simpler and faster, and works with commodity networked video cameras. Thirdly, a simple metric, named Focused-PSNR (F-PSNR), was proposed to measure the quality of interest regions within a frame. Finally, we presented an optimized content-based FEC scheme to classify and protect video packets based on their content contribution.

Experiments show that significant improvements could be obtained with our proposed approaches. With retransmission-based scheduling and streaming, content-based prioritizing scheme can achieve 6-11dB improvement in quality of tracked object (blob) compared to frame-based prioritizing scheme. When FEC is used instead of retransmission during streaming, content-based FEC scheme can outperform frame-based FEC scheme by 10–17% in terms of blob's quality (2–3dB) if videos are transmitted at their average data rate. If transmission rates are much lower, for example, equal to 60% of the average data rates, content-based FEC scheme can achieve an improvement up to 36% (4.6dB).

Chapter 4

Scheduling for content-based prioritized packets

The key is not to prioritize what's on your schedule,

but to schedule your priorities.

—Stephen R. Covey

4.1 Introduction

In Chapter 3, we have shown that by prioritizing video based on semantic content then streaming packets based on their priority, significant improvement in perceptual quality could be achieved. In our experiments, we used a modified version of First-Fit algorithm, which always sends the highest-priority packet first. This scheduling scheme, by emphasizing packet's priority, seems to be a natural candidate to schedule prioritized packets.

However, what would happen if we schedule packets based on their deadline, instead of their priority? Would it be better if we consider both priority and deadline at the same time? Or would it be even better if additional information about networks, such as RTT, is used in making schedule decisions? The purpose of this chapter is to study what and how information about packets and network should be used in scheduling prioritized packets.

In the literature, the most common packet's features to be used in scheduling are deadline and priority. Usually, schedulers either implicitly or explicitly use information on *priority* (the packet's contribution/effect on the quality of reconstructed data) and/or *deadline* – the latest time that the packet has to be received. For example, priority-based scheduling for layered or MPEG video is studied in [20, 219, 277] and time-based scheduling is studied by Shakkottai and Srikant [256].

Using either priority or deadline (but not both) to schedule packets may not work well in multimedia streaming. It is easy to find situations where earlier-deadline low-priority packets should be sent first, and other situations where less-urgent but higher-priority packets should be sent first. For example, let's consider the case when the pending packets are from a video I-frame with high priority and a playback delay of 1 second, and a B-frame with a lower priority but a very short playback delay. If only priority is used to decide the sending order, we may only send I-frame packets. In contrast, both I-frame and B-frame packets may be sent if packet's time is used. On the other hand, let's recall a common scenario in many applications, e.g., video surveillance, where packets within a frame have the same deadline but different priorities. If only deadline is considered, packets may be sent in a random or round-robin manner, and a high-priority packet can lose its transmission chance to lower-priority packets.

Some works consider a combination of various factors. For example, Chakareski et al. [50] proposes to send packets based on their inter-dependencies and their role in error concealment, with consideration of RTT, loss rate, and bandwidth. For each packet, the relationship with other packets and its role in error concealment indicate its priority. Another interesting work by Krasic et al. [163] proposes translating the user's preferences – temporal or spacial – into the assignment of priority to the packets.

This can also be viewed as information on the application types that is passed to the schedulers.

Ideally, schedulers should be aware of various information from different layers, such as the coding nature of data, the network conditions and the applications involved. The data coding affects not only the bandwidth consumption but also the sending order, which should be partly decided based on the data dependencies given by the coding scheme. Network conditions such as loss rate, RTT may vary over time and greatly affect the successful arrival of packets. In addition, different types of applications generate packets with different statistical properties, the requirements of perceptual quality may be different, therefore packet scheduling policy should be changed accordingly.

However, in many cases, gathering such information and passing it to schedulers across various layers are infeasible. This is especially true when the schedulers reside in a lower layer. Even if they can receive all information, processing such information may become a burden. Thus, it is equally obvious that we may not include all kinds of information in the scheduling process.

Furthermore, schedulers may operate in a heterogeneous environment and may have to simultaneously deal with data from different sources with different coding structures. If scheduling is expected between intermediate network nodes, information about packet abstraction should be compact enough to be stored in packet headers. Hence, it is desirable to use common and simple information in the scheduling process.

We believe that both priority and deadline are important, since they often represent the temporal and hierarchical dependencies within most multimedia data. For example, in our case, packets with user's interested regions are assigned higher priority than others using the content-based prioritizing scheme proposed in Chapter 3. The priority of a packet also reflects the dependency relationship of its macroblocks with macroblocks from other packets. However, knowing the packet's priority does not tell anything about its deadline. Therefore, considering only one of them is not enough: we should take into account both properties while scheduling packets.

One of our focuses is how to use these two properties in scheduling. Should we always send the packet with highest priority first or the packet with earliest deadline first? Or would it be better if both priority and deadline are considered at the same time in making schedule decisions?

Another question is that will additional network information, such as RTT, be helpful for packet scheduling? Packet scheduling decisions are about *when* to send packets[1]. Therefore, knowing the RTT value may help schedulers to answer many important questions, such as how long it would take a packet to reach receivers (e.g., RTT/2), and when a packet may be considered a lost packet (e.g., after an RTT without receiving an ACK).

To answer these questions, we study the performance of five different scheduling schemes. The modified version of the algorithm `FirstFit` [51], presented in Chapter 3, schedules packets primarily based on their priority. The scheduler `Urgent` sends packets based on their deadline first. Meanwhile, the modified version of `GenFlag` [66], namely `GenFlag2` employs both deadline and priority in making schedule decision. Two new scheduling algorithms are designed with RTT consideration. The first scheme, named `GenFlagNet`, is modified from `GenFlag2`. The second, named `EoH`, uses the same RTT consideration like `GenFlagNet` but gives the earliest-deadline packets more chances than `GenFlagNet` and `GenFlag2` do.

Our experiments show that the order in which packet's deadline and priority are used greatly affects the received quality. The difference between `FirstFit` and `Urgent`, in terms of F-PSNR, is around 4–5dB under an average packet loss rate of 2–5% and could be as high as 20–25dB when there is no network loss (queuing loss still occurs).

[1]Packet allocation and protection decisions (such as deciding the number of packets to retransmit or the number of FEC packets), on the other hand, is more related to the loss rate.

Meanwhile, considering both deadline and priority at the same time could improve quality in some scenarios. However, the improvement, compared to the best performer between `FirstFit` and `Urgent`, is not really significant. Surprisingly, taking RTT into consideration does not help much either, e.g., the difference between `GenFlag2` and `GenFlagNet` is only about 1–2dB in most of the cases.

The rest of this chapter is organized as follows. Section 4.2 will briefly describe some related works in scheduling area. The scheduling model and all algorithms are presented in Section 4.3. In Section 4.4, we study the performance of studied schedulers in different scenarios and Section 4.5 concludes the chapter.

4.2 Related works

In this section, we will briefly describe the use of priority and deadline for scheduling in various applications, from job scheduling in operating systems, multimedia streaming to general online scheduling.

Priority-based scheduling has been extensively studied in operating system research. For example, most Unix/Linux operating systems schedule requests based on their dynamic priorities, which are periodically changed according to the requests' status [67, 76]. Similarly, thread schedulers (dispatchers) in Windows NT/2000/CE/XP also serve the request with highest priority first and if there are several requests with the same priority value, a round robin scheme is applied [176,319]. However, one main problem with priority-based schemes is that they are not suitable for applications like data acquisition, robotic control or multimedia streaming, where results are normally wasted if their deadlines are over. To address this problem, various works have been proposed to improve the time-awareness of operating systems [67, 153, 176, 297].

Similarly, each multimedia packet usually has a deadline, beyond which the packet would be useless even if it is successfully received. In addition, many multimedia objects are encoded in a way that different packets have different priorities and the usability of most packets is dependent on others. Therefore, it is natural that both properties should be used in scheduling multimedia packets. However, most current multimedia schedulers are either priority-based [20, 219, 277] or time-based [22, 256].

Some works consider both priority and deadline. In the work by Chakareski et al. [50], the priority of a packet is indicated by its interdependencies with other packets and its role in error concealment. RTT, loss rate and bandwidth are also considered in the scheduling problem. However, the complexity of the solution makes it difficult to apply in online streaming. Krasic et al. proposed to group packets in windows based on time, and then stream them based on their priority [163]. This scheme is, in a way, similar to the idea of `Urgent` algorithm we study here.

In our streaming model (see Figure 1.1 and Figure 3.2), packets may reach schedulers in a disorderly manner due to the probabilistic behavior of the sources of multimedia objects and the network behavior. For example, an already sent packet may be lost and put back into the scheduler's queue. When the number of incoming packets exceeds the system capacity, schedulers have to decide which packets to drop and which packets to serve. In normal applications, all packets are treated in the same way, thus are normally dropped in a random way. In our case, different packet may have different priority, based on its semantic contribution, and different deadline. Therefore, schedulers have to treat each packet differently, and their decision problem should be solved by using online scheduling algorithms.

Online scheduling, since its first introduction by Graham [116] in 1966, has been studied and applied for various applications from distributed computing [6], load balancing [26] to buffer management in QoS networks. Under online setting, schedulers cannot see the entire input instance since requests (or jobs) arrive unpredictably over time and therefore they have to schedule based on the current knowledge.

A tool to measure the worst-case performance of online scheduling algorithms is the notion of *competitive ratio*, introduced by Sleator and Tarjan [267]. An online algorithm is $c - competitive$ if for any input instance, its gain is at least $1/c$ of the optimal gain provided by the off-line algorithm.

Mansour et al. [191] proposed a greedy algorithm as a preemptive queuing policy, in which (i) the size of each request is one byte, and (ii) the buffer is a FIFO (First-In-First-Out) queue. Therefore, their schedule problem becomes a buffer management problem, and the greedy algorithm simply drops the lowest value bytes when an overflow occurs. This greedy algorithm was proved to have a competitive ratio of 4 and then a ratio of 2 by Kesselman et al. [155]. Further improvements of the algorithm were presented by Kesselman et al. [156] and Mahdian et al. [189] to have better competitive ratio of 1.983 and 1.75 respectively.

In 2001, Chang et al. [51] considered a similar problem of scheduling requests with different values, sizes, deadlines and release times. However, the size of requests can be cut down by the scheduler and partially served requests also contribute to the total value of service. He proposed two algorithms, namely `FirstFit` and `EndFit`, and showed that both have a competitive ratio of 2. Later, Chin et al. [60] proposed a deterministic algorithm (`MIX`) with a competitive ratio of $e/(e-1) \approx 1.582$, in which jobs are shared between k processors.

In 2004, Bartal et al. [23] proposed two algorithms to online schedule unit-size jobs with a non-negative real weight (priority) and integer release time and deadline. The first algorithm, called RMix, was a randomized algorithm and was proved to have a competitive ratio of $e/(e-1)$. The deterministic algorithm, namely Edf_α, was proved to have a competitive ratio of $2-2/s+o(1/s)$. It is obvious that this ratio approaches to 2 as s increases. Because of the nature of our problem, we are interested in the s-bounded case where the deadline of each and every request is within s time units of its release time. In fact, if all requests are s-bounded (i.e their deadlines are within s time units of their release times), the schedule problem is equivalent to the buffer management problem with s-bounded delay.

The most recent work that is closely related to ours was shown by Chrobak et al. [66]. In their buffer management problem, packets have unit size, real weight and integral release time and deadline, therefore they can be transmitted only at integer time steps. A deterministic online algorithm called `GenFlag` with a competitive ratio of $64/33 \approx 1.939$ was proposed and was the first deterministic algorithm with a ratio better than 2. We conducted a few experiments and found that in most multimedia streaming scenarios, `GenFlag` was the best performer compared to RMix, Edf, `EndFit`. Therefore, `GenFlag` would be used in this study.

4.3 The scheduling model and algorithms

4.3.1 The scheduling model

We consider a system in which one sender is connected with one receiver via a lossy link. The sender wants to send one or multiple streams of packets to the receiver in a constant transmission rate R. Figure 3.2 shows the diagram of our system.

Packets could be newly generated and/or retransmission packets of the lost ones. Before transmitting into the network link, packets will be queued at the sender buffer. The buffer manager (or scheduler) can drop a packet out of the queue, e.g., when its deadline is over. However, pre-emption, i.e., disrupting the sending of a packet to send another one, is not allowed. Because the scheduler sees each packet as a request to be scheduled, we use the terms packet and request interchangeably hereafter.

Since we want to study the effects of using priority and deadline on scheduling, we assume that the buffer size is large enough so that no queueing packet is dropped due to buffer shortage. In the context of multimedia streaming where packets need to be received after a limited delay (e.g., a few GOPs or seconds), the number of packets within the buffer at anytime is also limited. Therefore, it is reasonable to assume that streaming servers could have enough scheduling buffer for such data. With that, we can eliminate the effects of buffer size on scheduling performance, and focus on what we want to study.

Each packet j is represented by 3 positive real and 1 positive integer values: its release time $st(j)$, deadline $dl(j)$, its priority $v(j)$, and size $sz(j)$ (integer, in bytes). Note that the size and release time of a packet are implicit information and always available to the scheduler. However, the deadline and priority are needed to be passed to the scheduler. The *span* of a packet j is the time interval $[st(j), dl(j)]$. The span corresponds to the playout delay of the data stream, which has different values for different types of data and applications. For example, the playout delay of video streaming applications should be less than 10s, while the delay of two-way conversational videos should be less than 400ms [134].

At each sending opportunity, say at time t, the scheduler **A** will decide which packet should be sent. Only one packet can be sent at any time t. Packet j is *pending* in the scheduler at t if $st(j) < t < dl(j)$ and j has not been sent. If its deadline $dl(j) > t$, packet j will be removed from the buffer of pending packets. The aim of a scheduler is to maximize the expected received quality.

Note that the transmission links may be lossy and may have time-vary delay, thus a packet may be lost or may reach the receiver later than its deadline. If the sender does not receive the acknowledgement of the packet j from the receiver after a certain windows time W, it will put the packet j into the buffer again if $dl(j) \leq t$. The window time W could be a pre-defined value or equal to the network RTT, which could be easily estimated, e.g., by using `ping`, RTCP protocol (see Section 1.3.2.2).

To facilitate the descriptions of followed scheduling algorithms, we denote Q_t the set of pending packets within the scheduler's queue at the sending opportunity t, h_t the highest-priority packet of Q_t, and e_t the earliest-deadline packet of Q_t.

4.3.2 Scheduler FirstFit – Highest-priority first

The original algorithm FirstFit [51] only uses priority, and no other information, to schedule packets. Particularly, it always serves the pending packet with highest priority. In this section, we modify FirstFit to take packet's deadline and size into the consideration.

Our modified version of FirstFit, called in short FirstFit here, sends packets with highest priority first. Pending packets are ordered according to their priority, and the scheduler sends the packet with highest priority first. In case $v(i) = v(j)$, the tie-breaking rule is to favor the packet with an earlier deadline, so the earliest-deadline packet among these two will be sent first. If $v(i) = v(j)$ and $dl(i) = dl(j)$, the packet with a larger size will be chosen. This is because under the same coding scheme and coding parameters, with the same priority and deadline, intuitively packets with more data would contribute more than packets with smaller size. The algorithm is presented as follows.

At every sending opportunity t, do the following.

1. Update the set of pending packets Q_t (remove packets with deadline later than t and add new coming packets).
2. Schedule the highest-priority packet h_t using the tie-breaking rule.

4.3.3 Scheduler Urgent – Earliest-deadline first

The algorithm Urgent always sends the pending packet with earliest deadline first. In case $dl(i) = dl(j)$, the tie-breaking rule is to send the packet with a higher priority. If $dl(i) = dl(j)$ and $v(i) = v(j)$, the packet with larger size will be chosen. The algorithm is shown in details as follows.

At every sending opportunity t, do the following.

1. Update the set of pending packets Q_t.

2. Schedule the earliest-deadline packet e_t (use tie-breaking rule if necessary).

4.3.4 Scheduler GenFlag2 – Priority and deadline

Proposed by Chrobak to solve a buffer management problem [66], the deterministic algorithm `GenFlag` assumes packets have unit size, real weight and integral release time and deadline, thus can be transmitted only at integer time. It is proven to have a competitive ratio of 64/33 with $\alpha = 7/11$ and $\beta = 8/11$.

In essence, at every sending opportunity t, `GenFlag` considers only a subset $Q_t^{'}$ of the set of pending packets Q_t. This subset $Q_t^{'}$ consists of those packets with priority larger than a certain threshold. The earliest-deadline packet $e_t^{'}$ within this subset and the highest-priority packet h_t are then considered to be sent. Generally, `GenFlag` would alternatively schedule $e_t^{'}$ (when $eFlag = false$) and h_t (when $eFlag = true$). However it puts more favor on the earlier packet $e_t^{'}$ by applying the following two rules: (i) If $eFlag = false$, packet $e_t^{'}$ of current $Q_t^{'}$ will be sent. However if $e_t^{'} = h_t$ then at the next time $t^{'}$, packet $e_{t'}^{'}$ will be sent, (ii) If $eFlag = true$ (i.e., the highest-priority packet h_t should be sent) but $e_t^{'}$ has a certain priority and urgent deadline then it will be sent instead.

To apply `GenFlag` in multimedia streaming scenarios where packets may have different sizes, step 4 of the original algorithm is slightly changed. The modified algorithm, named `GenFlag2`, is shown below.

Set $eFlag = false$.

At every sending opportunity t, do the following.

1. Update the set of pending packets Q_t.

2. Find the highest-priority packet h_t (use tie-breaking rule in Section 4.3.2 if necessary).

3. Find the earliest-deadline packet $e_t^{'}$ among the packets j whose $v(j) \geq \alpha v(h_t)$ (use tie-breaking rule in Section 4.3.3 if necessary).

4. If $eFlag = false$

    then schedule $e_t^{'}$

    If $e_t^{'} \neq h_t$ then set $eFlag = true$

    Else

    Set $eFlag = false$

    If $[t + sz(e_t^{'})/R \leq dl(e_t^{'}) < t + 2sz(e_t^{'})/R]$ and $[v(e_t^{'}) \geq \beta v(h_t)]$

        then schedule $e_t^{'}$

    Else schedule h_t

In step 4, the original condition for unit-size packet $[dl(e_t^{'}) = t + 1]$ is changed to $[t + sz(e_t^{'})/R \leq dl(e_t^{'}) < t + 2sz(e_t^{'})/R]$, where $sz(e_t^{'})/R$ is the time to send packet $e_t^{'}$. Because the original algorithm assumes all packets having unit size, the condition $[dl(e_t^{'}) = t+1]$ means that there is precisely enough time to send packet $e_t^{'}$. However in our case, $sz(e_t^{'})/R$, $dl(e_t^{'})$ and t have real values (not integral values anymore), if we simply replace $[dl(e_t^{'}) = t+1]$ by $[dl(e_s) = t + sz(e_t^{'})/R + 1]$, the latter condition may never be satisfied. The meaning of the modified condition $[t + sz(e_t^{'})/R \leq dl(e_t^{'}) < t + 2sz(e_t^{'})/R]$ is still the same as that of the original condition, i.e., if $e_t^{'}$ is not sent now, there will be no other chance to send it later.

