Video packet priority assignment based on spatio-temporal perceptual importance

S Sadhana Reddy

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद

Indian Institute of Technology Hyderabad

Department of Electrical Engineering

July 2014

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Sadhu 17/07/2014

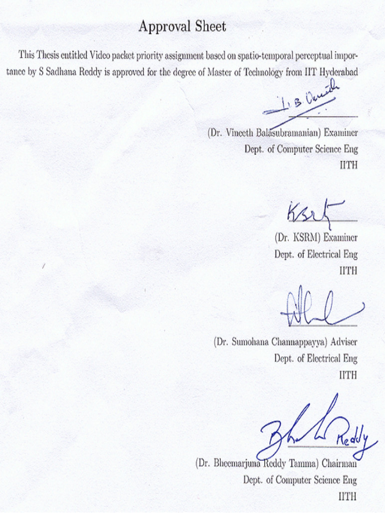
(Signature)

S Sadhana Reddy

(S Sadhana Reddy)

EE12M1028

(Roll No.)



Approval Sheet

This Thesis entitled Video packet priority assignment based on spatio-temporal perceptual importance by S Sadhana Reddy is approved for the degree of Master of Technology from IIT Hyderabad

(Dr. Vineeth Balasubramanian) Examiner
Dept. of Computer Science Eng
IITH

KSR

(Dr. KSRM) Examiner
Dept. of Electrical Eng
IITH

(Dr. Sumohana Channappayya) Adviser
Dept. of Electrical Eng
IITH

Reddy

(Dr. Bheemarjuna Reddy Tamma) Chairman
Dept. of Computer Science Eng
IITH

Acknowledgements

I would like to thank my guide Dr. Sumohana Channappayya for his guidance, creative suggestions and strong moral support. I would like to thank my labmates K. Manasa and K. V. S. N. L. Manasa Priya for their help in the intial stages of the algorithm implementation.

Abstract

A novel perceptually motivated two-stage algorithm for assigning priority to video packet data to be transmitted over the internet is proposed. Priority assignment is based on temporal and spatial features that are derived from low-level vision concepts. The motivation for a two-stage design is to be able to handle different application settings. The first stage of the algorithm is computationally very efficient and can be directly used in low-delay applications with limited computational resources. The two-stage method performs exceedingly well across a variety of content and can be used in less restrictive operating settings. The efficacy of the proposed algorithm (both stages) is demonstrated using an intelligent packet drop application where it is compared with cumulative mean squared error (cMSE) based priority assignment and random packet dropping. The proposed prioritization algorithm allows for packet drops that result in significantly lower perceptual annoyance at the receiver relative to the other methods considered.

The proposed algorithm requires no prior training with subjective scores thereby making it easier to implement and deploy. We have replaced the requirement for subjective evaluation by using objective perceptual quality metrics instead that correlate well with subjective scores. The combination of spatial and temporal features ensures good performance across a range of motion content. Also, the proposed algorithm makes minimal use of empirically determined parameters thereby making it applicable in a wide range of applications. Further, the performance of the proposed algorithm highlights the fact that perceptually motivated packet prioritisation is a promising approach to estimating the perceptual effects of packet loss.

Many cross layer techniques are lacking in an efficient priority assignment technique in the application layer and use cMSE which we have proven to be less efficient compared to our technique by deploying the proposed algorithm in a MAC centric approach for packet prioritization. So our technique can be used for packet priority assignment in application layer in the existing cross layer techniques to improve their performance.

Contents

Chapter 1

Introduction

1.1 Introduction to the problem of Multimedia Traffic management

Multimedia data has become the major component of all internet traffic and is continuing to grow at an exponential rate [1]. This growth can be attributed to the rapid advances in cellular technology, low-power device development and mobile operating system technologies. Apart from massive amounts of online content being generated by the "end user" on services such as YouTube and Vimeo, content previously considered to be restricted to radio, television, and the movie media has now moved to the internet – typical examples include popular streaming services such as Netflix and Hulu. This proliferation of multimedia content has put existing networks under tremendous stress. It is therefore imperative to manage multimedia traffic as efficiently as possible.

The problem of efficient multimedia traffic management has been widely studied from various perspectives and at different layers in the communication stack. We loosely classify the approaches as *application layer* and *cross-layer* techniques and present a brief survey of a representative set of such approaches.

1.1.1 Application Layer Oriented Perspective of the Problem

The application layer community has approached this problem as one of constrained no-reference video quality assessment and packet prioritization. The constraint is to estimate the perceptual effects of packet loss from either bitstream parsing or from partial decode of the compressed video bitstream. The estimated degradation in perceptual quality due to a packet loss is used to tag packets with appropriate priority. These tags are then used by network routers or switches to implement intelligent packet drop policies. Approaches include generalized linear models [2, 3, 4, 5], header decode based methods [6, 7] and human visual system based approaches such as those based on spatial, temporal and spatio-temporal saliency [8, 9].

Generalized linear model based methods typically extract features from the bitstream (with or without partial decoding) and combine them linearly to produce an estimate the perceptual importance of a frame. The weights associated with the features are determined by training against subjective scores of the videos from a training set. Identifying good features is paramount to these methods and several studies address this problem [10, 11].

Schier et. al. [12] present a low-complexity real-time technique for assigning priority at the macroblock level that does not require decode but only bitstream parsing. Their algorithm takes into account the macroblock partitioning mode, error propagation due to temporal dependency and error concealment strategy used by the decoder into account to obtain a distortion estimate for a macroblock. The priority assignment strategy implicitly estimates the effect of macroblock errors on perceived quality. Similar methods have been shown to be effective in [13, 14]. It should be noted that all these approaches only assume a lossy channel and do not explicitly model channel characteristics.

The SSIM index [15] and its several flavors are popular image quality assessment algorithms that have been shown to correlate well with subjective scores over a wide range of distortion types. It has been shown that visual importance pooling of the local SSIM indices significantly improves its performance [16]. Visual saliency [17] based SSIM has been shown to very effective as well [18]. Further, the SSIM index has been shown to be effective in quantifying the effects of packet loss [10, 9]. In [9], it is shown that saliency weighted SSIM index (and mean absolute difference (MAD), mean squared error (MSE)) provides good estimates of the perceptual effects of packet loss. The field of attention of the distorted video frames are estimated using Itti's saliency toolbox [17] and used to weight the overall computation of quality. This vision-inspired weighting has been shown to significantly improve perceptual quality estimation.

1.1.2 Cross-Layer Oriented Perspective of the Problem

From an information theoretic perspective, joint source-channel coding provides the optimal solution for robust multimedia transmission over noisy channels. This is however difficult to implement given the layered nature of the communication network stack. Cross-layer optimization techniques have

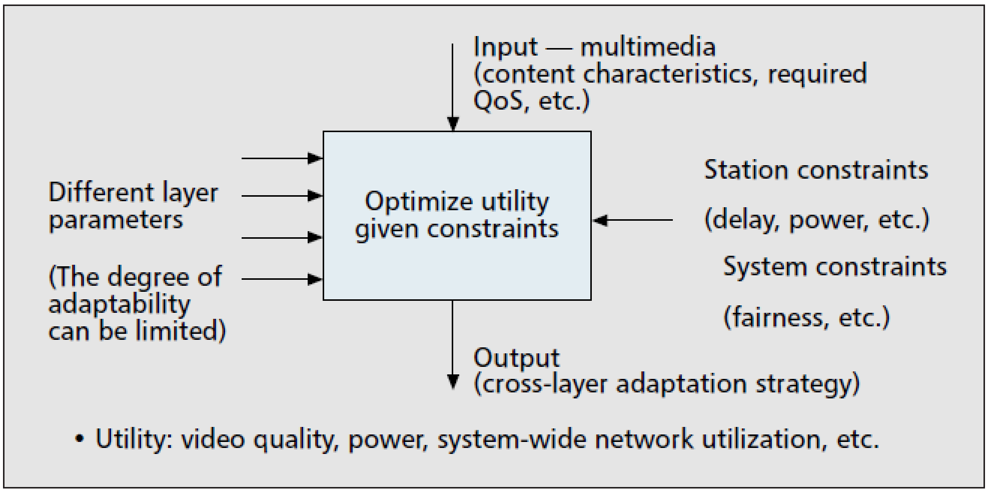


Figure 1.1: Conceptual Framework of Cross Layer Optimization [27]

been proposed to precisely address this issue and many promising solutions exist [19]. Singh et. al. propose a network-level interference shaping approach where the overall quality of experience of the received video (measured using a modified version of MS-SSIM [20]) is considered as a metric for evaluation. This metric is optimized by spreading in time (or shaping) the transmit power of interfering base stations so that jitter and packet loss at the video receiver is significantly reduced. Thakolsri et. al. [21] present a novel cross-layer optimization approach involving utility maximization where utility is defined as the temporal change in video quality. Specifically, the video SSIM (VSSIM) [22] is used as the utility or objective function and a resource allocation problem is solved. Quality-of-experience (QoE)-aware scheduling has been proposed by several authors where packet scheduling decisions are based on QoE [23, 24]. Kambhatla et. al. [25] propose a cross-layer solution at the MAC layer to find the optimal fragment size for each of the four priority classes assigned to H.264 slices. The optimization goal is to maximize goodput at known link conditions. Ha et. al. [26] provide a perceptually motivated technique for weighting the importance of frames and group of pictures (GOP). This weight is used to optimally choose a forward error correction (FEC) code at a given channel condition to provide perceptually unequal loss protection.

Given these interesting and robust solutions, it is clear that cross-layer approaches provide the way forward both in terms of approaching theoretical performance bounds and in terms of practical applicability. In the next section the cross layer optimization framework is described in detail.General cross layer framework is illustrated in Fig. 1.1.

Chapter 2

Literature Survey

2.1 Introduction to cross layer optimization problem

OSI(Open loop System Interconnection) model class for strict boundaries between communication layers. The lower layer services its immediate upper layer and the interfaces between layers are strictly defined. So eventhough the upper layer receives services from lower layer it is unaware of implementations and protocols in the lower layer. For it the lower layer is a black box with well defined output. This model's main purpose is that the implementation method and protocols in each layer can be updated without disturbing the communication system as long as the service provided by the layer to its upper layer is unaltered. But this model does not help in optimally utilising the available resources to deliver the best possible quality output to multimedia users. the shortcomings of this model and the need for cross layer optimization is explained in the following section.

2.2 Definition and Need for Cross Layer Optimization

2.2.1 Definition

Cross layer optimization is an escape from the pure waterfall-like concept of the OSI communications model with virtually strict boundaries between layers. The cross layer approach is used to optimally adapt the multimedia compression and transmission strategies jointly across the protocol stack in order to guarantee a predetermined multimedia quality at the receiver.

2.2.2 Need for Cross Layer Optimization

In recent years the research focus has been to adapt existing algorithms and protocols for multimedia compression and transmission to the rapidly varying and often scarce resources of wireless networks. However, these solutions often do not provide adequate support for multimedia applications in crowded wireless networks, when interference is high or stations are mobile. This is because the resource management, adaptation, and protection strategies available in the lower layers of the stack – the physical (PHY), medium access control (MAC), and network/transport layers – are optimized without explicitly considering the specific characteristics of multimedia applications, and conversely, multimedia compression and streaming algorithms do not consider the mechanisms provided by the lower layers for error protection, scheduling, resource management, and so on. This layered optimization leads to a simple independent implementation, but results in suboptimal multimedia (objective and/or perceptual quality) performance.

Alternatively, under adverse conditions, wireless stations need to optimally adapt their multimedia compression and transmission strategies jointly across the predetermined quality at the receiver. This scenario calls for a cross layer framework for jointly analyzing, selecting, and adapting the different strategies available at the various OSI layers in terms of multimedia quality, consumed power, and spectrum utilization. Developing such an integrated cross layer framework is of fundamental importance, since it not only leads to improved multimedia performance over existing wireless networks, but also provides valuable insights into the design of next generation algorithms and protocols for wireless multimedia systems. The cross-layer approach does not necessarily require a redesign of existing protocols, and can be performed by selecting and jointly optimizing the application layer and the strategies available at the lower layers, such as admission control, resource management, scheduling, error protection, and power control.

2.3 Cross Layer Optimization Problem

Joint Cross Layer Strategy S is defined as

$S = \{PHY_1, \ldots, PHY_{N_P}, MAC_1, \ldots, MAC_{N_M}, \ldots\}$

where N_P, N_M, N_A denote the number of adaptation and protection strategies available at the PHY, MAC, and APP layers, respectively.

Hence $N = N_P x N_M x N_A$ are the number of possible joint design strategies.

The Optimal Composite Strategy is given by following equation

$S^{opt}(x) = argmax_S Q(S(x))$

This strategy results in the best perceived multimedia Quality Q subject to the following constraints

$Delay(S(x)) \leq D_{max}$,

$Power(S(x)) \leq Power_{max}$,

system constraints such as fairness strategies and bandwidth allocation.

2.4 Challenges in solving a Cross Layer Optimization Problem

- Deriving analytical expressions for Q, Delay, and Power as functions of channel conditions is very challenging, since these functions are nondeterministic (only worst case or average values can be determined) and nonlinear, and there are dependencies between some of the strategies across layers.

- The algorithms and protocols at the various layers are often designed to optimize each layer independently and often have different objectives. Moreover, various layers operate on different units of multimedia traffic and take as input different types of information. For instance, the physical layer is concerned with symbols and depends heavily on the channel characteristics, while the application layer is concerned with semantics and dependencies between flows, and depends heavily on the multimedia content.

- The wireless channel conditions and multimedia content characteristics may change continuously, requiring constant updating of parameters.

- Formal procedures are required to establish optimal initialization, grouping of strategies at different stages (i.e., which strategies should be optimized jointly), and ordering (i.e., which strategies should be optimized first) for performing the cross layer adaptation and optimization.The selected procedure determines the rate of convergence and the values at convergence. The rate of convergence is extremely important, since the dynamic nature of wireless channels requires rapidly converging solutions.

- Finally, different practical considerations (e.g., buffer sizes, ability to change retry limits or modulation strategies at the packet level) for the deployed wireless standard (e.g., 802.11e QoS

MAC supports unequal error protection for different flows or delay awareness, unlike traditional 802.11a/b/g MAC) must be taken into account to perform the cross layer optimization.

2.5 Classification Of Cross Layer Solutions

Based on the order in which cross layer optimization is performed the possible solutions to a cross layer problem are classified as follows

- **Top-down Approach** – The higher-layer protocols optimize their parameters and the strategies at the next lower layer.
- **Bottom-up Approach** – The lower layers try to insulate the higher layers from losses and bandwidth variations.
- **Application-centric Approach** – The APP layer optimizes the lower layer parameters one at a time in a bottom-up (starting from the PHY) or top-down manner, based on its requirements.
- **MAC-centric Approach** – In this approach the APP layer passes its traffic information and requirements to the MAC, which decides which APP layer packets/flows should be transmitted and at what QoS level. The MAC also decides the PHY layer parameters based on the available channel information.
- **Integrated Approach** – In this approach, strategies are determined jointly. A possible solution to solve this complex cross-layer optimization problem in an integrated manner is to use learning and classification techniques. For this, we identify content and network features that can easily be computed and are good indicators of which composite (integrated) strategy is optimal.

2.6 Physical Layer Centric Strategy(A Bottom-up Approach)

The unpredictability of the wireless medium poses a major challenge to delivering a high quality of experience (QoE) for real-time video services. Bursty co-channel interference is a prominent cause of wireless throughput variability, which leads to video QoE degradation, even for a fixed average channel quality. [28] proposes and analyzes a network-level resource management algorithm termed interference shaping to smooth out the throughput variations (and hence improve the QoE) of video users by decreasing the peak rate of co-channel best effort users.

QoE is monitored and the existing feedback channels may be used for sending the calculated QoE to base station. Once the QoE status is known to the network, then the corresponding information can be shared among the base stations either through the backhaul or over dedicated overhead channels. In fact, information sharing among nearby base stations is already prevalent in LTE systems.The radio resource management (RRM) engine present at every base station is responsible for allocating resources in terms of bandwidth, power and time for each user. This engine is made aware of the type of traffic for each user through QoS class indicators (QCI) and QoS classes by the network. As an example, real-time video traffic would have a QCI value in the range of 1-3 whereas that for the best effort data would be in the range of 7-9. The resource allocation is then done in accordance to QoS requirements. Thus, the required power scaling can be handled similarly through a possible QoE specifier made available to RRM by the network.

Interference shaping can be used in two scenarios. One setup is where the macro base station serves the macro user streaming real time video and nearby small cells wish to use the same spectrum and hence act as co-channel interferers.Second setup is where two base stations carry a mix of real time video and bursty data over an OFDMA cellular system.

2.7 Well-known MAC Centric Approaches

A cross-layer priority-aware packet fragmentation scheme to enhance the quality of H.264 compressed bitstreams over bit-rate limited error-prone links in packet networks is proposed in [29]. In this method, the goodput, which is defined as the expected number of successfully received video bits per second (bps) normalized by the target video bit rate R bps., is derived as function of Channel BER and MAC layer fragment success rate. This objective is to find the optimal MAC layer fragment size such that goodput is maximized. Then the application layer video slices are aggregated into fragments of obtained optimal size before transmission. Here the video slice size is assumed to be fixed which can be done by tweaking the first four parameters in the Error Resilience/Slices section of the encoder.cfg file of JEG_JM 16.1.

Initially, the goodput is optimized without taking into consideration the priority/perceptual importance of the video slices. this method is called Priority agnostic optimization. When priority of video slices is taken into consideration for goodput optimization it is called priority aware optimization. In priority aware optimization, the video slices are first divided into two slice groups - low priority and high priority based on the their perceptual importance. The cMSE of the Slice loss induced video to lossless video is calculated for each slice. The slices with cMSE above the median of all slice cMSE values are assigned high priority and others are assigned low priority.

So instead of optimizing general goodput the weighted goodput which is the linear combination of individual priority goodputs is optimized. The weight for high priority goodput is computed as the ratio of mean cMSE of high priority slices to the mean cMSE of all slices in the pre- encoded video and similarly low priority goodput weight is computed.