4.3.5 Scheduler EoH – Earliest or Highest, and RTT

If networks are lossy, some packets may be lost. When a packet's loss is discovered, e.g., no acknowledgement received after the window period W, the lost packet will be put back to the scheduler's pending queue. So, one of our questions is that "Would schedulers perform better if they took this window period W into their consideration?"

The main idea of our scheduler `EoH` is to ensure that each high-priority packet will have a few opportunities to be resent (say K times). Thus, they have to be sent well before their deadline. Besides, we also want to make sure that the earliest-deadline packet e_t of the current pending set Q_t, regardless of its priority, will always has a chance to be considered for sending. The details of `EoH` are presented below.

Set K the number of times that the scheduler wishes to send the highest-priority packet h_t.

At every sending opportunity, say at the time t, do the following.

1. Update the set of pending packets Q_t (remove packets with deadline later than t and add new coming packets).

2. Find the highest-priority packet h_t (use tie-breaking rule in Section 4.3.2 if necessary).

3. Find the earliest-deadline packet e_t. (use tie-breaking rule in Section 4.3.3 if necessary).

4. If $\left(t + \frac{sz(e_t)}{R}\right) < dl(h_t) - (K-1)\left[W + \frac{sz(h_t)}{R}\right]$

    then schedule e_t

    Else

    schedule h_t

Step 4 can be interpreted as follows: At the time t, the scheduler will send either the earliest-deadline packet e_t or the highest-priority h_t of the set of current pending packets Q_t. If it knows that by sending e_t, there will not be enough time to send h_t at most K times, h_t will be sent. Otherwise, the earliest-deadline packet e_t is sent. So in the long term, high-priority packets are more favorable while in the short term, the earliest-deadline packet is given higher precedence by `EoH`.

If the packet loss probability is p, and we want the packet to be successfully received with a probability not less than a threshold p_0, then we could send the packet K time, where $1 - p^K \geq p_0 \Longrightarrow K \geq \frac{ln(1-p_0)}{ln(p)}$. Therefore, the value of K can be determined as in Equation 4.1.

$$K = \begin{cases} 1 & \text{if p = 0} \\ \left\lceil \frac{ln(1-p_0)}{ln(p)} \right\rceil & \text{otherwise} \end{cases} \tag{4.1}$$

We can either predefine K, p_0 or adaptively change their values based on the network conditions and the required quality of services. For example, if the required threshold probability is $p_0 = 99\%$, K should be equal 2 if the network probability of loss $p = 10\%$ or 3 if $p = 20\%$. In practice, the loss rate is normally less than 10% so $K = 2$ is often sufficient for most scenarios. This is the value of K used in this study.

4.3.6 Scheduler GenFlagNet – GenFlag2 and RTT

While `FirstFit` and `Urgent` always send highest-priority packet h_t and earliest-deadline e_t first, correspondingly, `GenFlag2` and `EoH` considers packet's priority and deadline at the same time. At each sending time t, both schedulers choose between the highest-priority packet and the earliest-deadline packet within their set of candidate packets.

For `GenFlag2`, the packets considered by `GenFlag2` must have a priority larger than a certain threshold. This set $Q_t^{'}$ is a subset of Q_t, the set of all pending packets at the sending time t. Therefore, the highest-priority packet h_t is the same for both

$Q_t^{'}$ and Q_t. However, the earliest-deadline packet $e_t^{'}$ in $Q_t^{'}$ may not be the packet with earliest deadline among all pending packets. Meanwhile, `EoH` always considers h_t together with the earliest-deadline packet e_t of the set Q_t, regardless of its priority.

Since they choose their "earliest-deadline" packets from different sets, this fact – instead of the inclusion of window period W in `EoH` – may be the reason for the possible performance difference between `EoH` and `GenFlag2`. To study the effects of this difference, we modify step 4 of `GenFlag2` by replacing the old condition: $[t + sz(e_t^{'})/R \leq dl(e_t^{'}) < t + 2sz(e_t^{'})/R]$ with a new condition using in scheduler `EoH`: $\left(t + \frac{sz(e_t^{'})}{R}\right) < dl(h_t) - (K-1)\left[W + \frac{sz(h_t)}{R}\right]$. We call this modified algorithm `GenFlagNet`.

4.4 Experiments

Our experiments are aimed to answer the following questions, in the context of streaming content-based prioritized videos. First, will sending highest-priority packet first be better than sending earliest-deadline packet first? The answer for this question could be found by comparing the performance of `FirstFit` and `Urgent`. Second, would it be better to consider both priority and deadline at the same time, instead of "priority first, then deadline", or vice versa? We try to answer this by comparing between `FirstFit`, `Urgent` and `GenFlag2`. Third, if network information such as RTT is considered in making scheduling decisions, will it improve the received quality? Performance comparison between `GenFlag2` and `GenFlagNet`, `EoH` leads us to an interesting observation, which will be described later in this section.

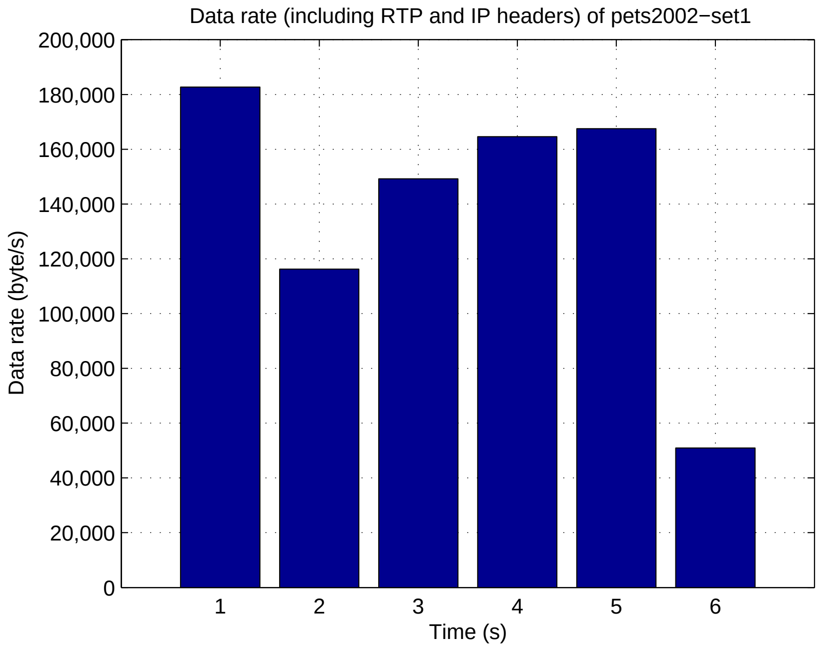
4.4.1 Test data and experiment settings

Two video segments extracted from surveillance videos in PETS benchmark datasets, `people_test_dataset1.mpg` (640x240 pixels) [212] and `Walk1.mpg` (384x288 pixels) [213], are used in our experiments. The first video, named `pets2002-set1.mpg`, consists of 142 frames and the second video, named `Walk1-man.mpg`, consists of 200 frames. Both are encoded in MPEG-1 IPPP format (one I-frame followed by 11 P-frames in one GOP) with frame rate of 25 fps.

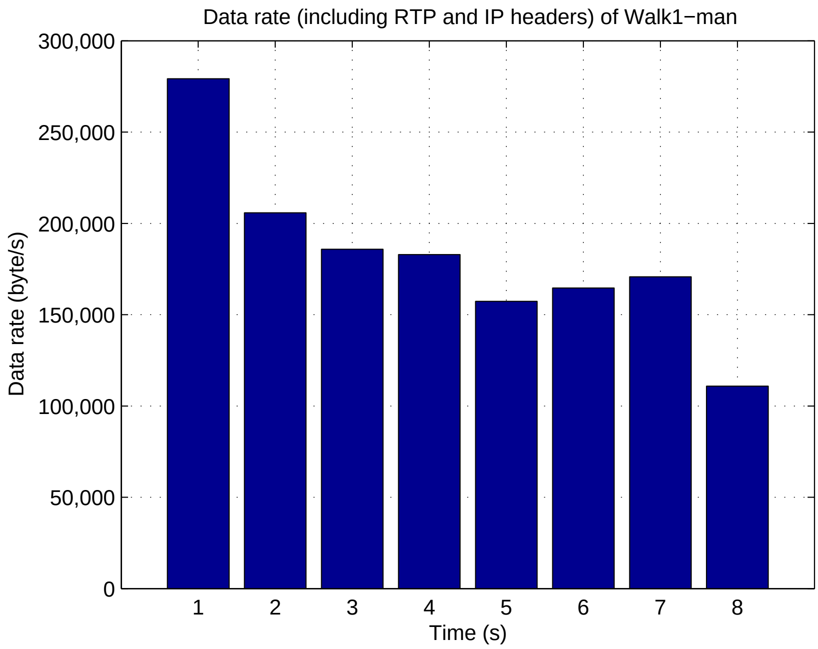
They are prioritized and packetized using our content-based prioritizing scheme `[Blob + SEQ]` presented in Chapter 3. This scheme prioritizes video macroblocks, slices and packets based on the semantic content they carry, the dependency relationships between macroblocks, and the syntax data in each packet. The packetizing process, followed the recommendations in RFC2250 [122], assures that each packet contains only data from a single frame and that its size is within the boundary of the MTU limit. After packetization, the average data rate (including RTP and IP headers) of `pets2002-set1.mpg` is about 124000 bytes/s, and that of `Walk1-man.mpg` is about 172000 bytes/s. Figure 4.1 shows the actual data rate of the two videos.

Packet loss is due to two reasons: *network loss* during transmission and *queue loss* at the scheduler's queue. To simulate network loss, a Markov 2-state model is used. If the network state is G (Good) then the packet is considered to be successfully received; if it is B (Bad) then the packet is either late or corrupted. The success arrival of a packet is generated by the Markov model with the transition matrix $[1 - p_{GB}, p_{GB}; p_{BG}, 1 - p_{BG}]$, where $p_{GB} = 0.05$, $p_{BG} = p_{GB}(1 - p)/p$ where p is the average network loss rate. In our experiments, p is varied from 0% to 10%. Queue loss is influenced by various complicated factors (e.g., how videos are prioritized, which scheduler is used, how bad the network loss is, how long the RTT is, the number of packets in current queue, their properties), therefore is not modelled.

The window period W is set to be equal RTT, which has two values of 100ms and 200ms in our experiments. These values reflect the average RTT measured on the Internet [241]. The Maximum Transmission Unit (MTU) value is 1500 bytes. RTP packets are assumed to be sent over IPv4 networks, with IP header of 20 bytes.



(a) pets2002-set1



(b) Walk1-man

Figure 4.1: Data rate (including RTP header and IP header) of the two videos – (a) pets2002-set1, (b) Walk1-man – after being prioritized and packetized by our content-based prioritizing scheme.

Videos are streamed at different transmission rates, varied from 80% to 140% of their average data rate.

4.4.2 Experimental results

The performance of each scheduler is measured by the video quality received. For each frame of the output video, we calculate the PSNR for the whole frame and the F-PSNR for its interest regions (see Chapter 3). Our experiment is repeated 15 times for each network configuration (average loss rate, bandwidth, RTT). The report values are the average PSNR and F-PSNR of 15 runs.

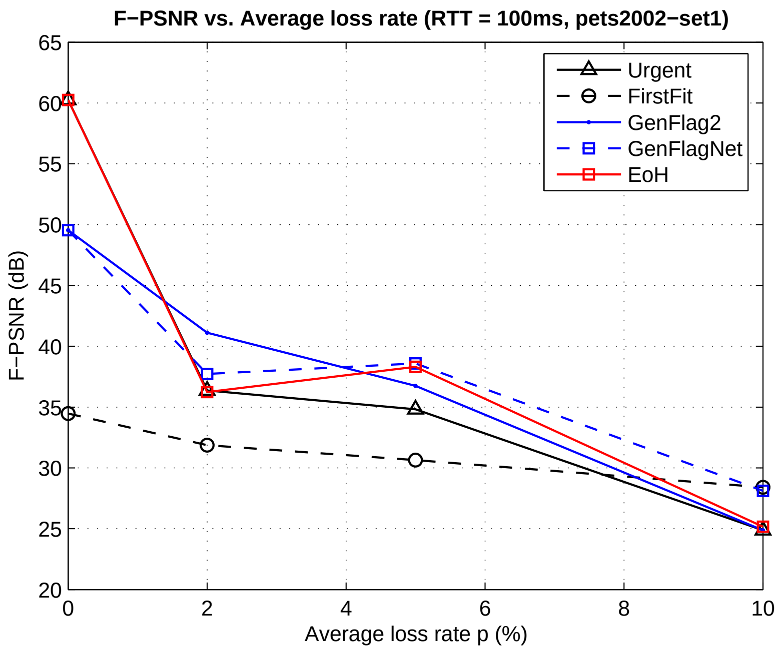
4.4.2.1 FirstFit vs. Urgent

Figure 4.2 and Figure 4.3 show the average F-PSNR of the two videos when they are streamed by different schedulers under different network conditions, at their average data rates. PSNR results are shown in Figure 4.4 and Figure 4.5. Average and standard deviations of PSNR and F-PSNR measurements for all schedulers while streaming two videos are summarized in Table 4.1 and Table 4.2.

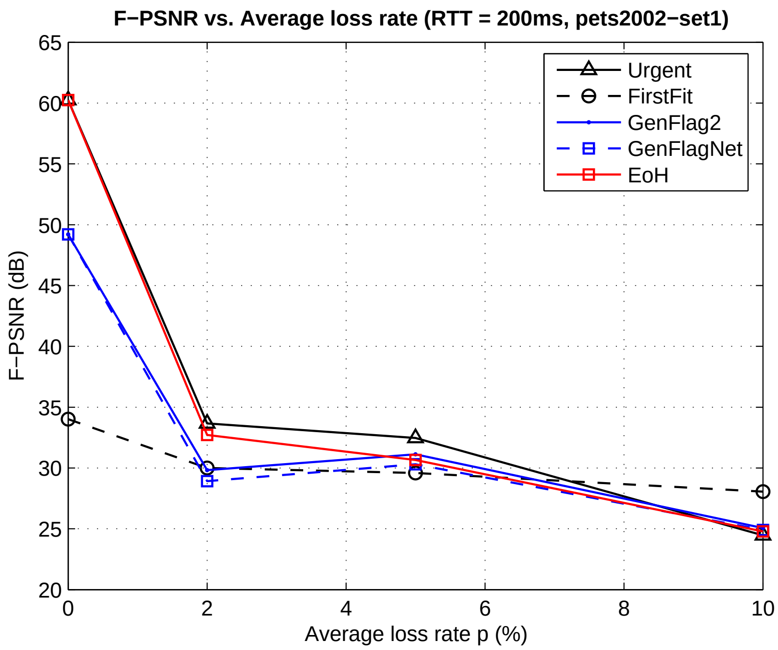
At a glance, Figure 4.2 and Figure 4.3 show that `FirstFit` and `Urgent` behave differently for the two videos. For example, `Urgent` performance looks better than `FirstFit` performance for `pets2002-set1.mpg`, and inversely for `Walk1-man.mpg`.

However, at 0% network loss, `Urgent` always significantly outperforms `FirstFit` for both videos, under different RTT values. For `pets2002-set1.mpg`, `Urgent` can achieve an average F-PSNR of around 60dB, while `FirstFit` can only produce an average F-PSNR of around 34dB. For `Walk1-man.mpg`, the average F-PSNR values are around 41dB with `Urgent` and 35dB with `FirstFit`. That is, the F-PSNR difference is about 6–26dB in this case.

Because there is no network loss, the only reason to prevent a packet from reaching the receiver is queue loss – being dropped from scheduler's queue because its deadline

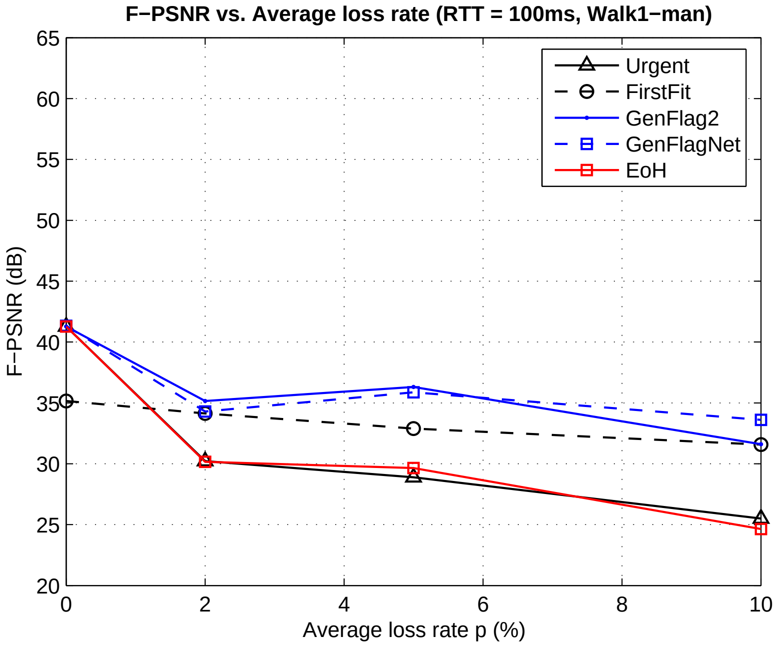


(a) RTT = 100ms

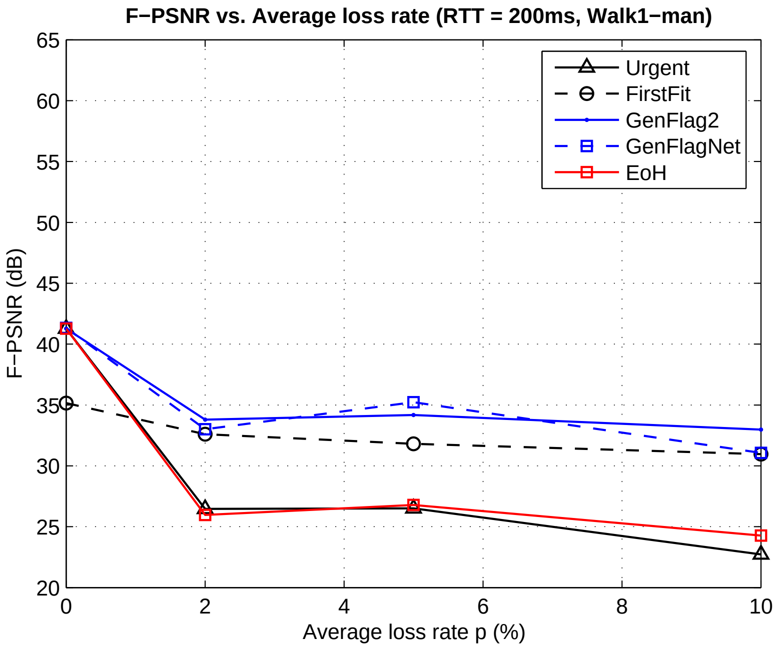


(b) RTT = 200ms

Figure 4.2: Average F-PSNR vs. Average network loss rate. Video pets2002-set1 is streamed with a bandwidth equal to its average data rate, under different average network loss rates and different RTT values: (a) RTT = 100ms, (b) RTT = 200ms.