The objective is to find the optimal fragment sizes for low priority and high priority fragments such that weighted goodput is maximized. For every second, The high priority fragments are transmitted first. The number of high priority slices generated per second is assumed to follow uniform distribution over [0,N] where N is the total number of slices generated per second.
As it is obvious, the priority aware optimization resulted in a better output video quality compared to priority agnostic optimization.

2.8 A Few Application Layer Centric Strategies(A Top-down Approach)

2.8.1 Packet Scheduling

A class of packet scheduling algorithms for video streaming over wireless channels is proposed in [30] by applying different deadline thresholds to video packets with different importance.From the viewpoint of channel status, if the channel is in good condition without errors, then it is advantageous to use EDF(earliest deadline first) criterion to send VPs in sequential order to obtain minimum average queue length in the receivers buffer. However, if the channel condition is poor with large error rates, then it is desirable to send more important VPs within GOPs first in order to achieve lower video distortion.

In the first packet scheduling algorithm, the importance of a video packet is determined by its relative position within its group of pictures(GOP).The algorithm consists of two steps
Assume F_n is to be currently on display at the receiver and $VP_{i,j}$ denotes jth video packet of frame i.

- Scan video packets $VP_{i,j}$ $(i > n)$ to choose a candidate $VP_{i,j}$, with smallest i, which is neither sent nor outstanding and satisfies the following requirement:
  $D(VP_{i,j}) > d(VP_{i,j}) + \Delta$
  where
  $D(VP_{i,j})$ denotes the time difference in seconds between current time t = n/f(frame rate) and

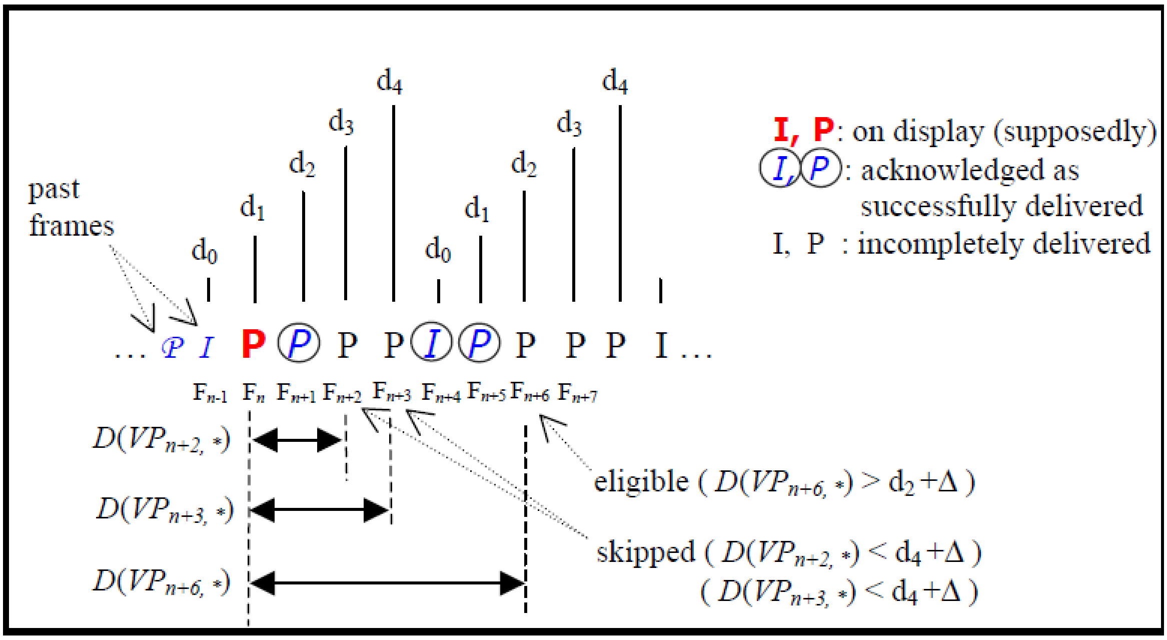

d4
d3
d2
d1
d0
d4
d3
d2
d1
d0
past frames
I, P: on display (supposedly)
(I)(P): acknowledged as successfully delivered
I, P : incompletely delivered
... P I P (P) P P (I) (P) P P P I ...
F_{n-1} F_n F_{n+1} F_{n+2} F_{n+3} F_{n+4} F_{n+5} F_{n+6} F_{n+7}
D(VP_{n+2}, *)
D(VP_{n+3}, *)
D(VP_{n+6}, *)
eligible (D(VP_{n+6}, *) > d2 +Δ)
skipped (D(VP_{n+2}, *) < d4 +Δ)
(D(VP_{n+3}, *) < d4 +Δ)

Figure 2.1: Scheduling Example [30]

the deadline of F_i, t = i/f,

Δ in seconds denotes the latency between the time the sender emits a VP and the time the VP arrives at the receiver,

deadline threshold $d(VP_{i,j}) = \frac{\beta\alpha_i}{M-1}, [sec]$ where β in second is called importance factor which determines the range of frames importance criterion is applied to, M is the GOP size and $\alpha = i \ mod \ M$.

If the receivers buffer has shorter queue length than β, then VPs to be sent have smaller $D(*)$ than their deadline thresholds, and as a result importance criterion will be applied to them. On the other hand, if the receivers buffer has larger queue length than β, then $D(*)$ of VPs to be sent is large enough and VPs are sent sequentially. β should be determined as an optimum value which depends on the receivers buffer size, initial amount of pre-roll buffer, statistics of burst error, etc.

- Rescan $VP_{i,j}, n < k < i$, to see if there is any VP with higher or the same importance as candidate $VP_{i,j}$.Then, among such VP(s), the sender finally chooses a VP with shortest deadline among them.

The procedure is illustrated in Fig 2.1.

Frame based scheduling is extended by adding motion texture discrimination.MPEG-4 supports data partitioning mode by separating the motion and the texture by motion marker inserted between mo-

tion and texture information within a VP. If the texture information is lost, this approach utilizes the motion information to conceal errors. Using this feature, the data in each frame with motion and texture blocks is rearranged; all motion vector fields are gathered in the motion block, and all DCT coefficient fields in the texture block. For frame F_i, the motion block is then divided into video packets denoted by $VP_{i,j}^{(m)}$ s and the texture block into$VP_{i,k}^{(t)}$ s. The deadline thresholds to the VPs are assigned as follows:
$d(VP_{i,j}^{(m)}) = \frac{\beta_m \alpha_i}{M-1}$
$d(VP_{i,k}^{(t)}) = \frac{\beta_t \alpha_i}{M-1}$
where β_m and β_t denote importance coefficients for motion and texture, respectively. $\beta_m < \beta_t$ to assign much lower priority to texture than motion.
As it is obvious frame based scheduling performs better than EDF and motion-texture based scheduling performs better than EDF and frame based scheduling in terms of video quality at the end user.

2.8.2 Cross Layer Perceptual ARQ Algorithm

An algorithm that combines applicationlevel information about the perceptual and temporal importance of each packet into a single priority value which drives packet selection at each retransmission opportunity is proposed in [31]. Hence, only the most most perceptually important packets are retransmitted, delivering higher perceptual quality and less bandwidth usage compared to the standard 802.11 MAC-layer ARQ scheme.

This ARQ scheme uses the IP-UDP-RTP/RTCP protocol stack. The algorithm used by the sender to implement the retransmission policy is based on a retransmission buffer RTX_{buf} . When a packet is sent, it is placed in the RTX_{buf} , waiting for its acknowledgement, and marked as unavailable for retransmission.The receiver periodically generates RTCP receiver reports (RR) containing an ACK or a NACK for each transmitted packet. A NACK is generated when the receiver detects a missing packet by means of the RTP sequence number. When an ACK is received, the corresponding packet in the RTX_{buf} is discarded because it has been successfully transmitted. If a NACK is received, the packet is marked as available for retransmission. Packets belonging to the RTX_{buf} that will never arrive at the decoder in time for playback are discarded.

Let B_{GOP} be the bandwidth needed to transmit the current GOP and B_{max} the maximum amount of bandwidth granted to the transmission. N_{rtx} retransmission opportunities are available for the current GOP, where $N_{rtx} = (B_{max} B_{GOP})/S_{pck}$ and S_{pck} is the average packet size. The time instants corresponding to the retransmission opportunities are determined as follows. The total size of each frame is first computed and then the smallest one is identified. The time instant of the first retransmission opportunity is set to be midway between the time instant of the first packet of the smallest frame interval and the last packet of the previous frame. The procedure is repeated until N_{rtx} opportunities have been determined, considering at each step the opportunities filled by packets of size S_{pck}.When a retransmission opportunity approaches, a priority function is computed for each packet marked as available in the RTX_{buf} and the one with the highest priority is transmitted. The priority function is given by

$V_{i,n} = D_{i,n} + wK\frac{1}{\Delta_{t,n}}$

where

$D_{i,n}$ is the distortion impact given by cMSE, w is weight which is used to control the relative importance of the perceptual and temporal terms, K is the product of mean distortion and receiver buffer length and $\Delta_{t,n}$ is the distance from deadline.

As it is obvious this ARQ scheduling performs better than standard 802.11 MAC-Layer ARQ in terms of video quality at the end user.

2.9 Video Packet Prioritization

The priority assignment to video packets was an essential step in all the algorithms mentioned in previous sections. cMSE was used by most of the algorithms for priority assignment but cMSE does not reflect the perceptual importance of a packet. So if a better packet priority assignment method is used instead of cMSE the above algorithms perform much more effieciently.

Generalised Linear Model(GLM), Classification and Regression Trees(CART) and 6 Stage approach are some of the popular methods for packet priority assignment using set of features.

subsectionFeature Extraction Certain features of a given video content clearly represent the perceptual importance of the video content. If these features are identified and extracted for each video packet then based on the magnitudes of these features for each packet the perceptual importance of that packet can be determined. Some such features are listed below

- **Height** at which the video content is present in the frame when it is decoded. Generally objects in the middle and top portions of the frame are given more attention by the user.

- When a video packet is lost the number of frames(**Duration**) which suffer a distortion because of it. The more is the error propogation the more important is the packet.

- The **Average Residual Energy** obtained from residual coefficients of all the macroblocks in the video packet. High texture content leads to high residual energy and texture masking may reduce the visibility of packet loss.

- The **Camera Motion** like panning, zooming etc,.Viewers are likely to follow, or track, consistent camera motion. This will enhance the visibility of temporal glitches.
- The **Mean Motion vector** of all the macroblocks in the video packet. High motion content implies the more perceptual temporal impairment results from the loss of this packet.
- The **Distance From SceneCut** is the distance of the video packet content from the scene cut in terms of display time. The packet losses near scene cut are masked and hence are not visible to the users.
- **SMSE** which measures the cMSE between saliency maps of original and loss impaired frames only in the position where loss happens and averaged over time and **STV** which measures temporal variation of the saliency map of loss-impaired frames.It is discovered that packet losses not only distort the video frames but also alter the distribution of salient regions across the affected frames spatially and temporally. It is also observed that packet losses are more visible in videos where the saliency map changes rapidly in time.

In addition to the above mentioned features many more features are known that are representative of the perceptual importance of the video packets [3, 32].
Some of these features can be obtained by parsing the bitstream whereas some need decoding like SMSE and STV.

2.9.1 GLM

Isolated packet losses are induced in the given set of videos and subjective evaluation of these losses is done by 12 users and the packet loss visibility of each lost packet(ρ) is given by the fraction of number of users who could perceive the loss. The features are extracted for these lost packets.
If the number of packets are N and the number of features are P then the Generalised Linear Model can be represented as
$g(\rho_i) = \gamma + \sum_{j=1}^{P} x_{ij}\beta_j$
where
g(.) is called link function assumed to be logit function given by $g(\rho) = log\left(\frac{\rho}{1-\rho}\right)$
$\beta_1, \beta_2, \beta_3, \ldots, \beta_p$ are coefficients of features and γ is the constant term which are to be estimated from data.

p_i is the packet loss visibility computed earlier from subjective evaluation for the ith packet.

To obtain the model coefficients for considered factors, an iteratively re-weighted least-squares technique is used to generate a maximum-likelihood estimate. The statistical software R is used for model fitting and analysis.

A model is trained on a fraction of the data (training set) and then tested using the remaining data points (testing set). A partition like this is known as a fold, and we repeat for different folds with different training and testing partitions of the data. The method discussed above(GLM) is applied to estimate the model coefficients from the training set for given factors, and then the performance error of the fitted model in the jth fold using the testing set is evaluated as follows:

$q_j = \frac{1}{3} \sum_{k=1}^{3} \left[\frac{1}{N_k} \sum_{ithpacketlossintestingsetk} (p_i - \bar{p}_i)^2 \right]$

where $\bar{p}_i$ is the predicted fraction of viewers who saw the ith packet loss, and N_k is the number of samples in the testing dataset k.

A four-fold cross-validation is chosen: the fitting process is done for a total of four times with four different folds, therefore producing 4 fitted models and q_j , $j = 1, 2, 3, 4$. This four-fold procedure is repeated four times with four different random seeds. The average performance error of these sixteen models is defined as

$Q = \frac{1}{16} \sum_{r=1}^{4} \sum_{j=1}^{4} q_j^r$

where the superscript r stands for the rth random seed.

For factor selection, Q is used to decide if a specific factor is significant and should be included in the model: for each considered factor added to the model, a Q is calculated by the 4-seeds-4-folds GLM modeling process. A factor is included only if the model with that factor included has smaller Q than the model without that factor. By the same idea, factors are excluded from the model if it has lower Q without them. To obtain the factor coefficients, the fitting from the seed that achieved the lowest performance error is used.

Given the set of features training algorithms other than GLM like CART [33] and 6 stage approach [5] can be used for training.

Once the coefficients are obtained given a test video packet its features are extracted and the inner product of the feature vector with coefficient vector gives the value of logit function from which the packet loss visibility ρ can be obtained. If ρ is less than 0.5 then packet is given low priority otherwise high priority.

Chapter 3

A Novel Video Packet Prioritization Algorithm

3.1 Contributions

GLM, CART and 6 stage approach need training for which subjective evaluation of training data is required which is a complex and time consuming process. So we propose a synthesis by analysis method to assign priority.We present a novel video packet priority assignment solution based on spatio-temporal perceptual importance estimation. This contribution can be classified as an *application layer* technique that is closest in philosophy to the works in [2, 3]. The first and foremost contribution of this work is that it is completely automated. Several application layer techniques [2, 3, 4] rely on the subjective evaluation of the effects of packet loss to train weights of linear models and choose thresholds. In this work, we eliminate this requirement by a careful choice of no-reference objective algorithms for the estimation of spatio-temporal perceptual quality. Importantly, we demonstrate that the elimination of the requirement for subjective evaluation does not result in a degradation of system performance.

The second contribution of this work is the adaptation of perceptual temporal quality metric (PTQM) [34] to the context of video packet prioritization. PTQM is a compressed domain video quality assessment technique that provides an estimate of temporal degradation caused by consistent and inconsistent frame dropping. In its original form, PTQM cannot be directly applied to measure the impact of packet loss since it attempts to estimate temporal quality for the entire video sequence. We define the temporal fluidity break measure (TFBM) that is inspired by the PTQM to estimates temporal significance at the frame level in the video. To the best of our knowledge, this is the first application of the PTQM in a multimedia communication framework.

In addition, we introduce few new parameters. All parameters in this use previously determined values (for e.g., in PTQM) and work well in the current setting. For TFBM we have chosen the threshold for packet prioritization as 1 – corresponds to no temporal distortion. The thresholds for the saliency weighted SSIM and cMSE are computed using local statistics and therefore data-dependent.

The efficacy of the proposed method is demonstrated by comparing it with existing priority assignment techniques using a packet loss experiment that measures the perceptual quality of the received degraded video.The problem is formulated in Section 3.2 and the proposed algorithm is presented in Section 3.3. Results are presented and discussed in Section 3.4

3.2 Problem Statement and Assumptions

The problem to be addressed is formalized as follows. Given a compressed video bitstream that is assumed to be in a network friendly form, how is priority assigned to individual packets such that it is representative of the packet's perceptual importance. The problem is addressed under two different settings: a) when decoding is not permitted and b) when decoding is permitted. The first setting reflects a scenario where priority assignment must be performed real-time (or faster) such as at a router where computational resources are limited. The second setting applies to the situation at a video server where user uploads typically happen in a non-real-time fashion and where significantly higher computational resources are available. The ultimate goal of this prioritization problem is to facilitate "perceptually-optimal" packet dropping policies in case of network congestion.

We assume without loss of generality that a NAL unit is packetized into a frame. This assumption is made to facilitate easier implementation and has been previously made in the literature [35]. With this assumption, we use the term packet and frame interchangeably in the rest of the work. For easier performance evaluation, we assume that a typical GOP contains only I frame.

3.3 Proposed Algorithm

We propose a two-stage algorithm for the assignment of priority to packets based on their temporal and spatial perceptual importance. These stages are detailed in the following subsections.

3.3.1 Stage 1: Temporal Perceptual Importance

Studies of the human visual system hypothesize that the eye perceives motion by inferring from the trajectory of moving objects or motion flow in a time sequence of two dimensional images formed on the retina [36]. Optical flow estimates are formed in the visual cortex and motion is inferred from these estimates. Deviations in motion trajectory or optical flow from a reference or expected path (of smooth flow) is readily perceived by the eye as has been demonstrated by the MOVIE index – the current state-of-the-art video quality assessment algorithm [37].

In the current context, optical flow is approximated by block motion vectors and the effect of packet loss on motion information is used to estimate the temporal perceptual importance of that packet. For e.g., if frames in a video with large motion content are lost, the resulting temporal distortion is easily perceived by the eye. So, we assign a temporal importance to each frame based on its motion content. The temporal importance of a frame is determined by comparing its motion content against a threshold. The methodology behind the choice of the threshold is inspired by the perceptual temporal quality metric (PTQM) [34]. We define the temporal fluidity break measure (TFBM) that uses features from the PTQM and quantifies temporal importance at a frame level. For completeness, we briefly outline the PTQM followed by a detailed description of the TFBM.

Perceptual Temporal Quality Metric (PTQM)

PTQM is a temporal quality metric for compressed video which accurately estimates the perceived temporal degradation introduced by both consistent and inconsistent frame dropping.