(a) RTT = 100ms



(b) RTT = 200ms

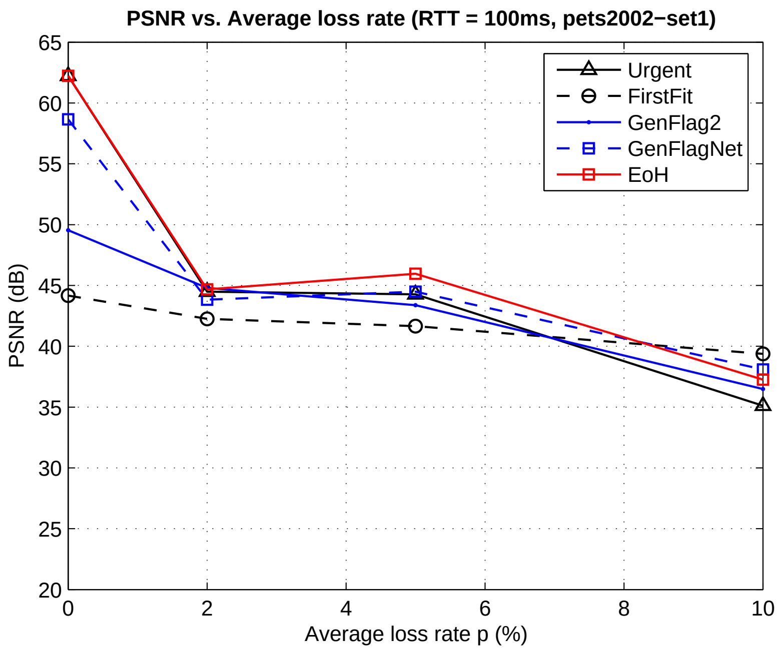
Figure 4.3: Average F-PSNR vs. Average network loss rate. Video Walk1-man is streamed with a bandwidth equal to its average data rate, under different average network loss rates and different RTT values: (a) RTT = 100ms, (b) RTT = 200ms.

is over. By scheduling packets based on their deadline, starting from the earliest, `Urgent` may send most packets across. However, `FirstFit` schedules all high-priority packets within the queue first, so some lesser-priority earlier-deadline packets may have their deadline expired (thus be dropped) well before all the high-priority packets are sent.

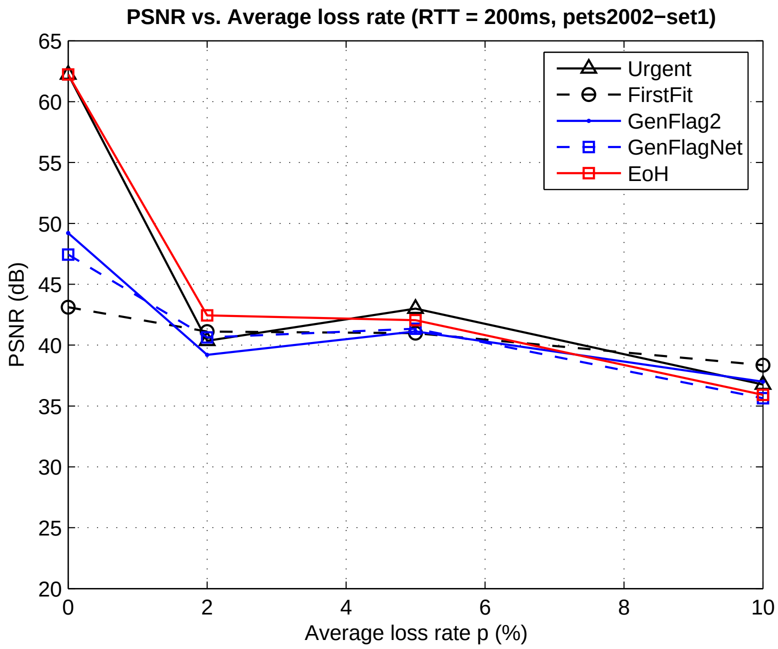
At 10% network loss rate, `FirstFit` is always better than `Urgent` for both videos, under different RTT values. The F-PSNR difference is about 4dB for `pets2002-set1.mpg` and 6-8dB for `Walk1-man.mpg`. This is because under a higher network loss rate and an average transmission rate, sending earliest-deadline packets first is no longer effective. Since these packets may have little priority – or quality contribution in our case, high-priority packets may be dropped out of the queue when `Urgent` keeps sending and re-sending earlier-deadline packets. Inversely, `FirstFit` policy almost assures the successful receipt of, at least, a certain number of essential packets. Therefore, as the loss rate increases, `Urgent` performance rapidly decreases while `FirstFit` performance is rather stable. That explains why after a network loss rate threshold, `FirstFit` starts to outperform `Urgent`.

Similar phenomenons could be observed from Figure 4.4 and Figure 4.5, where PSNR results are shown. At zero network loss, `FirstFit` performance is better than `Urgent` performance by about 19dB for `pets2002-set1.mpg`, and about 5dB for `Walk1-man.mpg`. At 10% network loss rate, the difference is much less significant. The performance of `FirstFit` is better than that of `Urgent` by about 2-4dB for `pets2002-set1.mpg`, and less than 1dB for `Walk1-man.mpg`.

It is obvious that neither the highest-priority-first policy (`FirstFit`) nor the earliest-deadline-first policy (`Urgent`) is good for all situations. Protecting high-priority packets is better when network loss rate is high, but mediocre in other cases. Sending earliest-deadline first works best at no network loss (or when transmission rate is high enough to cover the network loss), but the performance is rapidly de-

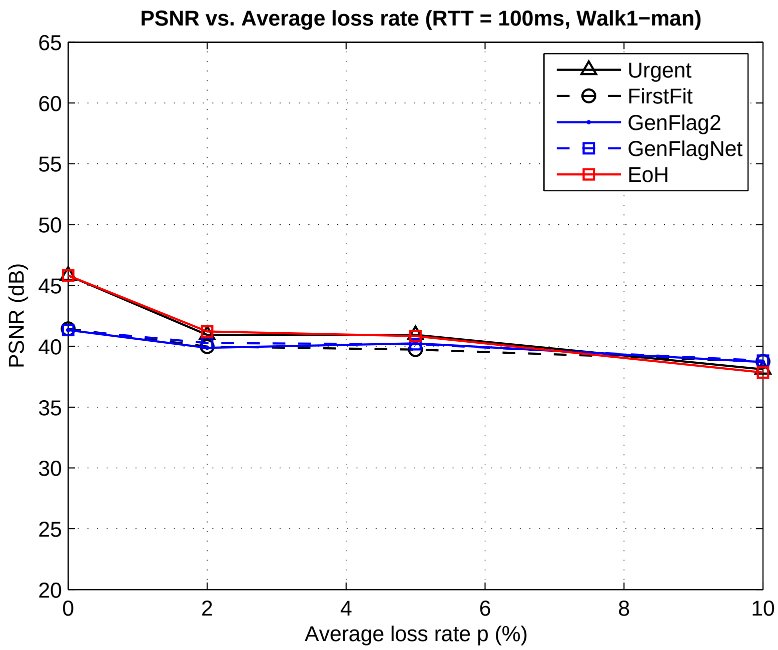


(a) RTT = 100ms

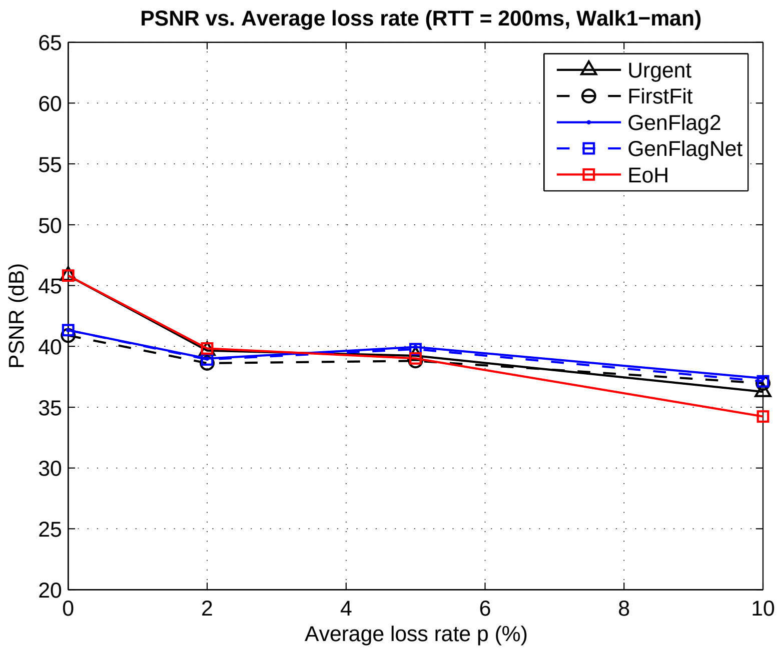


(b) RTT = 200ms

Figure 4.4: Average PSNR vs. Average network loss rate. Video pets2002-set1 is streamed with a bandwidth equal to its average data rate, under different average network rates and different RTT values: (a) RTT = 100ms, (b) RTT = 200ms.



(a) RTT = 100ms



(b) RTT = 200ms

Figure 4.5: Average PSNR vs. Average network loss rate. Video Walk1-man is streamed with a bandwidth equal to its average data rate, under different network loss rates and different RTT values: (a) RTT = 100ms, (b) RTT = 200ms.

creased when loss increases. In the next section, we will consider both priority and deadline at the same time, in the hope that it may offer a better solution.

Network loss rate	Scheduler	PSNR$_a \pm \sigma$		F-PSNR$_a \pm \sigma$	
		RTT=100ms	RTT=200ms	RTT=100ms	RTT=200ms
0%	Urgent	62.23 ± 0.00	62.23 ± 0.00	60.24 ± 0.00	60.24 ± 0.00
	FirstFit	44.17 ± 0.00	43.11 ± 0.00	34.46 ± 0.00	34.01 ± 0.00
	GenFlag2	58.65 ± 0.00	47.44 ± 0.00	49.54 ± 0.00	49.20 ± 0.00
	GenFlagNet	58.65 ± 0.00	47.44 ± 0.00	49.54 ± 0.00	49.20 ± 0.00
	EoH	62.23 ± 0.00	62.23 ± 0.00	60.24 ± 0.00	60.24 ± 0.00
2%	Urgent	44.48 ± 3.66	40.36 ± 7.08	36.37 ± 8.52	33.67 ± 9.22
	FirstFit	42.25 ± 0.84	41.11 ± 1.15	31.87 ± 1.63	30.00 ± 2.11
	GenFlag2	44.79 ± 3.14	39.20 ± 3.90	41.11 ± 8.33	29.82 ± 4.96
	GenFlagNet	43.83 ± 3.31	40.63 ± 1.79	37.72 ± 9.84	28.92 ± 2.76
	EoH	44.68 ± 4.85	42.44 ± 3.02	36.23 ± 9.47	32.72 ± 6.73
5%	Urgent	44.27 ± 6.16	43.00 ± 5.22	34.82 ± 10.44	32.47 ± 9.18
	FirstFit	41.66 ± 0.66	40.98 ± 1.07	30.64 ± 1.42	29.59 ± 2.33
	GenFlag2	43.37 ± 3.24	41.13 ± 3.11	36.74 ± 9.54	31.12 ± 6.00
	GenFlagNet	44.48 ± 3.32	44.34 ± 2.48	38.58 ± 8.81	30.30 ± 3.93
	EoH	45.96 ± 3.83	42.05 ± 3.93	38.31 ± 9.55	30.66 ± 5.03
10%	Urgent	35.11 ± 3.84	36.76 ± 1.48	24.87 ± 2.57	24.46 ± 1.83
	FirstFit	39.37 ± 0.76	38.35 ± 2.51	28.40 ± 1.40	28.05 ± 2.23
	GenFlag2	36.48 ± 2.47	37.00 ± 1.48	24.87 ± 2.61	25.06 ± 2.21
	GenFlagNet	38.09 ± 2.80	35.63 ± 3.95	28.11 ± 4.75	24.90 ± 4.14
	EoH	37.25 ± 2.13	35.91 ± 2.91	25.16 ± 2.66	24.79 ± 2.18

Table 4.1: Average and standard deviations of F-PSNR and PSNR measurements from 15 running times. Video pets2002-set1 is streamed with a bandwidth equal to its average data rate, under different network loss rates and different RTT.

4.4.2.2 GenFlag2 vs. FirstFit and Urgent

From Figure 4.2 and Figure 4.3, we can see that at zero network loss, `GenFlag2` F-PSNR performance is always higher than that of `FirstFit`. However, its results are always less than or equal to that of `Urgent`, which is the best performer in this case. Particularly, it outperforms `FirstFit` by around 15dB for `pets2002-set1.mpg`, and 6dB for `Walk1-man.mpg`. Correspondingly, its F-PSNR result is about 10dB lower than `Urgent`'s results for the first video, but they are almost the same for the second video.

Network loss rate	Scheduler	PSNR$_a \pm \sigma$		F-PSNR$_a \pm \sigma$	
		RTT=100ms	RTT=200ms	RTT=100ms	RTT=200ms
0%	Urgent	45.82 ± 0.00	45.82 ± 0.00	41.28 ± 0.00	41.28 ± 0.00
	FirstFit	44.43 ± 0.00	40.89 ± 0.00	35.15 ± 0.00	35.15 ± 0.00
	GenFlag2	41.33 ± 0.00	41.33 ± 0.00	41.28 ± 0.00	41.28 ± 0.00
	GenFlagNet	41.33 ± 0.00	41.33 ± 0.00	41.28 ± 0.00	41.28 ± 0.00
	EoH	45.82 ± 0.00	45.82 ± 0.00	41.28 ± 0.00	41.28 ± 0.00
2%	Urgent	40.93 ± 1.01	39.65 ± 1.16	30.22 ± 4.04	26.47 ± 2.03
	FirstFit	39.97 ± 0.52	38.61 ± 3.16	34.13 ± 2.46	32.60 ± 2.38
	GenFlag2	39.87 ± 0.63	39.00 ± 2.93	35.16 ± 5.13	33.80 ± 4.10
	GenFlagNet	40.27 ± 0.51	38.92 ± 2.09	34.29 ± 3.67	33.00 ± 3.14
	EoH	41.22 ± 0.88	39.82 ± 0.80	30.16 ± 3.19	25.96 ± 1.53
5%	Urgent	40.94 ± 1.05	39.23 ± 2.57	28.89 ± 3.59	26.51 ± 2.54
	FirstFit	39.72 ± 0.34	38.80 ± 2.32	32.89 ± 2.97	31.81 ± 2.36
	GenFlag2	40.23 ± 0.46	39.94 ± 0.46	36.30 ± 4.68	34.18 ± 3.93
	GenFlagNet	40.17 ± 0.35	39.77 ± 0.66	35.87 ± 3.96	35.23 ± 5.10
	EoH	40.83 ± 0.95	39.01 ± 3.76	29.64 ± 3.00	26.78 ± 2.39
10%	Urgent	38.10 ± 1.78	36.27 ± 3.03	25.50 ± 4.79	22.73 ± 1.87
	FirstFit	38.75 ± 0.56	36.96 ± 2.82	31.58 ± 2.24	30.96 ± 2.84
	GenFlag2	38.70 ± 0.80	37.38 ± 3.29	31.61 ± 3.68	32.98 ± 5.08
	GenFlagNet	38.82 ± 0.57	37.13 ± 4.11	33.61 ± 4.62	31.06 ± 3.50
	EoH	37.84 ± 1.52	34.23 ± 5.61	24.64 ± 2.16	24.27 ± 1.93

Table 4.2: Average and standard deviations of F-PSNR and PSNR measurements from 15 running times. Video Walk1-man is streamed with a bandwidth equal to its average data rate, under different network loss rates and different RTT.

At other network loss rates, `GenFlag2` offers a better F-PSNR result than the best performer between `FirstFit` and `Urgent` does, except for `pets2002-set1.mpg` at RTT of 200ms (see Table 4.1). However, its F-PSNR improvement is not much significant (considering its standard deviations): it is only around 2dB for most scenarios and goes up to 4dB in some cases.

The PSNR performance of `GenFlag2`, shown in Figure 4.4 and Figure 4.5, also has the same trend as its F-PSNR performance. One exception is that for `Walk1-man.mpg`, `GenFlag2` produces very similar results compared to `FirstFit`, which is slightly lower than the best performer `Urgent`.

Overall, `GenFlag2` may not be the best performer in all scenarios, but it provides a good performance for all videos in study. For example, `Urgent` has the best results at zero network loss rate and `FirstFit` works best at higher loss rate, e.g., 10% network loss rate. However, at other loss rates, `Urgent` is not good for `Walk1-man.mpg` and `FirstFit` is not good for `pets2002-set1.mpg`. Meanwhile, results from `GenFlag2` are reasonable good, i.e., approximately or better than the best results from `Urgent` and `FirstFit` in all situations.

4.4.2.3 GenFlag2 vs. GenFlagNet vs. EoH

While `EoH` and `GenFlag2`, `GenFlagNet` consider both deadline and priority at the same time, the ways they do are different.

The first difference is in the way packets are selected at each sending time t. For `GenFlag2` and `GenFlagNet`, only packets whose priority is above a limit are selected. Then, they alternatively send the highest-priority packet $h_t^{'}$ and the earliest-deadline packet $e_t^{'}$ of this set $Q_t^{'}$. Meanwhile, `EoH` chooses between the highest-priority packet h_t and the earliest-deadline e_t among all the current packets in queue Q_t. While $h_t^{'}$ and h_t are the same, e_t always has more urgent deadline than, or at least the same deadline as, the earliest-deadline packet $e_t^{'}$ in `GenFlag2` and `GenFlagNet`. Since urgent deadline packets are given more chances by `EoH`, we expect it may have better result in some situations.

The second difference among them is in deciding which packet is to be sent. In `GenFlag2`, the packet $e_t^{'}$ can seize the highest-priority packet's turn, if its deadline is urgent and its priority is higher than a specific threshold. This happens without considering deadline of the highest-priority packet, and therefore, the highest-priority packet may lose its chance to be sent if its deadline is just right after that of the earliest-deadline packet $e_t^{'}$. In `GenFlagNet` and `EoH`, we replace this seizing condition, so that e_t will only be sent if the highest-priority packet can be sent before its deadline by a time of $K \times W$. This is where the RTT consideration comes in ($W = RTT$).

To study the effects of considering RTT in making schedule decisions, we compare the performance of `GenFlag2` with that of `GenFlagNet`, which is modified from `GenFlag2` with RTT consideration. All Figure 4.2, Figure 4.3, Figure 4.4 and Figure 4.5 show that both schedulers have very similar performance. The difference between them, in F-PSNR and PSNR, is only around 1dB in most of time. That is, including RTT here does not significantly change the performance.