The dropping severity estimator s is computed to determine the number of consecutive frames that have been dropped. Even for the same amount of dropping severity the viewer perceives different levels of distortion, which is dependent on the motion activity present in the lost frames. A motion activity estimator is computed for the lost frames and is used to adjust the dropping severity level such that it reflects the amount of perceived distortion.

Temporal fluctuation estimator takes into account the fact that inconsistent frame droppings are perceptually more disturbing compared to consistent frame droppings and assigns a temporal quality fluctuation weight(TQF) to each dropping severity accordingly. TQF weight is normalized by a factor which is dependent on the frame rate of the video. For every scene temporal quality score is calculated by averaging the dropping severities weighted with output of temporal fluctuation estimator. The temporal quality of the whole sequence is given by averaging the quality scores of all the scenes.

Temporal Fluidity Break Measure (TFBM)

We denote the motion vector of a macroblock by $MV = (MV_x, MV_y)$, and compute its magnitude ($\sqrt{MV_x^2 + MV_y^2}$) for all the macroblocks in a packet. The average motion content of the packet is given by the average of the motion vector magnitudes of all the macro blocks in that packet. This mean motion vector magnitude is normalised such that it lies in the interval [0,10] by using the following formula (for frame k):

$$mmv_k = \frac{\sum_{i=1}^{\#\text{ macroblocks in frame } k} MMV_{ik}}{\max_{j \in \#\text{ frames in video}}\{\sum_{i=1}^{\#\text{ macroblocks in frame } j} MMV_{ij}\}} * 10, \tag{3.1}$$

mmv_k is the normalized mean motion vector, MMV_{ik} is the mean motion vector of macroblock i in frame k and MMV_{ij} is the mean motion vector of a macroblock i in frame j. The TFBM for frame k is given by

$$T_k = 1 - \left[\gamma.s^{\alpha - mmv_k}\right], \tag{3.2}$$

where $s = (1/R) * K$ is the dropping severity, R is frame rate of the video and K is a constant which we introduced so that the dynamic range of T_k is increased which in turn helps with better priority assignment, $\alpha = 11.5$ which is empirically determined in [34] and was found to work well in our application as well,

$$\gamma = \begin{cases} 1 & mmv_k > 4 \\ 0 & otherwise. \end{cases} \tag{3.3}$$

The threshold value of 4 was chosen empirically after it was found that a linear mapping of mean motion vectors of all frames of a video (for several test videos) to the interval [0,10] range resulted in the average of the mean motion vectors of all frames to be approximately 4.

From the expression for s it can be seen that as R increases the value of s decreases. This implies that significance of losing a frame in a high rate video is less compared to the significance of losing a frame in low rate video which is true in general. This is complemented by the fact that losing a frame with high motion content is more significant than losing a frame with low motion content by exponential raise of s by the term $\alpha - mmv_k$.

From (3.2) it is clear that only motion vector information is required to compute T_k. This information can be found by parsing the bitstream, thereby making it fast and easy to implement. Specifically, JEG JM 16.1 codec generates (by only parsing the bitstream when) an information file containing information of each NAL unit like the slice number, slice type, macroblock number, macroblock partition mode, macroblock position in the frame, motion vector (MV_x, MV_y) values, residual error DCT coefficients etc. We use this information to compute T_k for all the frames in a video.

Priority Assignment

The proposed packet priority assignment algorithm based on temporal importance estimation is summarized in Algorithm 1. For every packet in the video, its TFBM value T_k is computed and compared against a threshold τ_t. If T_k is lower than τ_t, then its priority is set to high (or 1) and to low (or 0) otherwise. Thus every packet is labeled or assigned priority using TFBM. In our work, τ_t was chosen to be 1 to highlight the importance of break in temporal fluidity on perception. In other words, TFBM is 1 when there is no temporal distortion due to frame loss.

```
Data: H.264 bitstream
Result: Packet priority assignment based on temporal importance
parse bitstream;
initialize packet count $k = 0$;
while Packets not exhausted do
    compute temporal fluidity break $T_k$ for current packet;
    if $T_k < \tau_t$ then
        set packet priority to 1;
    else
        set packet priority to 0;
    end
    increment packet count $k = k + 1$;
end
```

Algorithm 1: Priority assignment using TFBM.

3.3.2 Stage 2: Spatial Perceptual Importance

Saliency weighted SSIM index

As mentioned in Section 1.1.1, saliency weighted SSIM has been shown to work well in the context of quality assessment of videos subject to packet loss [9]. In this work, we propose the use of a saliency weighted SSIM index to measure spatial quality as well. We would however like to point out two subtle differences with the work in [9]. First, we do not assume the availability of the pristine reference video. Instead we use the decoded video without any packet loss as the reference. The saliency map is computed using Itti's saliency toolbox [17]. The saliency map is first computed for the decoded video (without inducing any packet errors) and the implementation is summarized in Algorithm 2.

```
Data: H.264 bitstream
Result: Frame-wise saliency map
Decode bitstream;
while Frames not exhausted do
    compute and save saliency map;
end
```

Algorithm 2: Computation of frame-wise saliency map.

After computing the saliency map, it is used to compute the spatial quality measure of a frame S_k by weighting SSIM index for that frame and is given by:

$$S_k = \sum_{i=1}^{N} w_i SSIM_{i,k}, k \in \{0, \ldots, F-1\}, \tag{3.4}$$

where the weight w_i for a window is computed as:

$$w_i = \frac{\mu_i}{\frac{1}{N}\sum_{j=1}^{N}\mu_j} i, i \in \{0, \ldots, N-1\}, \tag{3.5}$$

where μ_i is the average saliency value of window i, F is the total number of frames and N is the number of distinct blocks in a frame over which local SSIM is computed. It is to be noted that the video decoded without any induced packet errors is used as the "reference" in the computation of the SSIM index.

The flowchart of the second stage of the algorithm is shown in Algorithm 3 and detailed next. The video packet corresponding to the k^{th} frame is dropped and the resulting distorted bitstream is decoded to get the error concealed video. If the frame dropped is encoded as a B frame then it is extracted from the decoded video and saliency weighted SSIM is computed to estimate the perceptible spatial distortion present in the frame even after error concealment is performed by the

```
Data: H.264 bitstream
Result: Packet-wise spatial importance score
initialize packet count $k = 0$;
while each frame loss effect not computed do
    induce $k^{th}$ packet loss;
    decode lossy video;
    if Lost frame type $P$ then
        compute and save saliency-based spatial importance $S_k$ considering error propagation;
    else
        compute and save saliency-based spatial importance $S_k$;
    end
    increment packet count $k = k + 1$;
end
```

Algorithm 3: Computation of frame-wise saliency-based spatial importance.

decoder. If the frame dropped is a P frame then the dropped frame and the next 12 frames in the decode order are extracted from the video and saliency weighted SSIM is computed for each of these frames and average of these SSIM values is calculated and assigned to S_k.

Spatio-Temporal Packet Prioritisation

The temporal importance T_k given by the TFBM defined in (3.2) is computed for the k^{th} frame. In case of an implementation that uses Stage 1 alone, T_k is compared with a threshold τ_t and the packet is prioritized as 0 if $T_k < \tau_t$ and 1 otherwise. In the two-stage method, T_k is computed first.

```
Data: H.264 bitstream
Result: Packet priority assignment based on spatio-temporal importance
initialize packet count $k = 0$;
use previously computed $T_k, S_k$;
while NAL units not exhausted do
    if $T < \tau_t$ OR $S < \tau_s$ then
        set packet priority to 1;
    else
        set packet priority to 0;
    end
    increment packet count $k = k + 1$;
end
```

Algorithm 4: Spatio-temporal priority assignment.

Subsequently, the spatial importance S_k of the k^{th} frame is computed using the saliency weighted SSIM index defined in (3.4). An important consideration for spatial importance calculation is the propagation of spatial artifacts to subsequent frames due to inter-dependency of frames resulting from compression. The method adopted to handle error propagation was described in Section 3.3.2.

The temporal and spatial perceptual importance values are combined to assign packet priority using the function

$$P_k = \mathbf{1}((S_k < \tau_s) \bigcup (T_k < \tau_t)), \tag{3.6}$$

where $\tau_s = \mu + \sigma/2$, μ is the mean and σ is the standard deviation of the aggregate saliency-weighted SSIM values for entire video, $\bigcup$ is the union operator, and $\mathbf{1}()$ is the indicator function. As mentioned previously, $\tau_t = 1$. The choice of the threshold is to assign priority relative to the average saliency-weighted SSIM score for the entire video. The priority assignment algorithm is summarized in Algorithm 4.

In the discussion so far, the importance of each packet was considered in isolation. However, packet losses are typically bursty. Even though some frames have insignificant content their loss might contribute to significant distortions when they are lost in a group. One such simple case is when they are adjacent frames. To minimize adjacent frame drops, we take care that not more than 5 frames in a row are assigned low priority. This window size can be varied base on the motion content and the amount of dissimilarity between adjacent frames in a video.

For computational ease, each frame is enclosed in a single video packet. But our method can be applied to the scenario where a frame is divided into number of slices and spread across more than one video packet. In that case the our algorithm gives the spatio-temporal importance of that segment of the frame present in the video packet and priority is assigned to that packet accordingly. Since our priority is a binary in nature it is a single bit which can be accommodated in the header of the video packet (RTP packet) so that by parsing the header the network node can know the priority of the packet which influences the packet dropping decision made by the node.

3.4 Results and Discussion

The proposed algorithm is evaluated using three experiments and compared with a priority algorithm based on cumulative MSE and the case where packets are randomly dropped. The experiments, the dataset, and the results are presented in the following.

3.4.1 Experiments

Packet Loss Rate

To validate the proposed algorithm, we implement two packet dropping scenarios by making modifications to the rtp_loss code of a reference implementation of the H.264 codec (JEG JM 16.1) [38]. In the first scenario, packets are dropped randomly to meet a packet loss rate (PLR) constraint which is dictated by network conditions. In the second scenario, a priority file is given as input and contains information about the priority of packets. To meet the given PLR constraint, packets with zero priority are dropped first and packets with priority 1 are dropped only if the given PLR cannot be achieved even after exhausting all the zero priority packets. In our experiments, we assume that the network behaviour is mostly good and choose PLRs of 5% and 7.5%.

The proposed algorithm's performance is compared with random packet dropping and a cumulative MSE (cMSE) based packet prioritization method. To implement cMSE based prioritization, a threshold on the cumulative MSE of a packet is chosen such that on average, the packet priority statistics of the proposed method is satisfied. For CMSE based prioritization, the packet priority assignment function is given by

$$P_k = \mathbf{1}(cMSE_k > \tau_c), \tag{3.7}$$

where $\tau_c = \mu + \sigma/2$ here μ is the mean and σ is the standard deviation of the aggregate cMSE values for entire video. The motivation for this threshold is that it is analogous to the threshold chosen for the estimation of spatial importance in Stage-2 of the proposed algorithm. Further, this threshold results in roughly similar histograms of packet priorities as the proposed algorithm for a majority of the test videos.

Constrained Bandwidth using NS2

As observed in [2], packet loss rate experiments do not necessarily reflect a realistic scenario due to variable packet sizes. To evaluate the proposed algorithm in a realistic setting, we also conducted an experiment that simulates a bandwidth constrained data link. This is implemented using a simple bottleneck line connecting the source and destination nodes using the network simulator NS2 [39]. The network topology for this experiment is shown in Fig. 3.1.

The encoder is directed to produce the RTP video packets at a roughly constant bit rate of 1Mbps.The bottle neck link from Node A to Node B has channel bit rate which is less than the video bit rate and hence the buffer at node A overflows since the output link rate is lower than input video bit rate leading to packets being dropped from the buffer. The packets are dropped randomly from the buffer in case of random dropping. In case of cMSE based prioritization and the proposed algorithm based prioritization the low priority packets are dropped first and when the congestion is not still cleared then high priority packets are dropped. The distorted bitstream received at Node B is decoded and the quality of this video is used to judge the effectiveness of the proposed prioritization algorithm.The buffer size at node A and bottle neck link bit rate are variable parameters which decide the number of bits lost when the video packets are transmitted across this link. For different values of buffer sizes and Bottle neck link rates the experiment is performed and average VQM score of the videos at Node B are noted. As with the PLR case, the proposed algorithm is compared with cMSE based prioritization and random packet dropping.

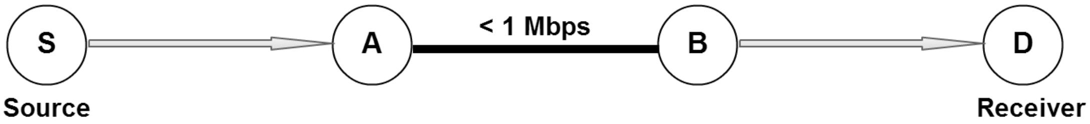


Figure 3.1: Network topology for the constrained bandwidth experiment.

Comparison with Cross-Layer Approach

In order to demonstrate the usefulness of the proposed two-stage perception inspired packet prioritization algorithm in a cross-layer optimization framework, we designed the following experiment. This experiment is also designed to demonstrate that the proposed method work well even when packetization happens at the slice level.

Typical cross layer optimization approaches adapt the lower layer parameters based on the importance of the incoming video packets from the application layer in order to achieve better quality video, use MSE or its variants to assign priority to the video packets in the application layer. This results in priority assignment which does not necessarily reflect the perceptual importance of the packets [25]. As mentioned in Section 1.1.1, Kambhatla et. al. [25] present a cross-layer optimized solution for finding the optimal fragment size given packet priority (assigned using cMSE), channel bitrate, bit error rate so that the overall goodput (defined as the number of successfully received video bits per second to the number of video bits generated per second) is maximized. In our experiment, we simply replace cMSE with SSIM in the packet priority assignment stage.

Specifically, we have set a threshold on the slice size as 150 bytes because of which each frame is split into multiple slices. A slice loss is induced in the bitstream and the resultant distorted bitstream is decoded. For P type Slice loss, SSIM is computed for the region affected in current frame as well as the regions affected in next 12 frames in decode order in order to take error propagation into account and averaged. For B slice loss, SSIM is computed for only the region affected in current frame. For slice losses where SSIM is 1 priority is set to 0 otherwise to 1. We simulate cMSE based priority assignment by using a threshold that is set to the mean of cMSE values for all the frames in the video. We show that the SSIM-based prioritization method outperforms the cMSE based technique. The VQM and PSNR scores for this method when cMSE and SSIM prioritisation schemes are used are listed in tables IV and V respectively.

Since our two-stage algorithm uses a combination of TBFM and saliency weighted SSIM which reflect the perceived temporal impairments and the perceived spatial distortion very well respectively, we hypothesize that our algorithm serves as a better packet prioritisation scheme than simple SSIM and hence would give better results when used in the cross layer optimisation methods in place of cMSE and SSIM.

3.4.2 Dataset

The robustness of the algorithm is tested by using a dataset composed of videos with varied motion content like camera zooming, panning, scene cuts, fast motion etc. The Container sequence in our dataset is a good example of slow/still motion, while the Football sequence has high motion content and the remaining sequences represent medium motion. The Foreman sequence has scene cuts, and camera panning and zooming are present in the Mobile and Flower sequences respectively. Our dataset includes 7 YUV 4:2:2 videos with a spatial resolution of 352×240 and a frame rate of 30fps encoded using H.264/AVC. I-B-P GOP structure with a single I frame is used. Each RTP packet in the bitstream contains a frame. The decoder uses frame copy type of error concealment. Packet priority assignment for each of these videos is done using the proposed algorithm and the cMSE-based method.

3.4.3 Results

The results of the above experiments are presented and evaluated next. Recalling from Section 3.3.2, a SSIM-based spatial quality evaluation method was used to estimate spatial importance. For performance evaluation to be unbiased, we purposefully wanted to avoid using SSIM-based or SSIM-inspired quality assessment algorithms to measure perceptual quality. Video Quality Metric (VQM) [40], a state-of-the-art full reference video quality assessment metric is used to evaluate the perceptual quality of the received video. Specifically, we used the reduced reference calibration version 2 (Calibration Selection) with fast low bandwidth model (model Selection) of the BVQM software [41].

The PLR experiment (Section 3.4.1) is performed at two loss rates (of 5% and 7.5%) that we feel are representative of fair network conditions. At each loss rate, the packet drop experiment is performed 10 times for each of the 7 test videos and for each dropping policy. The average VQM scores over the 10 trials for the 7 videos for different packet dropping policies are listed in Tables 3.1 and 3.2. From Table 3.1, it is clear that at a PLR of 5%, both Stage-1 and Stage-2 of the proposed algorithm outperform the other policies for a majority of the videos. Further, Stage-2 of the proposed algorithm clearly outperforms all other priority assignment policies. From Table 3.2, the trend is similar at a PLR of 7.5% where Stage-2 of the proposed method wins for a majority of the videos. Also, Stage-1 of the proposed algorithm easily outperforms the random packet dropping policy at both PLRs. We specifically mention this case since both these policies (Stage-1 and random) can be implemented in a real-time setting and do not require decoding.

Table 3.1: Packet loss rate experiment at a PLR of 5%. Algorithms evaluated using VQM (lower is better).

Clip	Cumulative MSE	Random	Proposed solution - Stage 1	Proposed solution - Stage 2
Carphone	0.4184	0.2771	0.3992	**0.0392**
Mobile	0.4507	0.5264	0.5848	**0.0492**
Foreman	0.4638	0.5560	0.4920	**0.2454**
Flower	0.5293	0.6639	0.0495	**0.0444**
Container	0.0344	0.1592	0.0275	**0.0273**
Hall monitor	0.3584	0.4234	0.3578	**0.2184**
Football	0.7351	0.6513	0.5938	**0.4839**

Table 3.2: Packet loss rate experiment at a PLR of 7.5%. Algorithms evaluated using VQM (lower is better).