To study how packet selection affects the final quality, we compare `GenFlagNet` with `EoH` (see Table 4.1 and Table 4.2). At zero network loss, `EoH` – like `Urgent` – is always better than `GenFlagNet`. At other loss rates, `GenFlagNet` often achieves higher results than `EoH` in terms of F-PSNR (especially for `Walk1-man.mpg`), while `EoH` is slightly better than `GenFlagNet` in terms of PSNR. These results are expected, since `GenFlagNet` focuses on sending more high-priority packets, thus may improve F-PSNR value (quality of interest regions). Meanwhile, `EoH` tries to send both highest-priority and earliest-deadline packets, thus may have better PSNR by reducing loss and error propagation for whole frames. However, since the quality of interest regions – indicated by F-PSNR value – is more important to users for these videos, `GenFlagNet` offers better results to users by protecting high-priority packets.

Talking about users' perception, note that at zero network loss rate, although `Urgent` and `EoH` always provide the best results, what we obtain from `GenFlag2` and `GenFlagNet` are only 4dB lower or even the same. For example, for `pets2002-set1.mpg`, `Urgent` and `EoH` achieve 62.23dB in PSNR and 60.24dB in F-PSNR; while `GenFlag2` and `GenFlagNet` get 58.65dB in PSNR and 49.54dB in F-PSNR. For `Walk1-man.mpg`, the first two schedulers achieve 45.82dB in PSNR and 41.28dB in F-PSNR; while `GenFlag2` and `GenFlagNet` get around 41.33dB in both PSNR and F-PSNR. Obviously at this level, the video quality received from the four schedulers is already very high, thus a normal viewer will not able to tell the difference in perception quality.

By carefully examining the results in Table 4.1 and Table 4.2 with the above observation, we may conclude that for normal network loss rate (less than 10%), GenFlag2 and GenFlagNet are the most suitable choice. At higher network loss rate, e.g., 10%, using FirstFit is a better option.

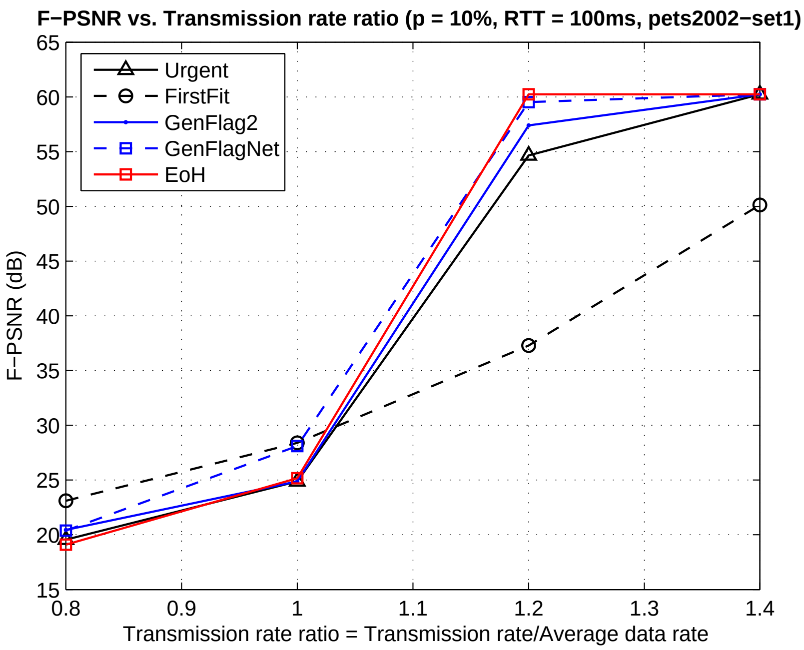
4.4.2.4 Further discussion

Figure 4.6, Figure 4.7, Figure 4.8, Figure 4.9, Table 4.3 and Table 4.4 show what happens when videos are streamed at various bandwidth, under an average network loss rate of 10%. If the bandwidth is lower than average data rate, the effects of network loss rate are expected to be worse. Inversely, sending with a bandwidth higher than average data rate may reduce the influence of network loss.

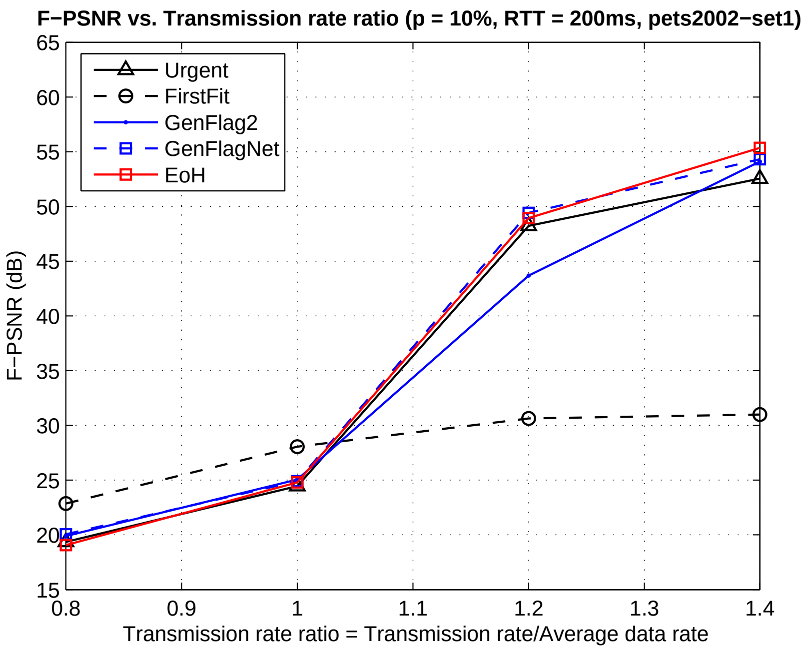
As expected, when the transmission rate is not enough (only 80% of the average data rate), FirstFit is the best performer. However when more bandwidth is available (e.g, the transmission rate ratio is 1.2 or 1.4), the video quality received from other schedulers is substantially improved. Though EoH and GenFlag2 seem to be better than others in various cases, we may say that the performances of Urgent, EoH, GenFlag2 and GenFlagNet are not significantly different from each other's, both quantitatively and qualitatively.

In summary, sending highest-priority first (FirstFit) normally provides the best results in bad network conditions (e.g., high network loss rate, low bandwidth). In normal conditions, where network loss rate is less than 10% and allocated bandwidth is around average data rate, it is better to consider priority and deadline at the same time in the way GenFlag2 does. Including RTT in this scenario does not help much in improving the received quality (GenFlagNet).

However, when bandwidth is high enough to cover network loss, including RTT and considering both the highest-priority packet and the earliest-deadline packet (EoH) will generally provide the best results in terms of dB. Nevertheless, the improvement upon an already-good video quality (e.g., 35–43dB) may not be obviously perceivable by normal viewers, thus reduce the needs to take RTT in consideration.

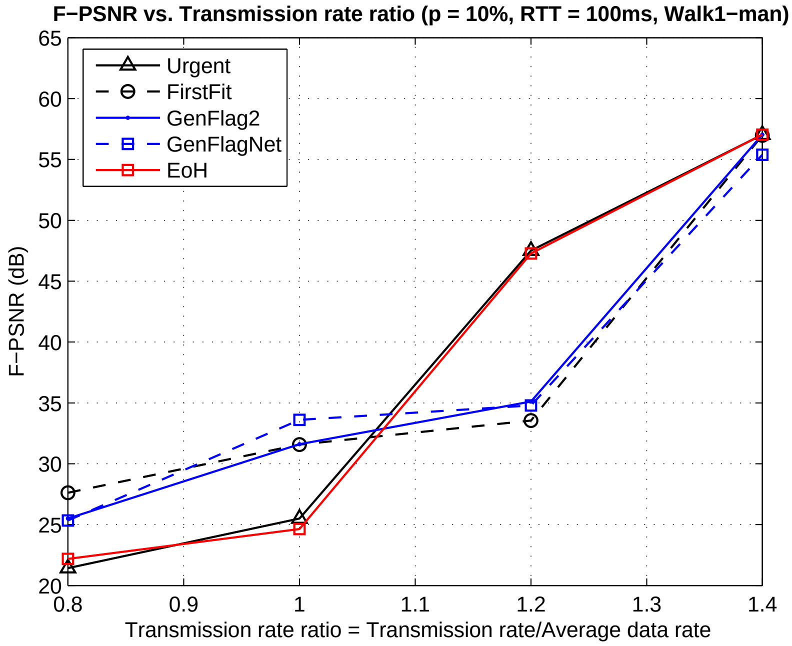


(a) RTT = 100ms

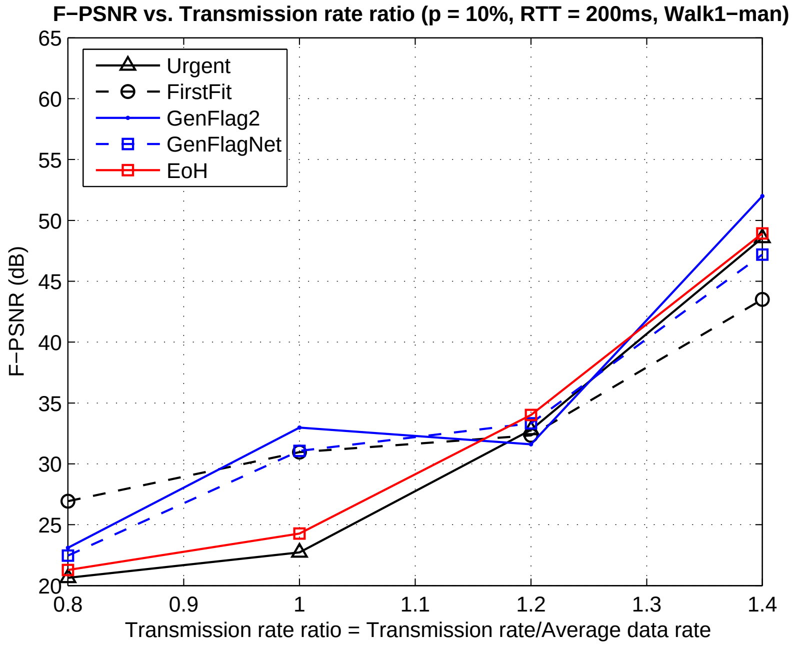


(b) RTT = 200ms

Figure 4.6: Average F-PSNR vs. Transmission rate ratio. Video pets2002-set1 is streamed with various transmission rates, under 10% network loss rate and different RTT values: (a) RTT = 100ms, (b) RTT = 200ms.

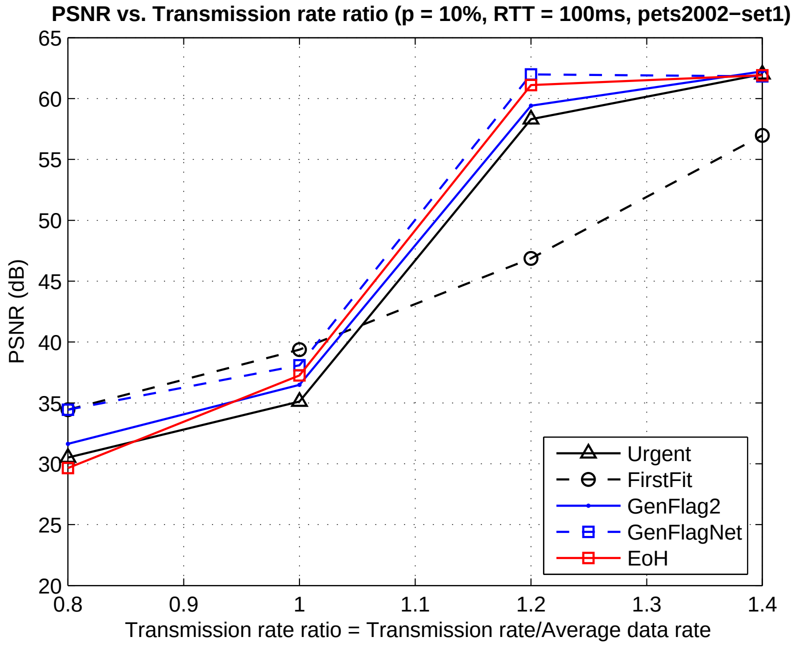


(a) RTT = 100ms

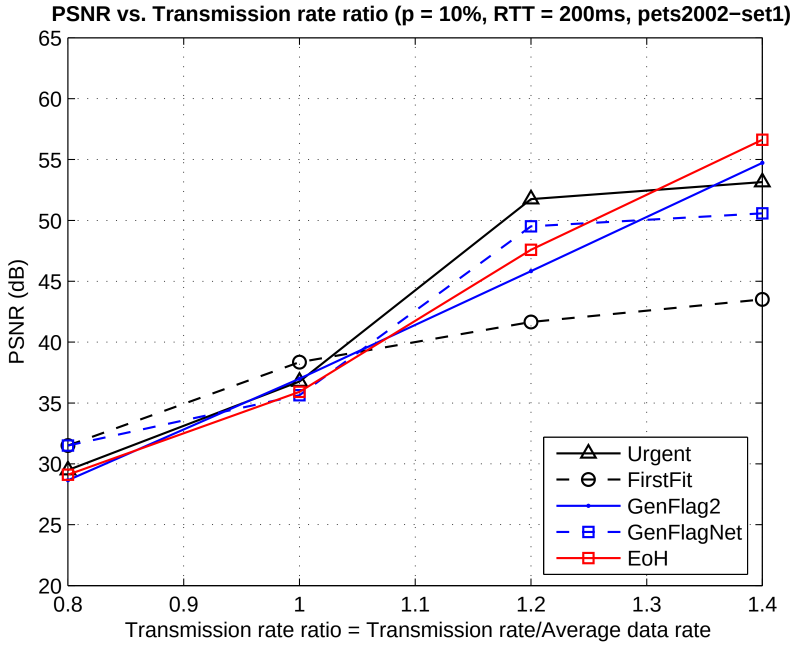


(b) RTT = 200ms

Figure 4.7: Average F-PSNR vs. Transmission rate ratio. Video Walk1-man is streamed with various transmission rates, under 10% network loss rate and different RTT values: (a) RTT = 100ms, (b) RTT = 200ms.

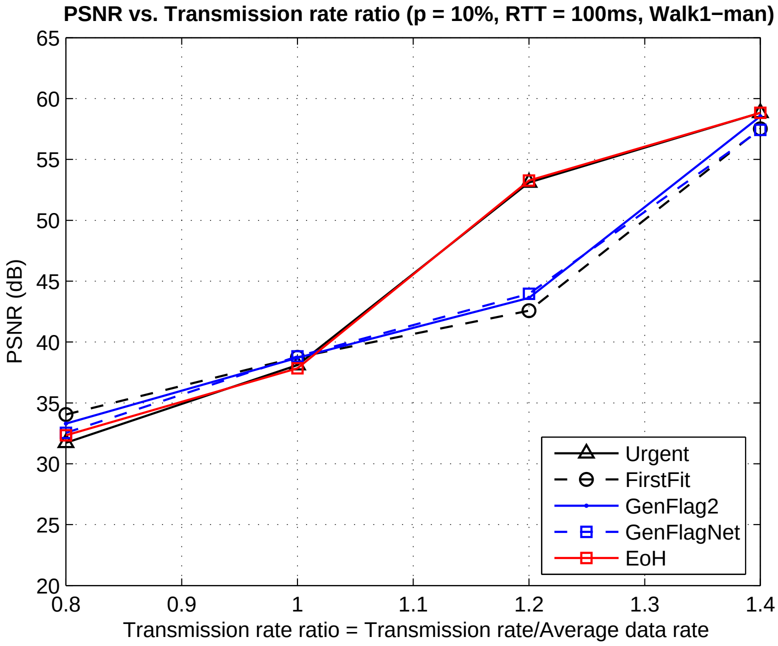


(a) RTT = 100ms

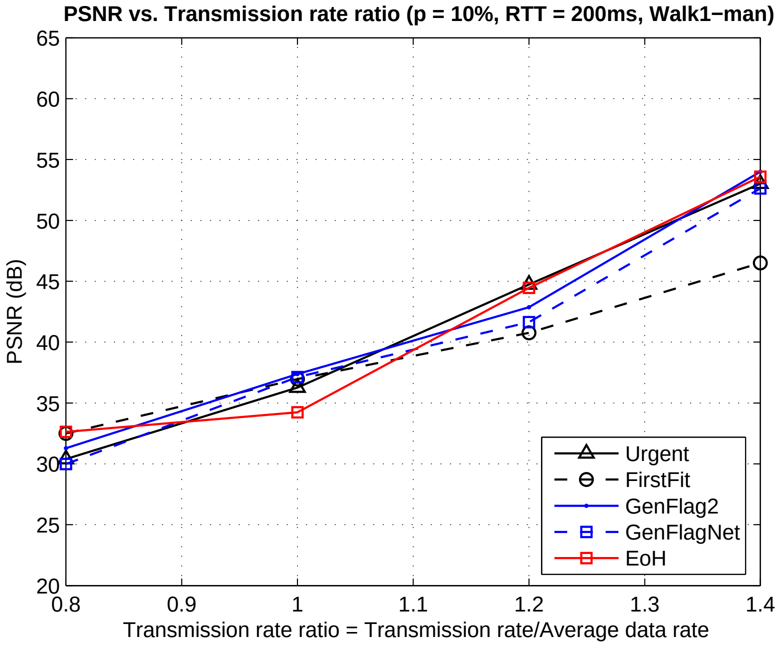


(b) RTT = 200ms

Figure 4.8: Average PSNR vs. Transmission rate ratio. Video pets2002-set1 is streamed with various transmission rates, under 10% network loss rate and different RTT values: (a) RTT = 100ms, (b) RTT = 200ms.



(a) RTT = 100ms



(b) RTT = 200ms

Figure 4.9: Average PSNR vs. Transmission rate ratio. Video Walk1-man is streamed with various transmission rates, under 10% network loss rate and different RTT values: (a) RTT = 100ms, (b) RTT = 200ms.