Clip	Cumulative MSE	Random	Proposed solution - Stage 1	Proposed solution - Stage 2
Carphone	0.4672	0.5796	0.4875	**0.4123**
Mobile	**0.0420**	0.5581	0.5917	0.0422
Foreman	0.5910	0.5939	0.5263	**0.2994**
Flower	0.6797	0.6742	0.0619	**0.0561**
Container	**0.0273**	0.1986	0.0278	0.0282
Hall monitor	**0.0409**	0.5271	0.4414	0.4014
Football	0.6559	0.6803	0.6166	**0.5267**

The results of the constrained bandwidth experiment (Section 3.4.1) are presented in Table 3.3. The average VQM scores for the four policies under consideration are shown in Fig. 3.2. From Table 3.3, it is clear that both stages of the proposed algorithm outperform the other dropping policies.

Table 3.3: VQM Scores for NS2 simulations. Algorithms evaluated using VQM (lower is better).

Clip	Cumulative MSE	Random	Propose solution - Stage 1	Proposed solution - Stage 2
Carphone	0.1905	0.2787	0.2397	**0.0518**
Mobile	0.0343	0.3522	0.0325	**0.0319**
Foreman	0.7923	0.6424	0.6298	**0.4323**
Flower	0.7235	0.1022	**0.0702**	0.0808
Container	0.0229	0.2251	**0.0122**	0.0209
Hall Monitor	**0.0368**	0.4691	0.2007	0.3032
Football	0.3215	0.4351	**0.0604**	0.3624

The result of the third experiment outlined in Section 3.4.1 is given in Tables 3.4, 3.5. This experiment was performed mainly to demonstrate that the proposed method works even when the assumption of one frame per packet is removed and that our perceptually motivated algorithm does indeed result in improved perceptual quality (as measured by VQM).

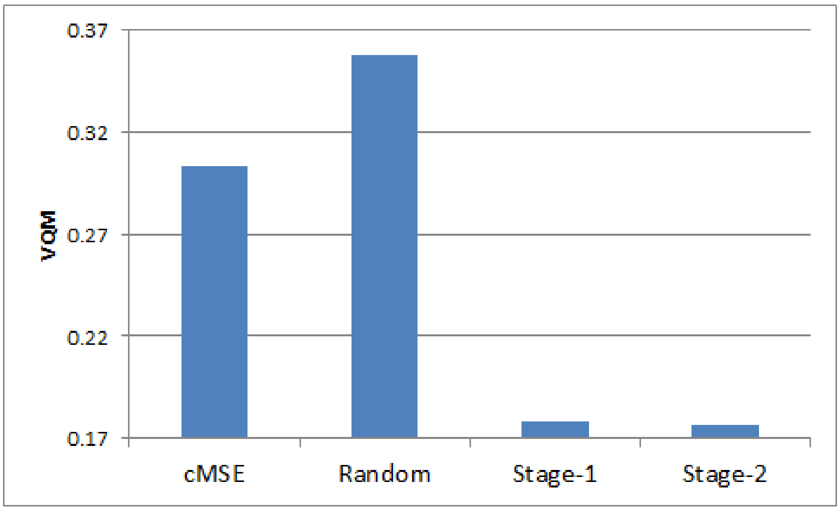


Figure 3.2: Average VQM Scores for NS2 simulation (lowerVQM score is better)

Table 3.4: VQM Scores for 1 % PLR

Clip	Cumulative MSE	SSIM
Carphone	0.1040	0.1037
Flower	0.0638	0.0488
Football	0.1213	0.0872

3.4.4 Discussion

As shown by the tables and plots in the previous section, the proposed algorithm performs better than the random dropping policy and cMSE based policy for a majority of the videos. It must be noted that the random and cMSE policies were chosen as competing policies since they compare with Stage-1 and Stage-2 of the proposed algorithm (respectively) in terms of computational complexity and the requirement for a “reference”. The “reference” in our experiments was the decoded bitstream when there was no packetloss.

The fact that the proposed algorithm performs better than the random dropping policy shows that the prioritisation is indeed effective. Instead of dropping packets randomly, where there is a high chance that perceptually important packet might be dropped, packets of low priority (low perceptual importance as marked by our algorithm) are dropped which assures that the output video quality is not significantly degraded.

This result is reinforced by the favorable comparison with the cMSE based policy as well. The proposed algorithm performs better than cMSE method because it assigns priority to the packets based on the saliency based weighted SSIM scores and temporal fluidity break measure scores which are known to closely correlate with the subjective scores. cMSE, on the other hand, assigns priority

Table 3.5: PSNR Scores for 1 % PLR

Clip	Cumulative MSE	SSIM
Carphone	31.9795	32.3649
Flower	28.9677	33.9549
Football	35.1218	35.1956

to the packets based on MSE averaged over all the frames affected by packet loss due to error propagation. It is well known that MSE does not correlate well with subjective scores [42].

We would also like to note that several of the interesting observations and pitfalls noted by Chang et. al. [2] are corroborated/addressed in this work. It was noted that the distance between lost frames in the case of dual loss plays an important role in the visibility of the error. In our proposed algorithm, we ensure that no more than five consecutive frames are assigned zero priority. It was noted that error concealment plays an important role in deciding frame loss visibility. Stage-2 of our algorithm takes this into account since perceptual importance is estimated after decoding. It was further observed in [2] that of all the factors considered, motion related factors are the most important ones in priority assignment. Our use of the TFBM in Stage-1 is in line with this observation.

In addition to better performance, the proposed algorithm requires no prior training with subjective scores thereby making it easier to implement and deploy. We have replaced the requirement for subjective evaluation by using objective perceptual quality metrics instead that correlate well with subjective scores. The combination of spatial and temporal features ensures good performance across a range of motion content. Also, the proposed algorithm makes minimal use of empirically determined parameters thereby making it applicable in a wide range of applications. Further, the performance of the proposed algorithm highlights the fact that perceptually motivated packet prioritisation is a promising approach to estimating the perceptual effects of packet loss.

3.5 Areas of Application

The proposed algorithm can be used for packet priority assignment at the server where pre-encoded video is stored. The video server cannot interfere with the encoding scheme of already encoded and stored videos in the server. Hence Scalable video coding which is an improved encoding technique and Joint Source Channel Coding are not feasible in this scenario.So the proposed method can be used as it does not interfere encoding process and only relies on encoded bitstream for priority assignment.

If the packets of different layers of the scalable encoded video are assigned priority using our algorithm then instead of dropping entire enhancement layer only low priority enhancement layer packets can be dropped under low bitrate channel conditions. Thus our method provides more scalability when used in conjunction with scalable video coding.

Many cross layer techniques are lacking in an efficient priority assignment technique in the application layer and use cMSE which in our paper is proven to be less efficient compared to our technique. So our technique can be used for packet priority assignment in application layer in the existing cross layer techniques to improve their performance.

3.6 Extension

The proposed algorithm is flexible and can be modified to create more priority levels than just two. We used binary just for simplicity and just as a simple case of illustration of our algorithm. But it is obvious from the algorithm implementation that it lends itself well for multi-class classification of packets as Saliency weighted SSIM and TFBM values are continuous values which can be quantized to multiple classes.

Chapter 4

Future Work

4.1 Alternative Priority Assignment Technique

GLM approximates the linear combination of features to be proportional to packet loss visibility. But the features need not be linearly related to packet loss visibility and the features are inter-dependent. To culminate these problems, a new approach can be used as follows

- Construct a column vector with features as its elements for each video packet.

- Construct an training matrix of size MxN with feature vectors as columns where M is the size of each feature vector and N is the number of training samples.

- Apply Principal Component Analysis(PCA) to de-correlate the features.

- Use a suitable supervised learning algorithm to classify the de-correlated feature vectors into required number of priority groups.

The labels for the groups in supervised learning are obtained by subjective evaluation of the loss-induced video as mentioned in Section 2.9.1.

4.2 Two Queues Methodology

In most of the cross layer optimization techniques the methodology is as follows

- Divide the Slices into Priority Groups

- Apply an existing priority agnostic optimization technique to this slice groups separately in the order of their priorities in every time slot.

In case of binary priority assignment, The cross layer problem is to decide which queue should be serviced first in a given time slot where there are two priority queues. A simple solution to this problem is to serve all the high priority packets first and then the low priority packets in a given time slot.An improved methodolgy which is more efficient is proposed to decide which queue is to be serviced at a given time as follows

- A revenue Function R is calculated for each queue in a given time slot and the queue with largest revenue during that time is serviced first.
- The Revenue Function R is given by $R_i = A_{1i}Q_i + A_{2i}\phi + A_{3i}P_i - A_{4i}D_i$
  where
  R_i is the instantaneous revenue function of the queue i
  Q_i is the buffer occupancy of queue i
  ϕ is the representative of the channel conditions in that time slot given SNR
  P_i is the importance of the head-of-the-line packet in the queue i in that instant
  D_i is the deadline of the head-of-the-line packet in the queue i in that instant
  $A_{1i}, A_{2i}, A_{3i}, A_{4i}$ are scheduler weights which are decided based on user's feedback and QoE.

The following four key factors determine the QoE of a video client and hence decide scheduler weights:
(a) average quality,
(b) temporal variability in quality,
(c) fraction of time spent rebuffering, and

(d) cost to the video client and video content provider.

Client preferences regarding the Rebuffering and cost are taken into account in this scenario.For instance, a video client may be willing to tolerate rebuffering in return for higher mean quality (for e.g., to watch a movie in HD over a poor network) and hemay want to tradeoff QoE versus delivery cost.

In a multi user environment, the individual optimization strategies should take into account the effect on other users and the optimization should be foresighted i.e., optimizing the short term video quality without taking into account the effect of the current decision on the long term quality is not a good methodology.

Chapter 5

Conclusion

5.1 Conclusion

We presented a novel two-stage algorithm for assigning priority to video packets. The first stage estimated the impact of packet loss on temporal quality while the second stage estimated the effect of packet loss on spatial quality. These estimates were made using perceptually motivated features. The spatial and temporal importance of packets was non-linearly combined to assign packet priority. The efficacy of the proposed method relative to the cMSE-based prioritization method was demonstrated using an intelligent packet drop application. The two-stage algorithm lends itself to application in different practical settings such as at a router or at a video server. Also, the proposed algorithm was tested using an I-B-P GOP but it works equally well for other GOP structures due to its GOP-structure independence. Further, the algorithm can be easily extended to handle multiple packet losses since TFBM accounts for temporal impairments.

Since layered optimization is sub-optimal and does not fulfill the goal of maximizing the QoE of the end user, cross-layer optimization is employed in multimedia traffic management. Eventhough cross-layer optimization defies the strict boundaries defined between layers at their interface, the number of parameters exchanged between layers is kept minimal.Content and network features that can easily be computed and are good indicators of which composite (integrated) strategy is optimal(i.e., provides best possible QoE) are identified and used as optimization parameters.

An alternate strategy to assign priority to video packets is proposed which can be implemented if an appropriate supervised learning algorithm is identified. This method is expected to work better than other objective packet classifying algorithms like GLM, CART as it takes the inter-dependence between features into account and eliminates the dependence.A revenue function is calculated to decide the order in which packets are to be serviced taking into account the packet importance, their respective queue length, network conditions and packet display deadline.

References

[1] C. Inc. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016 2012.

[2] Y.-L. Chang, T.-L. Lin, and P. C. Cosman. Network-Based H. 264/AVC Whole-Frame Loss Visibility Model and Frame Dropping Methods. *Image Processing, IEEE Transactions on* 21, (2012) 3353–3363.

[3] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. C. Cosman, and A. R. Reibman. A versatile model for packet loss visibility and its application to packet prioritization. *Image Processing, IEEE Transactions on* 19, (2010) 722–735.

[4] S. Kanumuri, P. C. Cosman, A. R. Reibman, and V. A. Vaishampayan. Modeling packet-loss visibility in MPEG-2 video. *Multimedia, IEEE Transactions on* 8, (2006) 341–355.

[5] S. Kanumuri, S. G. Subramanian, P. C. Cosman, and A. R. Reibman. Predicting H. 264 packet loss visibility using a generalized linear model. In Image Processing, 2006 IEEE International Conference on. IEEE, 2006 2245–2248.

[6] J. Ascenso, H. Cruz, and P. Dias. Packet-header based no-reference quality metrics for H.264/AVC video transmission. In Telecommunications and Multimedia (TEMU), 2012 International Conference on. 2012 174 –151.

[7] A. R. Reibman, V. A. Vaishampayan, and Y. Sermadevi. Quality monitoring of video over a packet network. *Multimedia, IEEE Transactions on* 6, (2004) 327–334.

[8] X. Gao, N. Liu, W. Lu, D. Tao, and X. Li. Spatio-temporal salience based video quality assessment. In Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on. IEEE, 2010 1501–1505.

[9] X. Feng, T. Liu, D. Yang, and Y. Wang. Saliency based objective quality assessment of decoded video affected by packet losses. In Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on. IEEE, 2008 2560–2563.

[10] A. R. Reibman and D. Poole. Characterizing packet-loss impairments in compressed video. In Image Processing, 2007. ICIP 2007. IEEE International Conference on, volume 5. IEEE, 2007 V–77.

[11] A. R. Reibman and D. Poole. Predicting packet-loss visibility using scene characteristics. In Packet Video 2007. IEEE, 2007 308–317.

[12] M. Schier and M. Welzl. Optimizing selective ARQ for H. 264 live streaming: A novel method for predicting loss-impact in real time. *Multimedia, IEEE Transactions on* 14, (2012) 415–430.

[13] M. Schier and M. Welzl. Content-aware selective reliability for DCCP video streaming. In Multimedia Computing and Information Technology (MCIT), 2010 International Conference on. IEEE, 2010 53–56.

[14] G. Sun, W. Xing, and D. Lu. A content-aware packets priority ordering and marking scheme for H. 264 video over diffserv network. In Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on. IEEE, 2008 1735–1738.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on* 13, (2004) 600–612.

[16] A. K. Moorthy and A. C. Bovik. Visual importance pooling for image quality assessment. *Selected Topics in Signal Processing, IEEE Journal of* 3, (2009) 193–201.

[17] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20, (1998) 1254–1259.

[18] Q. Ma and L. Zhang. Saliency-based image quality assessment criterion. In Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, 1124–1133. Springer, 2008.

[19] M. van Der Schaar et al. Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms. *Wireless Communications, IEEE* 12, (2005) 50–58.

[20] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on, volume 2. IEEE, 2003 1398–1402.

[21] S. Thakolsri, W. Kellerer, and E. Steinbach. QoE-based cross-layer optimization of wireless video with unperceivable temporal video quality fluctuation. In Communications (ICC), 2011 IEEE International Conference on. IEEE, 2011 1–6.

[22] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Signal processing: Image communication* 19, (2004) 121–132.

[23] K. Piamrat, A. Ksentini, C. Viho, and J.-M. Bonnin. Qoe-aware admission control for multimedia applications in ieee 802.11 wireless networks. In Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th. IEEE, 2008 1–5.

[24] K. Piamrat, K. D. Singh, A. Ksentini, C. Viho, and J.-M. Bonnin. QoE-aware scheduling for video-streaming in High Speed Downlink Packet Access. In Wireless Communications and Networking Conference (WCNC), 2010 IEEE. IEEE, 2010 1–6.

[25] K. K. Kambhatla, S. Kumar, S. Paluri, and P. C. Cosman. Wireless H. 264 Video Quality Enhancement Through Optimal Prioritized Packet Fragmentation. *Multimedia, IEEE Transactions on* 14, (2012) 1480–1495.

[26] H. Ha, J. Park, S. Lee, and A. C. Bovik. Perceptually unequal packet loss protection by weighting saliency and error propagation. *Circuits and Systems for Video Technology, IEEE Transactions on* 20, (2010) 1187–1199.

[27] M. Van Der Schaar et al. Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms. *Wireless Communications, IEEE* 12, (2005) 50–58.

[28] S. Singh, J. G. Andrews, and G. de Veciana. Interference shaping for improved quality of experience for real-time video streaming. *Selected Areas in Communications, IEEE Journal on* 30, (2012) 1259–1269.

[29] K. K. Kambhatla, S. Kumar, and P. Cosman. Prioritized packet fragmentation for H. 264 video. In Image Processing (ICIP), 2011 18th IEEE International Conference on. IEEE, 2011 3233–3236.

[30] S. H. Kang and A. Zakhor. Packet scheduling algorithm for wireless video streaming. In International Packet Video Workshop, volume 2002. 2002 .

[31] P. Bucciol, G. Davini, E. Masala, E. Filippi, and J. C. De Martin. Cross-layer perceptual ARQ for H. 264 video streaming over 802.11 wireless networks. In Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE, volume 5. IEEE, 2004 3027–3031.

[32] T. Liu, X. Feng, A. Reibman, and Y. Wang. Saliency inspired modeling of packet-loss visibility in decoded videos. In International Workshop VPQM. 2009 1–4.

[33] A. R. Reibman, S. Kanumuri, V. Vaishampayan, and P. C. Cosman. Visibility of individual packet losses in MPEG-2 video. In Image Processing, 2004. ICIP'04. 2004 International Conference on, volume 1. IEEE, 2004 171–174.

[34] K.-C. Yang, C. Guest, K. El-Maleh, and P. Das. Perceptual Temporal Quality Metric for Compressed Video. *Multimedia, IEEE Transactions on* 9, (2007) 1528 –1535.

[35] Z. Li, J. Chakareski, X. Niu, Y. Zhang, and W. Gu. Modeling and analysis of distortion caused by Markov-model burst packet losses in video transmission. *Circuits and Systems for Video Technology, IEEE Transactions on* 19, (2009) 917–931.

[36] B. A. Wandell. Foundations of vision. Sinauer Associates, 1995.

[37] K. Seshadrinathan and A. Bovik. Motion Tuned Spatio-Temporal Quality Assessment of Natural Videos. *Image Processing, IEEE Transactions on* 19, (2010) 335 –350.

[38] VQEG STL: Tools and Subjective Labs Setup, http://vqegstl.ugent.be/?q=taxonomy/term/2/.

[39] The Network Simulator – NS2, http://www.isi.edu/nsnam/ns/.

[40] M. H. Pinson and S. Wolf. A new standardized method for objectively measuring video quality. *Broadcasting, IEEE Transactions on* 50, (2004) 312–322.

[41] VQM Software: http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm.

[42] Z. Wang and A. C. Bovik. Mean squared error: love it or leave it? A new look at signal fidelity measures. *Signal Processing Magazine, IEEE* 26, (2009) 98–117.