Transmission rate ratio	Scheduler	PSNR$_a \pm \sigma$		F-PSNR$_a \pm \sigma$	
		RTT=100ms	RTT=200ms	RTT=100ms	RTT=200ms
0.8	Urgent	30.51 ± 0.59	29.49 ± 2.52	19.56 ± 0.84	19.36 ± 0.91
	FirstFit	34.44 ± 0.81	31.50 ± 4.41	23.10 ± 0.98	22.86 ± 0.92
	GenFlag2	31.64 ± 0.66	28.66 ± 4.20	20.45 ± 0.87	19.89 ± 0.99
	GenFlagNet	31.48 ± 0.62	29.06 ± 3.70	20.38 ± 1.16	20.05 ± 1.02
	EoH	29.65 ± 1.96	29.10 ± 2.11	19.10 ± 0.92	19.06 ± 0.66
1.0	Urgent	35.11 ± 3.84	36.76 ± 1.48	24.87 ± 2.57	24.46 ± 1.83
	FirstFit	39.37 ± 0.76	38.35 ± 2.51	28.40 ± 1.40	28.05 ± 2.23
	GenFlag2	36.48 ± 2.47	37.00 ± 1.48	24.87 ± 2.61	25.06 ± 2.21
	GenFlagNet	38.09 ± 2.80	35.63 ± 3.95	28.11 ± 4.75	24.90 ± 4.14
	EoH	37.25 ± 2.13	35.91 ± 2.91	25.16 ± 2.66	24.79 ± 2.18
1.2	Urgent	58.30 ± 6.42	51.75 ± 7.37	54.65 ± 11.57	48.22 ± 15.05
	FirstFit	46.86 ± 3.29	41.65 ± 5.92	37.29 ± 9.51	30.64 ± 3.79
	GenFlag2	59.41 ± 3.56	45.83 ± 9.05	57.40 ± 6.32	43.68 ± 14.22
	GenFlagNet	61.99 ± 0.92	49.51 ± 5.87	59.53 ± 2.76	49.42 ± 13.41
	EoH	61.11 ± 3.22	47.58 ± 12.55	60.24 ± 0.00	48.95 ± 14.54
1.4	Urgent	62.00 ± 0.87	53.15 ± 5.77	60.24 ± 0.00	52.54 ± 12.88
	FirstFit	56.97 ± 6.87	43.51 ± 4.19	50.13 ± 12.58	30.99 ± 3.36
	GenFlag2	62.23 ± 0.00	54.73 ± 9.25	60.24 ± 0.00	54.10 ± 11.48
	GenFlagNet	61.83 ± 1.54	50.58 ± 9.88	60.24 ± 0.00	54.31 ± 9.61
	EoH	61.91 ± 1.24	56.62 ± 5.63	60.24 ± 0.00	55.35 ± 10.46

Table 4.3: Average and standard deviations of F-PSNR and PSNR measurements from 15 running times. Video pets2002-set1 is streamed with different transmission rate ratio, 10% network loss rate and different RTT.

4.5 Summary

In this chapter, we studied the effects of packet scheduling on the streaming quality of content-based prioritized videos. Debating that using either packet's priority or deadline is not enough, we considered both priority and deadline in making scheduling decisions. However, how deadline and priority are used greatly affects the scheduler's sensitivity to RTT and loss rate, thus the received quality. Particularly, while the video quality obtained from most scheduling algorithms changes significantly with respect to RTT and loss rate, the output received by sending highest-priority first (`FirstFit`) is relatively stable. Therefore, this scheduling policy works best under bad network conditions, e.g., high loss rate, low bandwidth. On the other hand,

Transmission rate ratio	Scheduler	PSNR$_a \pm \sigma$		F-PSNR$_a \pm \sigma$	
		RTT=100ms	RTT=200ms	RTT=100ms	RTT=200ms
0.8	Urgent	31.72 ± 3.07	30.39 ± 5.09	21.42 ± 1.55	20.63 ± 1.16
	FirstFit	34.04 ± 0.68	32.49 ± 4.55	27.62 ± 1.30	26.93 ± 1.65
	GenFlag2	33.30 ± 0.54	31.28 ± 5.27	25.50 ± 1.86	23.11 ± 1.41
	GenFlagNet	32.54 ± 3.15	29.99 ± 4.73	25.34 ± 2.14	22.46 ± 1.19
	EoH	32.33 ± 0.43	32.63 ± 1.45	22.18 ± 1.18	21.28 ± 0.88
1.0	Urgent	38.10 ± 1.78	36.27 ± 3.03	25.50 ± 4.79	22.73 ± 1.87
	FirstFit	38.75 ± 0.56	36.96 ± 2.82	31.58 ± 2.24	30.96 ± 2.84
	GenFlag2	38.70 ± 0.80	37.38 ± 3.29	31.61 ± 3.68	32.98 ± 5.08
	GenFlagNet	38.82 ± 0.57	37.13 ± 4.11	33.61 ± 4.62	31.06 ± 3.50
	EoH	37.84 ± 1.52	34.23 ± 5.61	24.64 ± 2.16	24.27 ± 1.93
1.2	Urgent	53.10 ± 5.16	44.73 ± 3.35	47.51 ± 11.87	32.79 ± 10.20
	FirstFit	42.58 ± 1.97	40.76 ± 3.88	33.55 ± 5.13	32.33 ± 4.73
	GenFlag2	43.62 ± 1.65	42.87 ± 1.69	35.11 ± 6.12	31.61 ± 3.14
	GenFlagNet	43.96 ± 2.19	41.64 ± 4.95	34.79 ± 4.71	33.33 ± 3.84
	EoH	53.27 ± 6.23	44.46 ± 5.36	47.28 ± 12.83	34.01 ± 9.35
1.4	Urgent	58.83 ± 0.00	53.02 ± 4.69	57.03 ± 0.00	48.56 ± 12.50
	FirstFit	57.52 ± 2.72	46.50 ± 2.79	57.03 ± 0.00	42.53 ± 12.71
	GenFlag2	58.53 ± 1.17	54.00 ± 3.53	57.03 ± 0.00	51.99 ± 10.30
	GenFlagNet	57.42 ± 3.92	52.62 ± 3.59	55.38 ± 6.39	47.19 ± 12.01
	EoH	58.83 ± 0.00	53.58 ± 4.88	57.03 ± 0.00	48.93 ± 11.49

Table 4.4: Average and standard deviations of F-PSNR and PSNR measurements from 15 running times. Video Walk1-man is streamed with different transmission rate ratio, 10% network loss rate and different RTT.

sending the earliest-deadline packet first is better in good conditions. We found that considering the highest-priority and earliest-deadline packet within a set of high-priority packets (`GenFlag2`'s approach), is normally good for most situations. In short, our study suggests the following schedule policy.

- Bad network condition (loss rate higher than 10%, low bandwidth): Send the highest-priority packet first (FirstFit).

- Good network condition (zero loss rate or bandwidth is 20% higher than the average data rate): Send the earliest-deadline packet first (Urgent).

- Normal condition: Within a set of high-priority packets, choose between the highest-priority packet and earliest-deadline packet (GenFlag2).

We also studied the effects of network characteristics like RTT on the scheduling performance. Intuitively, it may have substantial influence on the sending time, so considering RTT in making schedule decisions is reasonable. However, under our content-based video streaming scenarios, we find that the scheduling performance is not significantly changed with or without RTT consideration. For more demanding real-time applications, it may be different, but that will be the topic of our future studies.

Chapter 5

Conclusions

What we call the beginning is often the end.

And to make an end is to make a beginning.

The end is where we start from.

—Thomas Stearns Eliot

Unlike normal data packets, multimedia data packets usually have different priority. From coding perspective, some data packets may be more essential than others. From user perspective, a certain type of data or a specific information may be more important than the rest. Therefore, delivering a successful multimedia streaming experience to users over a limited bandwidth and lossy network is usually challenging, due to the random nature of packet loss and its potentially devastating effects on streaming quality.

In this thesis, we investigated the problems of prioritizing and delivering packets in multimedia streaming. Under a lossy network, the sender has to decide which packets are to be further protected from losses, which packets are to be sent, how to send them, and when to send them. The priority of a packet could be either based on its position in the coding interdependencies (syntax-based) or based on its semantic content (content-based). We studied these problems under different network scenarios, with different types of information available to the sender and found that significant quality improvements could be obtained if a good packet allocation, protection and/or scheduling scheme is employed. Besides, content-based prioritization could greatly improve the perceived quality compared to syntax-based prioritization.

5.1 Our approaches and contributions

Knowing that packet loss is the main cause of quality degradation in multimedia streaming, we started with a review on common approaches that minimize the effects of packet loss. We then studied how to stream packets over multiple paths, and found that by implicitly prioritizing packets based on their coding interdependencies, better quality could be achieved. This leads us to another question: "What should be used to prioritize packets?" Using video surveillance as an example, we found that if packets are prioritized based on their semantic content rather than syntax data, we could provide much better quality to users. The new question now is on how to deliver these prioritized packets. We found that there is no single best scheduling policy for all situations, and a good algorithm should consider both priority and deadline of the packets. We argue that our work supports the following general themes:

- "Applications know best" [70, 217], but users know better. Compared to low layers, applications may know which data (packet) requires more protection. However, it may not know what information or content is more important to users, as human do. Therefore, understanding user requirements is essential to design and deliver a good streaming service.
- It is always good to prioritize, either based on coding perspective or based on user perspective. But significant improvement could be achieved if prioritization is based on users' interest.

- Delivery prioritized packets is not trivial as it is commonly assumed. Considering priority only may not be the best policy. In many cases, considering both deadline and priority can bring about better results. Furthermore, RTT's effects are not that significant as we expected.

5.1.1 Review and user requirements

In Chapter 1, we presented a comprehensive review on common approaches that minimize the effects of packet loss, with a focus on transmission-based methods. We discussed these methods in the consideration of common user requirements and network characteristics, which are sometimes mistakenly exaggerated or understated in literature. For example, user requirements are not very stringent as they are traditionally believed, e.g., conversational video can stand a one-way delay up to 400ms, while streaming audio and video can tolerate a one-way delay up to 10s [2, 3, 134, 137]. Similarly, although network behaviors are unpredictable, research shows that their constancy could be safely assumed in scale of minutes [332]. Meanwhile, a common RTT value on the current Internet is normally around 134–160ms, or at most 200ms [74, 241]. For such RTT values, link-layer retransmission could be used if the required delay is around 1s, and application-layer retransmission could be used if the required delay is 2–3s [92]. Such observations motivated us to investigate and compare FEC-based and retransmission-based delivery approaches in better light, as well as lay the foundation for subsequent chapters. For example, given the above values of RTT and required delay, it is obviously possible to use retransmissions for packet recovery in streaming applications. The large delay may also allow us to optimize packet allocating and scheduling plans before transmission.

5.1.2 The benefits of prioritization

Chapter 2 studied the problem of streaming multimedia packets over multiple paths. A common way is to use Multiple Description Coding (MDC) to create independent packets with similar quality contribution, thus any packet could be sent over any path. We showed that by using Layer Coding (LC, in which packets are implicitly prioritized by grouping into different layers based on their interrelationships) instead of MDC, a sender can cleverly decide which packets to send over which path, therefore can provide much better quality under critical network conditions. We demonstrated this observation by comparing the reconstructed quality between streaming MDC and streaming LC data over a two-path network. The experimental results cleared the belief that MDC is better than LC for multimedia streaming.

5.1.3 What and how to prioritize?

While Chapter 2 showed the benefits of prioritizing data, Chapter 3 addressed the question of what to prioritize. We argued that instead of prioritizing syntax data, we should consider what contents are important to users, and prioritize such contents to improve the perceived quality. For example, in video surveillance, we could identify the regions of interest where users are more likely to pay attention to. We showed that prioritizing packets based on semantic regions of interests within frame could achieve dramatic quality improvement compared to prioritizing packets based on syntax (e.g., frame type).

To objectively measure the quality of regions of interests, a new performance metric called Focused-PSNR (F-PSNR) was proposed. F-PSNR is similar to PSNR, except that it restricts the calculation of the metric only to regions of interest. Our experiments showed that the videos obtained by our method could have 6–11dB higher F-PSNR than what obtained by standard method used in literature. For subjective measurements, videos were shown to different users and graded by Mean Opinion

Square (MOS). While the videos produced by the standard method have relatively low MOS (0.9–2.2), our methods provided much better video quality (MOS of 7.8–9.2).

The above results are obtained when packets are protected by possible retransmission(s). We then extended our content-based prioritizing scheme to consider FEC protection when no retransmission is allowed. In this case, how should we allocate FEC packets? We showed that content-based FEC performance could be 10–17% higher than frame-based FEC under normal conditions, and even higher (36%) under severer bandwidth constraint.

5.1.4 How to send prioritized packets?

Chapter 4 shifted the focus from packet prioritization and FEC protection to packet scheduling. We showed that while scheduling packets primarily based on their priority seems to be a natural way for prioritized packets, it only works best under difficult network conditions, e.g., high loss rate and limited bandwidth, but mediocre in other scenarios. If the network is good (very low loss rate, high bandwidth), sending packets based on their deadline first offers significant better quality. We found that considering both highest-priority packet and earliest-deadline packet within a set of high-priority packets often provides good performance in most situations.

We also studied the effects of RTT on scheduling performance. Under our content-based video streaming scenarios, we found that scheduling performance is not significantly changed with or without RTT consideration. It is surprising, since intuitively RTT may have substantial influence on the output quality, and considering RTT in making schedule decision is expected to be beneficial.

5.2 Future research

For future works, several extensions are worth to be investigated.

In Chapter 2, we assumed a simple one sender, one receiver model, so that the optimal packet allocation for multiple paths could be calculated off-line at the sender. However, in multicast scenario where a receiver wants to receive the same data from multiple senders, how should we coordinate the senders to find an optimal packet allocation? Having a central coordinator, which receives information from all senders, computes and sends optimal solution back to the senders, may not be feasible, scalable and adaptive enough. Perhaps, the receiver itself in this case is a better candidate to decide its senders policy.

It is also interesting to study the effects of RTT on scheduling performance in other scenarios, such as real-time, interactive games or conversational video applications, where delay requirement is more stringent than streaming. Furthermore, the work in Chapter 4 could be extended to scheduling of multiple multimedia streams with different priorities and deadlines, and packets within each stream also have different priorities and deadlines. Should we optimize the total quality of all streams, or optimize the quality of the most important stream? In the first case, how to define and measure the total quality?

There are no easy answer for these questions, since perception is always personal, and different users often have different perception, requirements and priorities [38, 243, 322]. Within an image or audio segment, different users would focus on different regions and aspects, thus the quality of such region-of-interests would determine their perception about the overall quality. Therefore, users should be the center and the quality judge of any multimedia system.

The same user may even vary his or her priorities with time. For example, a typical news program in television normally has three sections: introduction by the host(s), news clips, and weather forecast. In the first section, audio is normally more important than video since the images of host(s) and studio are not just unimportant but also usually unchanged. In the second section, the relative importance between video and audio depends on the content of each clips and user's preference. For instance, audio content of an economic news is often more important than its accompanied video, while images of scores in sport news is more important than the commenting audio. In the section of weather forecast, the video usually contains more information and easier to understand, thus is more important, than the audio.

Therefore, it is beneficial to have users' feedback on what is important to them, and how they define what is good. Knowing which data are more important for the user would certainly help the system to know where to put higher protection during encoding process, where to allocate more resources during transmission process, and where to emphasize its concealment effort. In Chapter 3 and Chapter 4, we assume that in video surveillance, moving objects like pedestrians, cars are the regions of interests for users. However, sometimes a non-moving object, e.g., an unaccompanied bag, a deserted car, could be more intrigued and important to users.

Nevertheless, designing and implementing a multimedia streaming system to fulfil such requirements is very complex and requires knowledge from different disciplines, e.g., psychology, computer vision, computer network. Recently, a few works have been published on this area [58,79,168,180,308]. We believe that our work may contribute towards developing such user-centric systems: ones that are constantly aware of and adapt to users' needs.

Bibliography

[1] *Open Computer Vision Library (version 1.0)*. http://www.intel.com/technology/computing/opencv, November 2006.

[2] 3rd Generation Partnership Project: Technical Specification Group Services and Systems Aspects. Services and services capabilities (release 6). Technical Specification 3GPP TS 22.105 V6.4.0, September 2005.

[3] 3rd Generation Partnership Project: Technical Specification Group Services and Systems Aspects. Quality of Service (QoS) concept and architecture (release 6). Technical Specification 3GPP TS 23.107 V6.4.0, March 2006.

[4] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 121–132, August 2004.

[5] Philippe Aigrain, Hong-Jiang Zhang, and Dragutin Petkovic. Content-based representation and retrieval of visual media: A state-of-the-art review. *Multimedia Tools and Applications*, 3(3):179–202, 1996.

[6] Miklos Ajtai, James Aspnes, Cynthia Dwork, and Orli Waarts. A theory of competitive analysis for distributed algorithms. In *Symposium Foundations of Computer Science*, pages 401–411, 1994.

[7] Aditya Akella, Srinivasan Seshan, and Anees Shaikh. An empirical evaluation of wide-area Internet bottlenecks. In *ACM SIGCOMM Internet Measurement Conference*, pages 101–114, October 2003.

[8] Mohammed Ebrahim Al-Mualla, Cedric Nishan Canagarajah, and David R. Bull. *Video coding for mobile communications: efficiency, complexity, and resilience*. Academic Press, April 2002.

[9] Andres Albanese, Johannes Blomer, Jeff Edmonds, Michael Luby, and Madhu Sudan. Priority encoding transmission. In *IEEE Symposium on Foundations of Computer Science*, pages 604–612, 1994.

[10] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Resilient overlay networks. In *ACM Symposium on Operating Systems Principles*, October 2001.

[11] David G. Andersen, Alex C. Snoeren, and Hari Balakrishnan. Best-path vs. multi-path overlay routing. In *ACM SIGCOMM Internet Measurement Conference*, October 2003.

[12] John G. Apostolopoulos. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In *SPIE Conference on Visual Communication and Image Processing*, January 2001.

[13] John G. Apostolopoulos, Wai-Tian Tan, Susie J. Wee, and Gregory W. Wornell. Modeling path diversity for multimedia description video communication. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 2161–2164, May 2002.

[14] John G. Apostolopoulos, Tina Wong, Wai-Tian Tan, and Susie J. Wee. On multiple description streaming with content delivery networks. In *IEEE Conference on Computer Communications*, June 2002.

[15] Supavadee Aramvith, Chia-Wen Lin, Sumit Roy, and Ming-Ting Sun. Wireless video transport using conditional retransmission and low-delay interleaving. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):558–565, June 2002.

[16] Jerry Ash. Max allocation with reservation bandwidth constraints model for DiffServ-aware MPLS traffic engineering & performance comparisons. RFC 4126, The Internet Engineering Task Force, June 2005. http://www.ietf.org/rfc/rfc4126.txt.

[17] Cristina Aurrecoechea, Andrew T. Campbell, and Linda Hauw. A survey of QoS architectures. *Multimedia Systems*, 6(3):138–151, 1998.

[18] Yossi Azar, Joseph (Seffi) Naor, and Raphael Rom. Routing strategies for fast networks. In *IEEE Conference on Computer Communications*, pages 170–179, 1992.

[19] Yossi Azar, Joseph (Seffi) Naor, and Raphael Rom. Routing strategies for fast networks. *IEEE Transactions on Computers*, 45(2):165–173, February 1996.

[20] Sandeep Bajaj, Lee Breslau, and Scott Shenker. Uniform versus priority dropping for layered video. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 131–143, October 1998.

[21] Ivan V. Bajic and John W. Woods. Domain-based multiple description coding of images and video. *IEEE Transactions on Image Processing*, 12:1211–1225, October 2003.

[22] Scott A. Banachowski, Joel Wu, and Scott A. Brandt. Missed deadline notification in best-effort schedulers. In *SPIE/ACM Multimedia Computing and Networking*, pages 123–135, January 2004.

[23] Yair Bartal, Francis Y. L. Chin, Marek Chrobak, Stanley P. Y. Fung, Wojciech Jawor, Ron Lavi, Jiri Sgall, and Tomás Tichý. Online competitive algorithms for maximizing weighted throughput of unit jobs. In *Symposium on Theoretical Aspects of Computer Science*, pages 187–198, 2004.

[24] Ali C. Begen, Yucel Altunbasak, and Ozlem Ergun. Fast heuristics for multi-path selection for multiple description encoded video streaming. In *IEEE International Conference on Multimedia and Expo*, volume 3, pages 517–520, July 2003.

[25] Ali C. Begen, Yucel Altunbasak, and Ozlem Ergun. Multi-path selection for multiple description encoded video streaming. In *IEEE International Conference on Communications*, volume 3, pages 1583–1589, May 2003.

[26] Piotr Berman, Moses Charikar, and Marek Karpinski. On-line load balancing for related machines. *Journal of Algorithms*, 35:108–121, 2000.

[27] Josh Bernoff. The real potential of Internet video. Research report, Forrester Research, September 2005.

[28] Samrat Bhattacharjee, Kenneth L. Calvert, and Ellen W. Zegura. Active networking and the end-to-end argument. In *IEEE International Conference on Network Protocols*, October 1997.

[29] John C. Bicket. Bit-rate selection in wireless networks. Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, February 2005.

[30] Richard E. Blahut. *Algebraic codes for data transmission*. Cambridge University Press, 2003.

[31] Steven Blake, David L. Black, Mark A. Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated ser-

vices. RFC 2475, The Internet Engineering Task Force, December 1998. http://www.ietf.org/rfc/rfc2475.txt.

[32] Marjory S. Blumenthal and David D. Clark. Rethinking the design of the Internet: the end-to-end arguments vs. the brave new world. *ACM Transactions on Internet Technology (TOIT)*, 1(1):70–109, August 2001.

[33] Paul Bocheck and Shih-Fu Chang. Content-based dynamic resource allocation for VBR video in bandwidth limited networks. May 1998.

[34] Jean-Chrysostome Bolot. Characterizing end-to-end packet delay and loss in the Internet. *Journal of High Speed Networks*, 2(3):289–298, September 1993.

[35] Jean-Chrysostome Bolot and Andrés Vega-García. Control mechanisms for packet audio in the Internet. *IEEE Conference on Computer Communications*, pages 232–239, April 1996.

[36] Jean-Chrysotome Bolot. End-to-end packet delay and loss behavior in the internet. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 289–298, 1993.

[37] Vincent Bottreau, Marion Benetiere, Boris Felts, and Béatrice Pesquet-Popescu. A fully scalable 3D subband video codec. In *IEEE International Conference on Image Processing*, volume 2, pages 1017–1020, October 2001.

[38] Anna Bouch, Allan Kuchinsky, and Nina Bhatti. "Quality is in the eye of the beholder": Meeting users' requirements for Internet quality of service. In *CHI Conference on Human Factors in Computing Systems*, pages 297–304, 2000.

[39] Alan C. Bovik. *Handbook of image and video processing.* Elsevier/Academic Press, 2nd edition, May 2005.

[40] Jill M. Boyce and Robert D. Gaglianello. Packet loss effects on MPEG video sent over the public Internet. In *ACM International Conference on Multimedia*, 1998.

[41] Robert (Bob) T. Braden, David D. Clark, and Scott Shenker. Integrated services in the Internet architecture: an overview. RFC 1633, The Internet Engineering Task Force, June 1994. http://www.ietf.org/rfc/rfc1633.txt.

[42] Robert (Bob) T. Braden, Lixia Zhang, Steve Berson, Shai Herzog, and Sugih Jamin. Resource ReSerVation Protocol (RSVP). RFC 2205, The Internet Engineering Task Force, September 1997. http://www.ietf.org/rfc/rfc2205.txt.

[43] Andrew P. Bradley and Fred W. M. Stentiford. Visual attention for region of interest coding in JPEG 2000. *Journal of Visual Communication and Image Representation*, 14:232–250, January–March 2003.

[44] John Byers, Michael Frumin, Gavin Horn, Michael Luby, Michael Mitzenmacher, Alex Roetter, and William Shaver. FLID-DL: Congestion control for layered multicast. In *International workshop on Networked Group Communication*, November 2000.

[45] Jianfei Cai and Chang Wen Chen. Video streaming: an FEC-based novel approach. In *Canadian Conference on Electrical and Computer Engineering*, volume 1, pages 613–618, 2001.

[46] Jianfei Cai, Qian Zhang, Wenwu Zhu, and Chang Wen Chen. An FEC-based error control scheme for wireless MPEG-4 video transmission. In *Wireless Communications and Networking Conference*, volume 3, pages 1243–1247, 2000.

[47] Jeffrey D. Case, Mark Fedor, Martin Lee Schoffstall, and James R. Davin. A Simple Network Management Protocol (SNMP). RFC 1157, The Internet Engineering Task Force, May 1990. http://www.ietf.org/rfc/rfc1157.txt.

[48] Pittsburgh Supercomputing Center. Experimental TCP Selective Acknowledgment implementations. http://www.psc.edu/networking/projects/sack/, May 2006.

[49] Jacob Chakareski and Bernd Girod. Rate-distortion optimized packet scheduling and routing for media streaming with packet diversity. In *IEEE Data Compression Conference*, 2003.

[50] Jacob Chakareski, Sangeun Han, and Bernd Girod. Layered coding vs. multiple descriptions for video streaming over multiple paths. In *ACM International Conference on Multimedia*, pages 422–431, 2003.

[51] Ee-Chien Chang and Chee Yap. Competitive online scheduling with level of service. In *International Computing and Combinatorics Conferences*, pages 453–462, 2001.

[52] Shih-Fu Chang, Di Zhong, and Raj Kumar. Real-time content-based adaptive streaming of sports videos. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 139–146, December 2001.

[53] Yung-Chi Chang, Chih-Wei Hsu, Wei-Min Chao, and Liang-Gee Chen. Interactive content-aware video streaming system with fine granularity scalability. *The Journal of VLSI Signal Processing*, 44(1–2):117–134, August 2006.

[54] Mainak Chatterjee, Giridhar D. Mandyam, and Sajal K. Das. Fast ARQ in high speed downlink packet access for WCDMA systems. In *European Wireless*, pages 451–457, February 2002.

[55] Mainak Chatterjee, Giridhar D. Mandyam, and Sajal K. Das. Joint reliability of Medium Access Control and Radio Link Protocol in 3G CDMA systems. *IEEE Transactions on Computers*, 54(12):1584–1597, December 2005.

[56] Thomas M. Chen and Alden W. Jackson. Commentaries on "Active networking and end-to-end arguments". *IEEE Network*, 12(3):66–71, May/June 1998.

[57] Yingwei Chen and William A. Pearlman. Three-dimensional subband coding of video using the zerotree method. In *SPIE Conference on Visual Communication and Image Processing*, March 1996.

[58] Yu-Hao Chen, Jin-Hau Kuo, Wei-Ta Chu, and Ja-Ling Wu. Movie emotional event detection based on music mood and video tempo. In *IEEE International Conference on Consumer Electronics. Digest of Technical Papers*, pages 151–152, January 2006.

[59] Lianji Cheng, Wenjun Zhang, and Li Chen. Rate-distortion optimized unequal loss protection for FGS compressed video. *IEEE Transactions on Broadcasting*, 50(2):126– 131, June 2004.

[60] Francis Y. L. Chin and Stanley P. Y. Fung. Online scheduling for partial job values: Does timesharing or randomization help? *Algorithmica*, 37(3):149–164, 2003.

[61] Ananthanarayanan Chockalingam and Gang Bao. Performance of TCP/RLP protocol stack on correlated fading DS-CDMA wireless links. *IEEE Transactions on Vehicular Technology*, 49(1):28–33, January 2000.

[62] Jeong-Yong Choi and Jitae Shin. Content-aware packet-level interleaving method for video transmission over wireless networks. In *International Conference on Wired/Wireless Internet Communications*, pages 149–158, May 2005.

[63] Lai-U Choi, Wolfang Killerer, and Eckehard G. Steinbach. Cross layer optimization for wireless multi-user video streaming. In *IEEE International Conference on Image Processing*, volume 3, pages 2047–2050, October 2004.

[64] Sunghyun Choi and Youngkyu Choi. IEEE 802.11 MAC-level FEC with retransmission combining. *IEEE Transactions on Wireless Communications*, 5(1):203–211, January 2006.

[65] Philip A. Chou, Alexander E. Mohr, Albert Wang, and Sanjeev Mehrotra. Error control for receiver-driven layered multicast of audio and video. *IEEE Transactions on Multimedia*, 3(1):108–122, March 2001.

[66] Marek Chrobak, Wojciech Jawor, Jiri Sgall, and Tomás Tichý. Improved online algorithms for buffer management in QoS switches. In *European Symponium on Algorithm*, 2004.

[67] Hao-Hua Chu and Klara Nahrstedt. A soft real time scheduling server in unix operating system. In *Interactive Distributed Multimedia Systems and Telecommunication Services*, pages 153–162, 1997.

[68] Israel Cidon, Raphael Rom, and Yuval Shavitt. Multi-path routing combined with resource reservation. In *IEEE Conference on Computer Communications*, pages 92–100, 1997.

[69] David D. Clark and David L. Tennenhouse. Architectural considerations for a new generation of protocols. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 200–208, September 1990.

[70] Michele Clark and Kevin Jeffay. Application-level measurements of performance on the vBNS. In *IEEE International Conference on Multimedia Computing and Systems*, volume 2, page 362, Washington, DC, USA, 1999. IEEE Computer Society.

[71] Mark Cooper, editor. *Open Architecture as Communications Policy.* Center for Internet and Society, June 2004.

[72] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, 1997.

[73] Daniel J. Costello, Joachim Hagenauer, Kideki Imai, and Stephen B. Wicker. Applications of error-control coding. *Information Theory, IEEE Transactions on*, 44(6):2531–2560, October 1998.

[74] R. Les Cottrell, Saad Ansari, Parakram Khandpur, Ruchi Gupta, Richard Hughes-Jones, Michael Chen, Larry McIntosh, and Frank Leers. Characterization and evaluation of TCP and UDP-based transport on real networks. In *International workshop on Protocols for Fast-Long Distance Network*, February 2005.

[75] Shuguang (Robert) Cui. *Cross-layer optimization in energy constrained networks*. PhD thesis, Department of Electrical Engineering, Stanford University, June 2005.

[76] Kevin Dankwardt. Real time and Linux. *Linux Journal*, January 2002.

[77] Stephen E. Deering and Robert M. Hinden. Internet Protocol, Version 6 (IPv6). RFC 2460, The Internet Engineering Task Force, December 1998. http://www.ietf.org/rfc/rfc2460.txt.

[78] L. Peter Deutsch and Jean-Loup Gailly. ZLIB compressed data format specification version 3.3. RFC 1950, The Internet Engineering Task Force, May 1996. http://www.ietf.org/rfc/rfc1950.txt.

[79] Nevenka Dimitrova. Context and memory in multimedia content analysis. *IEEE Multimedia*, 11(3):7–11, July/September 2004.

[80] Xiaowei Ding and Kaushik Roy. A novel bitstream level joint channel error concealment scheme for realtime video over wireless networks. In *IEEE Conference on Computer Communications*, volume 3, pages 2163–2173, March 2004.

[81] Danny Dolev. The Byzantine generals strike again. *Journal of Algorithms*, 3(1):14–30, March 1982.

[82] Jianyu Dong and Yuan F. Zheng. Content-based retransmission for 3-D wavelet video streaming over lossy networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(9):1125–1133, September 2006.

[83] Vent A. Dubendorf. *Wireless data technologies*. Wiley, 2003.

[84] Robert E. Van Dyck and David J. Miller. Transport of wireless video using separate, concatenated, and joint source-channel coding. *Proceedings of the IEEE*, 87(10):1734–1750, October 1999.

[85] Touradj Ebrahimi and Fernando Pereira. *The MPEG-4 book*. Prentice Hall, 2002.

[86] David Eckhardt and Peter Steenkiste. Measurement and analysis of the error characteristics of an in-building wireless network. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 243–254, 1996.

[87] Randall R. Stewart (editor). Stream Control Transmission Protocol. RFC 4960, The Internet Engineering Task Force, September 2007. http://www.ietf.org/rfc/rfc4960.txt.

[88] Yiftach Eisenberg, Carlos E. Luna, Thrasyvoulos N. Pappas, Randall Berry, and Aggelos K. Katsaggelos. Joint source coding and transmission power management for energy efficient wireless video communications. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):411–424, June 2002.

[89] Hala Elaarag. Improving TCP performance over mobile networks. *ACM Computing Surveys*, 34(3):357–374, September 2002.

[90] Alexandros Eleftheriadis and Dimitris Anastassiou. Meeting arbitrary QoS constraints using dynamic rate shaping of coded digital video. In *ACM International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 95–106, April 1995.

[91] Faruk Eryurtlu, Ahmet M. Kondoz, and Barry G. Evans. Sub-band video coding using edge information. In *IEE colloquium on Coding for packet video and speech transmission*, November 1992.

[92] Minoru Etoh and Takeshi Yoshimura. Advances in wireless video delivery. *Proceedings of the IEEE: Special issue on Advances in video coding and delivery*, 93(1):111–122, January 2005.

[93] Nariman Farvardin and Vinay A. Vaishampayan. Optimal quantizer design for noisy channels: An approach to combined source-channel coding. *IEEE Transactions on Information Theory*, 33(6):827–838, November 1987.

[94] Francois Le Faucheur and Wai Sum Lai. Requirements for support of differentiated services-aware MPLS traffic engineering. RFC 3564, The Internet Engineering Task Force, July 2003. http://www.ietf.org/rfc/rfc3564.txt.

[95] Nick Feamster and Hari Balakrishnan. Packet loss recovery for streaming video. In *International Packet Video Workshop*, April 2002.

[96] Federal CIO Council's Architecture and Infrastructure Committee. *IPv6 Transition Guidance (Version 1.0)*, February 2006.

[97] John G. Fletcher. An arithmetic checksum for serial transmissions. *IEEE Transactions on Communication*, 30(1):247–252, January 1982.

[98] Markus Flierl and Bernd Girod. Video coding with motion-compensated lifted wavelet transforms. *IEEE Transactions on Image Communication*, 19:561–575, 2004.

[99] Sally Floyd and Kevin Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, August 1999.

[100] Sally Floyd, Mark Handley, Jitendra Padhye, and Jörg Widmer. Equation-based congestion control for unicast applications. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 43–56, August 2000.

[101] Sally Floyd and Van Jacobson. Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3(4):365–386, 1995.

[102] Gian Luca Foresti. Object recognition and tracking for remote video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1045–1062, October 1999.

[103] Thomas L. Friedman. *The world is flat: a brief history of the twenty-first century*, chapter The ten forces that flattened the world, pages 72–75. Farrar, Straus and Giroux, 1st revised and expanded edition, January 2006.

[104] Pascal Frossard and Olivier Verscheure. AMISP: a complete content-based MPEG-2 error-resilient scheme. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(9):989–998, September 2001.

[105] Pascal Frossard and Olivier Verscheure. Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery. *IEEE Transactions on Image Processing*, 10(12):1815–1825, December 2001.

[106] Shigeru Fukunaga, Toshihisa Nakai, and Hiroshi Inoue. Error resilient video coding by dynamic replacing of reference pictures. In *IEEE Global Telecommunications Conference*, volume 3, pages 1503–1508, November 1996.

[107] Borko Furht, Stephen W. Smolliar, and Hong-Jiang Zhang. *Video and image processing in multimedia systems*. Kluwer Academic, 1995.

[108] Marco Furini and Donald F. Towsley. Real-time traffic transmission over the Internet. *IEEE Transactions on Multimedia*, 3(1):33–40, March 2001.

[109] Michael Gallant and Faouzi Kossentini. Rate-distortion optimized layered coding with unequal error protection for robust internet video. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):357–372, March 2001.

[110] Vijay K. Garg. *Wireless network evolution: 2G to 3G*. Prentice Hall, 2001.

[111] Mohammed Ghanbari. *Standard codecs: Image compression to advanced video coding*. The Institution of Electrical Engineers, June 2003.

[112] Jerry D. Gibson. *Digital compression for multimedia: principles and standards*. Morgan Kaufmann, 1988.

[113] Bernd Girod and Niko Färber. Feedback-based error control for mobile video transmission. *Proceedings of the IEEE*, pages 1707–1723, October 1999.

[114] Vivek K. Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, 18(5):74–93, September 2001.

[115] Vivek K. Goyal and Jelena Kovacevic. Generalized multiple description coding with correlating transforms. *IEEE Transactions on Information Theory*, 47(6):2199–2224, September 2001.

[116] Ronald L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45:1563–1581, 1966.

[117] Matthias Grossglauser, Srinivasan Keshav, and David N. C. Tse. RCBR: A simple and efficient service for multiple time-scale traffic. *IEEE/ACM Transactions on Networking*, 5(6):741–755, December 1997.

[118] Stephen R. Gulliver and George Ghinea. Stars in their eyes: what eye-tracking reveals about multimedia perceptual quality. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 34(4):472–482, July 2004.

[119] Mark Handley, Sally Floyd, Jitendra Padhye, and Jörg Widmer. TCP Friendly Rate Control (TFRC): Protocol specification. RFC 3448, The Internet Engineering Task Force, January 2003. http://www.ietf.org/rfc/rfc3448.txt.

[120] Khaled Harfoush, Azer Bestavros, and John Byers. Measuring bottleneck bandwidth of targeted path segments. volume 3, pages 2079–2089, March–April 2003.

[121] Christian Hoene. *Internet telephony over wireless links.* PhD thesis, Faculty of Electronical Engineering and Computer Science, Berlin University of Technology, 2006.

[122] Don Hoffman, Gerard Fernando, Vivek Goyal, and M. Reha Civanlar. RTP payload format for MPEG1/MPEG2 video. RFC 2250, The Internet Engineering Task Force, January 1998. http://www.ietf.org/rfc/rfc2250.txt.

[123] Ningning Hu, Li (Erran) Li, Zhuoqing Morley Mao, Peter Steenkiste, and Jia Wang. Locating Internet bottlenecks: algorithms, measurements, and implications. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 41–54, August 2004.

[124] Ningning Hu and Peter Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE Journal on Selected Areas in Communications: Special issue in Internet and WWW Measurement, Mapping, and Modeling*, 21(6), August 2003.

[125] International Organization for Standardization. *ISO/IEC 11172-1:1993: Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 1: Systems*, 1993.

[126] International Organization for Standardization. *ISO/IEC 13818-1:1996: Information technology - Generic coding of moving pictures and associated audio information: Systems*, 1996. Latest update: 2000.

[127] International Organization for Standardization. *ISO/IEC 14496-1:2004: Information technology - Coding of audio-visual objects - Part 1: Systems*, 2004.

[128] International Organization for Standardization. *ISO/IEC 15444-1:2004: Information technology - JPEG 2000 image coding system: Core coding system*, 2004.

[129] International Organization for Standardization. *ISO/IEC 14496-10:2005: Information technology - Coding of audio-visual objects - Part 10: Advanced Video Coding*, March 2005.

[130] International Telecommunication Union. *Recommendation H.262: Information technology - Generic coding of moving pictures and associated audio information: Video*, July 1995. Latest update: February 2000.

[131] International Telecommunication Union. *Recommendation P.800: Methods for subjective determination of transmission quality*, August 1996.

[132] International Telecommunication Union. *Recommendation P.800: Methods for subjective determination of transmission quality*, August 1996.

[133] International Telecommunication Union. *Recommendation BS.1387-1: Method for objective measurements of perceived audio quality (PEAQ), Revised 11/01*, January 2001.

[134] International Telecommunication Union. *Recommendation G.1010: End-user multimedia QoS categories*, November 2001.

[135] International Telecommunication Union. *Recommendation P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, February 2001.

[136] International Telecommunication Union. *Recommendation BT.500-11: Methodology for the subjective assessment of the quality of television pictures*, June 2002.

[137] International Telecommunication Union. *Recommendation Y.1541ar: Network performance objectives for IP-based services*, May 2002. Latest update: February 2003.

[138] International Telecommunication Union. *Recommendation G.107: The E-model, a computational model for use in transmission planning*, March 2005.

[139] International Telecommunication Union. *Recommendation H.264: Advanced video coding for generic audiovisual services*, March 2005.

[140] International Telecommunication Union. *Recommendation P.862: Revised Annex A — Reference implementations and conformance testing for Recommendations P.862, P.862.1 and P.862.2*, November 2005.

[141] International Telecommunication Union. *Recommendation G.107: New Appendix II - Provisional impairment factor framework for wideband speech transmission*, June 2006.

[142] International Telecommunication Union. *Recommendation P.800.1 : Mean Opinion Score (MOS) terminology*, July 2006.

[143] International Telecommunication Union. *Recommendation P.800.1: Mean Opinion Score (MOS) terminology*, July 2006.

[144] International Telecommunication Union. *Recommendation P.862.3: Application guide for objective quality measurement based on Recommendations P.862, P.862.1 and P.862.2*, November 2007.

[145] Laurent Itti and Christof Koch. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10(1):161–169, January 2001.

[146] Van Jacobson. Congestion avoidance and control. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 314–329, August 1988.

[147] Van Jacobson. Pathchar: A tool to infer characteristics of Internet paths. Talk at Mathematical Sciences Research Institute, April 1997. ftp://ftp.ee.lbl.gov/pathchar.

[148] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2002.

[149] Manish Jain and Constantinos Dovrolis. Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In *ACM SIGCOMM Internet Measurement Conference*, pages 272–277, October 2004.

[150] Abbas Jamalipour and Sirin Tekinay. Fourth generation wireless networks and interconnection standards. *IEEE Personal Communications*, October 2001.

[151] Wenqing Jiang and Antonio Ortega. Multiple description speech coding for robust communication over lossy packet networks. In *IEEE International Conference on Multimedia and Expo*, pages 444–447, 2000.

[152] Liang jin Lin and Antonio Ortega. Bit-rate control using piecewise approximated rate-distortion characteristics. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(4):446–459, August 1998.

[153] Michael B. Jones and John Regehr. CPU reservations and time constraints: Implementation experience on Windows NT. In *USENIX Windows NT Symposium*, pages 93–102, 1999.

[154] Matthias Jung, Jorg Nonnenmacher, and Ernst W. Biersack. Reliable multicast via satellite: Uni-directional versus Bi-directional communication. In *Kommunikation in Verteilten Systemen*, pages 264–275, 1999.

[155] Alex Kesselman, Zvi Lotker, Yishay Mansour, Boaz Patt-Shamir, and Maxim Sviridenko. Buffer overflow management in QoS switches. *SIAM Journal on Computing*, 33(3):563–583, 2004.

[156] Alex Kesselman, Yishay Mansour, and Rob van Stee. Improved competitive guarantees for QoS buffering. In *European Symposium on Algorithm*, 2003.

[157] Shoaib Khan, Svetoslav Duhovnikov, Eckehard G. Steinbach, Marco Sgroi, and Wolfgang Kellerer. Application-driven cross layer optimization for video streaming over wireless networks. *IEEE Communications Magazine*, January 2006.

[158] Beong-Jo Kim, Zixiang Xiong, and William A. Pearlman. Low bit-rate scalable video coding with 3d set partitioning in hierarchical trees (3D SPIHT). *IEEE Transactions on Circuits and Systems for Video Technology*, 10(8):1374–1387, December 2000.

[159] Chang-Su Kim and Sang-Uk Lee. Multiple description coding of motion fields for robust video transmission. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:999–1010, September 2001.

[160] Changick Kim and Jenq-Neng Hwang. Object-based video abstraction for video surveillance systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(12):1128–1138, December 2002.

[161] Joohee Kim, Russell M. Mersereau, and Yucel Altunbasak. Error-resilient image and video transmission over the Internet using unequal error protection. *IEEE Transactions on Image Processing*, 12(2):121–131, February 2003.

[162] Oleg Komogortsev and Javed Khan. Predictive perceptual compression for real time video communication. In *ACM International Conference on Multimedia*, pages 220–227, 2004.

[163] Charles Krasic, Jonathan Walpole, and Wu-Chi Feng. Quality-adaptive media streaming by priority drop. In *ACM International Workshop on Network and Operating System Support for Digital Audio and Video*, June 2003.

[164] Wilson Kwok and Huifang Sun. Multi-directional interpolation for spatial error concealment. *IEEE Transactions on Consumer Electronics*, 39:455–460, August 1993.

[165] Wei Lai, Xiao-Dong Gu, Ren-Hua Wang, Li-Rong Dai, and Hong-Jiang Zhang. A region based multiple frame-rate tradeoff of video streaming. In *IEEE International Conference on Image Processing*, volume 3, pages 2067–2070, October 2004.

[166] Wei Lai, Xiao-Dong Gu, Ren-Hua Wang, Wei-Ying Ma, and Hong-Jiang Zhang. A content-based bit allocation model for video streaming. In *IEEE International Conference on Multimedia and Expo*, volume 2, pages 1315–1318, June 2004.

[167] Karthik Lakshminarayanan, Venkata N. Padmanabhan, and Jitendra Padhye. Bandwidth estimation in broadband access networks. In *ACM SIGCOMM Internet Measurement Conference*, pages 314–321, 2004.

[168] Chia-Hsun Jackie Lee, Jon Wetzel, Chiun-Yi Ian Jang, Yang-Ting Bowbow Shen, Ting-Han Daniel Chen, and Ted Selker. Attention meter: A vision-based input toolkit for interaction designers. In *CHI Conference on Human Factors in Computing Systems*, April 2006.

[169] Hung-Ju Lee, Tihao Chiang, and Ya-Qin Zhang. Scalable rate control for mpeg-4 video. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(6):878–994, September 2000.

[170] Yen-Chi Lee, Joohee Kim, Yucel Altunbasak, and Russell M. Mersereau. Performance comparisons of layered and multiple description coded video streaming over error-prone networks. In *IEEE International Conference on Communication*, 2003.

[171] Arnaud Legout and Ernst W. Biersack. PLM: Fast convergence for cumulative layered multicast transmission schemes. In *ACM SIGMETRICS International Conference on Measurement and modeling*, pages 13–22, June 2000.

[172] Liyuan Li, Weimin Huang, Irene Y. H. Gu, and Qi Tian. Foreground object detection from videos containing complex background. In *ACM International Conference on Multimedia*, pages 2–10, 2003.

[173] Yi J. Liang, John G. Apostolopoulos, and Bernd Girod. Model-based delay-distortion optimization for video streaming using packet interleaving. In *36th Asilomar Conference on Signals, Systems and Computers*, November 2002.

[174] Yi J. Liang, John G. Apostolopoulos, and Bernd Girod. Analysis of packet loss for compressed video: Does burst-length matter? In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2003.

[175] Yi J. Liang, Eckehard G. Steinback, and Bernd Girod. Real-time voice communication over the Internet using packet path diversity. In *ACM International Conference on Multimedia*, pages 444–447, October 2001.

[176] Chih-Han Lin, Hao-Hua Chu, and Klara Nahrstedt. A soft real-time scheduling server on the Windows NT. In *USENIX Windows NT Symposium*, 1998.

[177] Haitao Lin and Sajal K. Das. TCP performance analysis of CDMA systems with RLP and MAC layer retransmissions. In *IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, pages 313–320, 2002.

[178] Haitao Lin and Sajal K. Das. Performance study of link layer and MAC layer protocols to support TCP in 3G CDMA systems. *IEEE Transactions on Mobile Computing*, 4(5):489–501, September 2005.

[179] Shunan Lin, Yao Wang, Shiwen Mao, and Shivendra S. Panwar. Video transport over ad-hoc networks using multiple paths. In *IEEE International Symposium on Circuits and Systems*, volume 1, pages 57–60, 2002.

[180] Dan Liu, Lie Lu, and Hong-Jiang Zhang. Automatic mood detection from acoustic music data. In *International Symposium on Music Information Retrieval*, 2003.

[181] Jun Liu and Mark Crovella. Using loss pairs to discover network properties. In *ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[182] Qingwen Liu, Shengli Zhou, and Georgios B. Giannakis. Cross-layer scheduling with prescribed QoS guarantees in adaptive wireless networks. *IEEE Journal on Selected Areas in Communications*, 23(5):1056–1066, May 2005.

[183] Tiecheng Liu and Chekuri Choudary. Content-aware streaming of lecture videos over wireless networks. In *IEEE International Symposium on Multimedia Software Engineering*, pages 458–465, December 2004.

[184] Tiecheng Liu and Chekuri Choudary. Real-time content analysis and adaptive transmission of lecture videos for mobile applications. In *ACM International Conference on Multimedia*, pages 400–403, 2004.

[185] Yong Liu, Yu Gu, Honggang Zhang, Weibo Gong, and Don Towsley. Application level relay for high-bandwidth data transport. In *Workshop on Networks for Grid Applications*, October 2004.

[186] Carlos E. Luna, Yiftach Eisenberg, Thrasyvoulos N. Pappas, Randall Berry, and Aggelos K. Katsaggelos. Joint source coding and data rate adaptation for energy efficient wireless video streaming. *IEEE Journal on Selected Areas in Communications*, 21:1710–1720, December 2003.

[187] Henrik Lundgren, Erik Nordström, and Christian Tschudin. Coping with communication gray zones in IEEE 802.11b based ad hoc networks. In *ACM International Workshop on Wireless Mobile Multimedia*, pages 49–55, September 2002.

[188] Petteri Luukkanen, Zhigang Rong, and Lin Ma. Performance of 1XTREME system for mixed voice and data communications. In *IEEE International Conference on Communications*, volume 5, pages 1411–1415, 2001.

[189] Mohammad Mahdian, Nikhil Bansal, Lisa K. Fleischer, Tracy Kimbrel, Baruch Schieber, and Maxim Sviridenko. Further improvements in competitive guaran-

tees for QoS buffering. In *International Colloquium on Automata, Languages, and Programming*, 2004.

[190] Jukka Manner and Xiaoming Fu. Analysis of existing quality-of-service signaling protocols. RFC 4094, The Internet Engineering Task Force, May 2005. http://www.ietf.org/rfc/rfc4094.txt.

[191] Yishay Mansour, Boaz Patt-Shamir, and Ofer Lapid. Optimal smoothing schedules for real-time streams. In *ACM Symposium on Principles of Distributed Computing*, pages 21–29, January 2000.

[192] Shiwen Mao, Shunan Lin, Shivendra S. Panwar, Yao Wang, and Emre Celebi. Video transport over ad hoc networks: Multistream coding with multipath transport. *IEEE Journal on Selected Areas in Communications*, 21(10):1721–1737, December 2003.

[193] Shiwen Mao, Shunan Lin, Yao Wang, Shivendra S. Panwar, and Yihan Li. Multipath video transport over wireless ad hoc networks. *IEEE Wireless Communications: Special issue on advances in wireless video*, 12(4):42–49, August 2005.

[194] Oge Marques and Borko Furht. *Distributed multimedia databases: Techniques and applications*, chapter Content-based visual information retrieval. Idea Group Publishing, 2001.

[195] Matt Mathis, Jamshid Mahdavi, Sally Floyd, and Allyn Romanow. TCP Selective Acknowledgment options. RFC 2018, The Internet Engineering Task Force, October 1996. http://www.ietf.org/rfc/rfc2018.txt.

[196] N. Matsuo, M. Yuito, and Y. Tokunaga. Packet interleaving for reducing speech quality degradation in packet voice communications. In *IEEE Global Telecommunications Conference*, pages 1787–1791, 1987.

[197] Nick F. Maxemchuk. *Dispersity routing in store-and-forward networks.* PhD thesis, University of Pennsylvania, Philadelphia, May 1975.

[198] Ketan Mayer-Patel, Long Le, and Georg Carle. An MPEG performance model and its application to adaptive forward error correction. In *ACM International Conference on Multimedia*, pages 1–10, 2002.

[199] Ahmed Mehaoua and Raouf Boutaba. The impacts of errors and delays on the performance of MPEG2 video communications. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2195–2198, 1999.

[200] Jeffrey Mogul and Stephen E. Deering. Path MTU discovery. RFC 1191, The Internet Engineering Task Force, November 1990. http://www.ietf.org/rfc/rfc1191.txt.

[201] Ghanbari Mohammed. Two-layer coding of video signals for VBR networks. *IEEE Journal on Selected Areas in Communications*, 7(5):771–781, June 1989.

[202] Alexander E. Mohr, Richard E. Ladner, and Eve A. Riskin. Approximately optimal assignment for unequal loss protection. In *IEEE International Conference on Image Processing*, September 2000.

[203] Alexander E. Mohr, Eve A. Riskin, and Richard E. Ladner. Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction. *IEEE Journal on Selected Areas in Communications: Special issue on error-resilient image and video transmission*, 18(6):819–828, June 2000.

[204] MSU Graphics and Media Lab (Video Group), http://www.compression.ru/video/quality_measure/perceptual_video_quality_tool_en.html. *MSU Perceptual Video Quality tool (version 1.0)*, January 2006.

[205] MSU Graphics and Media Lab (Video Group), http://www.compression.ru/video/quality_measure/vqmt_faq_en.html. *MSU Video Quality Measurement Tool: FAQ*, March 2007.

[206] Michael John Muuss. The story of the ping program, 1993. http://ftp.arl.mil/ mike/ping.html.

[207] King N. Ngan, Thomas Meier, and Douglas Chai. *Advanced video coding: principles and techniques*. Elsevier, 1st edition, 1999.

[208] King N. Ngan, Chi W. Yap, and Keng T. Tan. *Video coding for wireless communication systems*. Marcel Dekker, 2001.

[209] Thinh P. Q. Nguyen and Avideh Zakhor. Protocols for distributed video streaming. In *IEEE International Conference on Image Processing*, pages 185–188, September 2002.

[210] Vu-Thanh Nguyen, Wei-Tsang Ooi, and Ee-Chien Chang. Packet allocation for layer media transmission with path diversity. Technical Report TR21/03, School of Computing, National University of Singapore, 2003.

[211] Jens-Rainer Ohm. Advances in scalable video coding. *Proceedings of the IEEE: Special issue on Advances in video coding and delivery*, 93(1):42–56, January 2005.

[212] IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. `people_test_dataset1.mpg`. http://ftp.pets.rdg.ac.uk/PETS2002/PEOPLE/TESTING/DATASET1, June 2002.

[213] IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. `walk1.mpg`. http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1, May 2004. EC funded CAVIAR project/IST 2001 37540.

[214] Wei-Tsang Ooi. *Multimedia Network Toys.* http://sourceforge.net/projects/mnt/, February 2007.

[215] Wei-Tsang Ooi, Brian Smith, Sugata Mukhopadhyay, Haye Hsi Chan, Steve Weiss, and Matthew Chiua. The Dalí: a multimedia software library. In *SPIE Multimedia Computing and Networking*, January 1999.

[216] Ariel Orda and Raphael Rom. Routing with packet duplication and elimination in computer networks. *IEEE Transactions on Communications*, 36(7):860–866, July 1998.

[217] David E. Ott and Ketan Mayer-Patel. Transport-level protocol coordination in distributed multimedia applications. In *ACM International Conference on Multimedia*, pages 644–645, 2003.

[218] Jitendra Padhye, Victor Firoiu, Donald F. Towsley, and James F. Kurose. Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145, April 2000.

[219] SangHoon Park, Seungjoo Lee, and JongWon Kim. Network-adaptive high definition MPEG-2 streaming over IEEE 802.11a WLAN using frame-based prioritized packetization. In *WMASH*, pages 84–87, 2005.

[220] Vern Paxson. End-to-end Internet packet dynamics. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, volume 27, pages 139–154, September 1997.

[221] Vern Paxson. End-to-end Internet packet dynamics. *IEEE/ACM Transactions on Networking*, 7(3):277–292, June 1999.

[222] Vern Paxson. Strategies for sound Internet measurement. In *ACM SIGCOMM Internet Measurement Conference*, pages 263–271, October 2004.

[223] Vern Paxson. *Measurements and analysis of end-to-end Internet dynamics*. PhD thesis, University of California, Berkeley, April 2007.

[224] Colin Perkins, Orion Hodson, and Vicky Hardman. A survey of packet-loss recovery techniques for streaming audio. *IEEE Network Magazine*, 12(5):40–48, September/October 1998.

[225] Mark R. Pickering and John F. Arnold. A perceptually efficient VBR rate control algorithm. *IEEE Transactions on Image Processing*, 3(5):527–532, September 1994.

[226] Jon Postel. User Datagram Protocol. RFC 768, The Internet Engineering Task Force, August 1980. http://www.ietf.org/rfc/rfc0768.txt.

[227] Jon Postel. Transmission Control Protocol. RFC 793, The Internet Engineering Task Force, September 1981. http://www.ietf.org/rfc/rfc0793.txt.

[228] Eugene Prange. Cyclic error-correcting codes in two symbols. Technical report AFCRC-TN-57-103, Air Force Cambridge Research Center, Cambridge, Mass., 1957.

[229] Ravi Prasad, Constantinos Dovrolis, Margaret, and KC Claffy. Bandwidth estimation: metrics, measurement techniques, and tools. *IEEE Network*, 17(6):27–35, November-December 2003.

[230] Karumbu Premkumar and Ananthanarayanan Chockalingam. Performance analysis of RLC/MAC and LLC layers in a GPRS protocol stack. *IEEE Transactions on Vehicular Technology*, 53(5):1531–1546, 2004.

[231] Rohit Puri, Kang-Won Lee, Kannan Ramchandran, and Vaduvur Bharghavan. An integrated source transcoding and congestion control paradigm for video streaming in the Internet. *IEEE Transactions on Multimedia*, 3(1):18–32, March 2001.

[232] Rohit Puri and Kannan Ramchandran. Multiple description source coding using forward error correction codes. In *The 33rd Asilomar Conference on Signals, Systems, and Computers*, volume 1, pages 342–346, October 1999.

[233] Jae-Young Pyun, Jun-Suk Lee, Jin-Woo Jeong, Jae-Hwan Jeong, and Sung-Jea Ko. Robust error concealment for visual communications in burst-packet-loss networks. *IEEE Transactions on Consumer Electronics*, 49(4):1013–1019, November 2003.

[234] Hayder Radha, Mihaela van der Schaar, and Yingwei Chen. The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP. *IEEE Transactions on Multimedia*, 3:53–68, March 2001.

[235] Parameswaran Ramanathan and Kang G. Shin. Delivery of time-critical messages using a multiple copy approach. *ACM Transactions on Computer Systems*, 10(2):144–166, 1992.

[236] Kamisetty R. Rao, Zoran S. Bojkovic, and Dragorad A. Milovanovic. *Multimedia communication systems: Techniques, standards, and networks.* Prentice Hall, April 2002.

[237] Kamisetty R. Rao, Zoran S. Bojkovic, and Dragorad A. Milovanovic. Multimedia communications across networks. In *Multimedia communication systems: Techniques, standards, and networks*, chapter 6. Prentice Hall, April 2002.

[238] Theodore S. Rappaport. *Wireless communications: principles and practice.* Prentice Hall PTR, 1995.

[239] Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *SIAM Journal on Applied Mathematics*, 8:300–304, 1960.

[240] Amy R. Reibman, Hamid Jafarkhani, Yao Wang, Michael T. Orchard, and Rohit Puri. Multiple description coding for video using motion compensated

prediction. In *IEEE International Conference on Image Processing*, October 1999.

[241] Internet Traffic Report. Global Response Time for the past 30 days. http://www.internettrafficreport.com/30day.htm, May 2006.

[242] Internet Traffic Report. Global Response Time for the past 30 days. http://www.internettrafficreport.com/30day.htm, September 2007.

[243] Richard Restak. *Mysteries of the mind.* National Geographic Society, 2000.

[244] Injong Rhee. Error control techniques for interactive low-bit rate video transmission over the Internet. *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 1998.

[245] Amir Said and William A. Pearlman. A new, fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and System for Video Technology*, 6(3):243–250, 1996.

[246] Jerome H. Saltzer, David P. Reed, and David D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4):277–288, November 1984.

[247] Stefan Savage, Thomas Anderson, Amit Aggarwal, David Becker, Neal Cardwell, Andy Collins, Eric Hoffman, John Snell, Amin Vahdat, Geoff Voelker, and John Zahorjan. Detour: Informed Internet routing and transport. *IEEE Micro*, 19(1):50–59, January 1999.

[248] Stefan Savage, Andy Collins, Eric Hoffman, John Snell, and Thomas Anderson. The end-to-end effects of Internet path selection. *SIGCOMM Computer Communication Review*, 29(4):289–299, October 1999.

[249] Eric Schoonover and Stephan Beckert. Global traffic, bandwidth, and pricing trends and wholesale market outlook. Pacific Telecommunications Council (PCT'07), January 2007. www.ptc.org/events/ptc07/program/presentation/ST_BeckertSchoonover.pdf.

[250] Henning Schulzrinne. Internet technical resources. http://www.cs.columbia.edu/~hgs/internet/, September 2003.

[251] Henning Schulzrinne, Stephen L. Casner, Ron Frederick, and Van Jacobson. RTP: A Transport protocol for real-time applications. RFC 1889, The Internet Engineering Task Force, January 1996. http://www.ietf.org/rfc/rfc1889.txt.

[252] Ben Scott, Mark Cooper, and Jeannine Kenney. Why consumer demand Internet freedom - Network Neutrality: Fact vs. Fiction. http://www.savetheinternet.com/, May 2006.

[253] Andrew Secker and David S. Taubman. Highly scalable video compression with scalable motion coding. *IEEE Transactions on Image Processing*, August 2004.

[254] Tacha Serif, Stephen R. Gulliver, and George Ghinea. Infotainment across access devices: the perceptual impact of multimedia QoS. In *ACM Symposium on Applied Computing*, pages 1580–1585. ACM Press, March 2004.

[255] Sergio D. Servetto and Klara Nahrstedt. Video streaming over the public Internet: Multiple description codes and adaptive transport protocols. In *IEEE International Conference on Image Processing*, 1999.

[256] Sanjay Shakkottai and Rayadurgam Srikant. Scheduling real-time traffic with deadlines over a wireless channel. *Wireless Network*, 8(1):13–26, 2002.

[257] Yufeng Shan and Avideh Zakhor. Cross layer techniques for adaptive video streaming over wireless networks. In *IEEE International Conference on Multimedia and Expo*, volume 1, pages 277–280, 2002.

[258] Jerome M. Shapiro. Embedded image coding using zero-trees of wavelet coefficients. *IEEE Transaction on Signal Processing*, 4(12):3445–3462, 1993.

[259] Jitae Shin, JongWon Kim, and C. C. Jay Kuo. Content-based packet video forwarding mechanism in differentiated service networks. In *IEEE Packet Video Workshop*, May 2000.

[260] Mihail L. Sichitiu. Cross-layer scheduling for power efficiency for wireless sensor networks. In *IEEE Conference on Computer Communications*, volume 3, pages 1740–1750, March 2004.

[261] Deepinder Sidhu, Raj Nair, and Shukri Abdallah. Finding disjoint paths in networks. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 43–51, 1991.

[262] Thomas Sikora. Trends and perspectives in image and video coding. *Proceedings of the IEEE: Special issue on Advances in video coding and delivery*, 93(1):6–17, January 2005.

[263] Raghavendra Singh, Antonio Ortega, Lionel Perret, and Wenqing Jiang. Comparison of multiple description coding and layered coding based on network simulations. In *SPIE Conference on Visual Communication and Image Processing*, 2000.

[264] Dorgham Sisalem and Henning Schulzrinne. The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme. In *ACM International Workshop on Network and Operating System Support for Digital Audio and Video*, 1998.

[265] Dorgham Sisalem and Adam Wolisz. LDA+ TCP-friendly adaptation: A measurement and comparison study. In *ACM International Workshop on Network and Operating System Support for Digital Audio and Video*, June 2000.

[266] Dorgham Sisalem and Adam Wolisz. MLDA: a TCP-friendly congestion control framework for heterogeneous multicast environments. In *Eighth International Workshop on Quality of Service*, pages 65–74, June 2000.

[267] Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

[268] Joel Sommers, Paul Barford, Nick Duffield, and Amos Ron. Improving accuracy in end-to-end packet loss measurement. In *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 157–168, August 2005.

[269] Vineet Srivastava and Mehul Motani. Cross-layer design: a survey and the road ahead. *IEEE Communications Magazine*, 43(12):112–119, December 2005.

[270] Randall R. Stewart, Qiaobing Xie, Ken Morneault, Chip Sharp, Hanns Juergen Schwarzbauer, Tom Taylor, Ian Rytina, Malleswar Kalla, Lixia Zhang, and Vern Paxson. Stream Control Transmission Protocol. RFC 2960, The Internet Engineering Task Force, October 2000. http://www.ietf.org/rfc/rfc2960.txt.

[271] Jonathan Stone. *Checksums in the Internet.* PhD thesis, Department of Computer Science, Stanford University, August 2001.

[272] Jonathan Stone, Randall R. Stewart, and Douglas Otis. Stream Control Transmission Protocol (SCTP) checksum change. RFC 3309, The Internet Engineering Task Force, September 2002. http://www.ietf.org/rfc/rfc3309.txt.

[273] Jacob Strauss, Dina Katabi, and M. Frans Kaashoek. A measurement study of available bandwidth estimation tools. In *ACM SIGCOMM Internet Measurement Conference*, 2003.

[274] Yih-Ching Su, Chu-Sing Yang, and Chen-Wei Lee. Optimal FEC assignment for scalable video transmission over burst error channel with loss rate feedback. *Signal Processing: Image Communication*, 18(7):537–547, August 2003.

[275] Gary Sullivan and Thomas Wiegand. Video compression - From concepts to the H.264/AVC standard. *Proceedings of the IEEE: Special issue on Advances in video coding and delivery*, 93(1):18–31, January 2005.

[276] Cisco Systems. *Internetworking Technologies Handbook*. Networking Technology. Cisco Press, 4th edition, September 2003.

[277] Kun Tan, Richard Ribier, and Shih-Ping Liou. Content-sensitive video streaming over low bitrate and lossy wireless network. In *ACM International Conference on Multimedia*, pages 512–515, September 2001.

[278] Wai-Tian Tan. *Video compression and streaming over packet-switched networks*. PhD thesis, University of California at Berkeley, 2000.

[279] Wai-Tian Tan and Avideh Zakhor. Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol. *IEEE Transactions on Multimedia*, pages 172–186, June 1999.

[280] Wai-Tian Tan and Avideh Zakhor. Video multicast using layered FEC and scalable compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):373–386, March 2001.

[281] David S. Taubman. High performance scalable image compression with ebcot. *IEEE Transactions on Image Processing*, 9(7):1158–1170, 2000.

[282] David S. Taubman and Avideh Zakhor. Multirate 3-D subband coding of video. *IEEE Transactions on Image Processing*, 3:572–589, 1994.

[283] Renata Teixeira, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker. In search of path diversity in ISP networks. In *ACM SIGCOMM Internet Measurement Conference*, pages 313–318, 2003.

[284] Informa Telecoms and Media. World Cellular Information Service, December 2005.

[285] TeleGeography. Press release: TeleGeography finds global VoIP traffic slowing. www.telegeography.com/press/releases/2004-11-15.php, November 2004.

[286] Chao Tian and Sheila S. Hemami. Universal multiple description scalar quantization: analysis and design. *IEEE Transactions on Information Theory*, 50(9):2089–2102, September 2004.

[287] Michael Topic. *Streaming media demystified*. McGraw-Hill, 2002.

[288] Anne M. Treisman and Garry Gelade. A feature intergration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[289] Christos Tryfonas. *Video transport over packet-switched networks*. PhD thesis, University of California at Santa Cruz, 1999.

[290] Thierry Turletti, Sacha Fosse Parisis, and Jean-Chrysostome Bolot. Experiments with a layered transmission scheme over the Internet. Technical Report 3296, INRIA, November 1997.

[291] Vinay A. Vaishampayan. Design of multiple description scalar quantizer. *IEEE Transactions on Information Theory*, 39:821–834, May 1993.

[292] Vinay A. Vaishampayan and Sam John. Balanced interframe multiple description video compression. In *IEEE International Conference on Image Processing*, volume 3, pages 812–816, October 1999.

[293] Mihaela van der Schaar, Santhana Krishnamachari, Sunghyun Choi, and Xiaofeng Xu. Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs. *IEEE Journal on Selected Areas in Communications*, 21(10):1752–1763, December 2003.

[294] Mihaela van der Schaar and Yun-Ting Lin. Content-based selective enhancement for streaming video. In *IEEE International Conference on Image Processing*, volume 2, pages 997–980, October 2001.

[295] Mihaela van der Schaar and Sai Shankar N. Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms. *IEEE Wireless Communications*, 12(4):50–58, August 2005.

[296] John Vince and Rae Earnshaw. *Digital media: the future.* Springer-Verlag, 2000.

[297] William von Hagen. Real-time and performance improvements in the 2.6 Linux kernel. *Linux Journal*, April 2005.

[298] Benjamin W. Wah, Xiao Su, and Dong Lin. A survey of error-concealment schemes for real-time audio and video transmissions over the Internet. In *International Symposium on Multimedia Software Engineering*, pages 17–24, December 2000.

[299] Yao Wang and Shunan Lin. Error resilient video coding using multiple description motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:438–453, June 2002.

[300] Yao Wang, Michael T. Orchard, and Amy R. Reibman. Multiple description image coding for noisy channels by pairing transform coefficients. In *IEEE Workshop on Multimedia Signal Processing*, pages 419–424, June 1997.

[301] Yao Wang, Michael T. Orchard, Vinay A. Vaishampayan, and Amy R. Reibman. Multiple description coding using pairwise correlating transforms. *IEEE Transactions on Image Processing*, 10(3):351–366, March 2001.

[302] Yao Wang, Shivendra S. Panwar, Shunan Lin, and Shiwen Mao. Wireless video transport using path diversity: Multiple description vs. layered coding. In *IEEE International Conference on Image Processing*, 2002.

[303] Yao Wang, Amy R. Reibman, and Shunan Lin. Multiple description coding for video delivery. *Proceedings of the IEEE: Special issue on Advances in video coding and delivery*, 93(1):57–70, January 2005.

[304] Yao Wang, Stephan Wenger, Jiangtao Wen, and Aggelos K. Katsaggelos. Error resilient video coding techniques. *IEEE Signal Processing Magazine*, 17(4):61–82, July 2000.

[305] Yao Wang and Qin-Fan Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86(5):974–997, May 1998.

[306] Ye Wang, Wendong Huang, and Jari Korhonen. A framework for robust and scalable audio streaming. In *ACM International Conference on Multimedia*, pages 144–151, October 2004.

[307] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[308] Cheng-Yu Wei, Nevenka Dimitrova, and Shih-Fu Chang. Color-mood analysis of films based on syntactic and psychological models. In *IEEE International Conference on Multimedia and Expo*, volume 2, pages 831–834, June 2004.

[309] Stephan Wenger, Gerd Knorr, Jorg Ott, and Faouzi Kossentini. Error resilience support in H.263+. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 867–877, November 1998.

[310] Jörg Widmer, Robert Denda, and Martin Mauve. A survey on TCP-friendly congestion control. *IEEE Network*, 15(3):28–37, 2001.

[311] Stefan Winkler. *Digital video quality: Vision models and metrics*. Wiley, March 2005.

[312] DaPeng Wu, YiWei Thomas Hou, and Ya-Qin Zhang. Transporting real-time video over the Internet: challenges and approaches. *Proceedings of the IEEE*, 88(12):1855–1877, December 2000.

[313] Huahui Wu, Mark Claypool, and Robert E. Kinicki. A model for MPEG with forward error correction and TCP-friendly bandwidth. In *ACM International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 122–130, June 2003.

[314] Huahui Wu, Mark Claypool, and Robert E. Kinicki. Adjusting forward error correction with temporal scaling for TCP-friendly streaming MPEG. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 1(4):315–337, November 2005.

[315] Huahui Wu, Mark Claypool, and Robert E. Kinicki. On combining temporal scaling and quality scaling for streaming MPEG. In *ACM International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 111–116, May 2006.

[316] Ruiqin Xiong, Jizheng Xu, Feng Wu, Shipeng Li, and Ya-Qin Zhang. Layered motion estimation and coding for fully scalable 3D wavelet video coding. In *IEEE International Conference on Image Processing*, August 2004.

[317] Maya Yajnik, Sue B. Moon, James F. Kurose, and Donald F. Towsley. Measurement and modeling of the temporal dependence in packet loss. In *IEEE Conference on Computer Communications*, pages 345–352, March 1999.

[318] Zhenyu Yang and Klara Nahrstedt. A bandwidth management framework for wireless camera array. In *International workshop on Network and Operating Systems Support for Digital Audio and Video*, 2005.

[319] Paul Yao. Windows CE 3.0: Enhanced real-time features provide sophisticated thread handling. *MSDN magazine*, November 2000.

[320] Wanghong Yuan, Klara Nahrstedt, Sarita V. Adve, Douglas L. Jones, and Robin H. Kravets. GRACE-1: Cross-layer adaptation for multimedia quality and battery energy. *IEEE Transactions on Mobile Computing*, 5(7):799–815, July 2006.

[321] Wanghong Yuana, Klara Nahrstedta, Sarita V. Advea, Douglas L. Jonesb, and Robin H. Kravetsa. Design and evaluation of a cross-layer adaptation framework for mobile multimedia systems. In *SPIE/ACM Multimedia Computing and Networking*, January 2003.

[322] Richard D. Zakia. *Perception and imaging.* Focal Press, 2nd edition, 2002.

[323] Amgad Zeitoun, Zhiheng Wang, and Sugih Jamin. Rttometer: Measuring path minimum RTT with confidence. In *IEEE Workshop on IP Operations and Management*, October 2003.

[324] Wenjun Zeng and Bede Liu. Geometric structure based directional filtering for error concealment in image/video transmission. *SPIE Wireless Data Transmission at Information Systems/Photonics East*, pages 145–156, October 1995.

[325] Wenjun Zeng and Bede Liu. Geometric-structure-based error concealment with novel applicationsin block-based low-bit-rate coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:648–665, June 1999.

[326] Fan Zhai, Yiftach Eisenberg, and Aggelos K. Katsaggelos. *Handbook of Image and Video Processing*, chapter Joint source-channel coding for video communications, pages 1065–1082. Elsevier Academic Press, 2nd edition, June 2005.

[327] Fan Zhai, Yiftach Eisenberg, Thrasyvoulos N. Pappas, Randall Berry, and Aggelos K. Katsaggelos. Rate-distortion optimized hybrid error control for real-time packetized video transmission. In *IEEE International Conference on Communications*, volume 3, pages 1318–1322, June 2004.

[328] Fan Zhai, Carlos E. Luna, Yiftach Eisenberg, Thrasyvoulos N. Pappas, Randall Berry, and Aggelos K. Katsaggelos. Joint source coding and packet classification for real-time video transmission over differentiated services networks. *IEEE Transactions on Multimedia*, 7(4):716–726, August 2005.

[329] Lixia Zhang, Stephen E. Deering, and Deborah Estrin. RSVP: A new resource ReSerVation protocol. *IEEE network*, 7(5), September 1993.

[330] Qian Zhang, Wenwu Zhu, and Ya-Qin Zhang. End-to-end QoS for video delivery over wireless Internet. *Proceedings of the IEEE: Special issue on Advances in video coding and delivery*, 93(1):123–134, January 2005.

[331] Qinqing Zhang and Saleem A. Kassam. Hybrid ARQ with selective combining for fading channels. *IEEE Journal on Selected Areas in Communications*, 17(5):867–880, May 1999.

[332] Yin Zhang, Nick Duffield, Vern Paxson, and Scott Shenker. On the constancy of Internet path properties. In *ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[333] Li Zhao, Jian Huang, Yuwen He, Shiqiang Yang, and Yuzhuo Zhong. Content-based video streaming over the internet for universal multimedia access. In *SPIE Image and Video Communications and Processing*, volume 5022, pages 781–791, May 2003.

[334] Li Zhao, Qi Wang, Yuwen He, Shiqiang Yang, and Yuzhuo Zhong. A novel content-based video streaming algorithm for fine granular scalable coding. In *IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, pages 210–214, October 2001.

[335] Haitao Zheng and Jill M. Boyce. An improved UDP protocol for video transmission over Internet-to-wireless networks. *IEEE Transactions on Multimedia*, 3(3):356–365, September 2001.

[336] Johnny Zweig and Craig Partridge. TCP alternate checksum options. RFC 1146, The Internet Engineering Task Force, March 1990. http://www.ietf.org/rfc/rfc1146.txt.