QoP-Driven Scheduling for MPEG Video Decoding

Shaoxiong Hua and Gang Qu, Member, IEEE

Abstract — *MPEG video decoding algorithm has been embedded into many consumer electronic products. In this paper, we demonstrate that the completion ratio (CR) does not represent well the quality of presentation (QoP) of MPEG video. We then propose a new QoP metric, which 1) is based on frame completion ratio but 2) differentiates firm and soft deadlines and also 3) considers the frame dependency. We show that, on a set of simulated MPEG movies, the proposed QoP metric measures the QoP of the movies much better than the completion ratio. We then present a set of online scheduling algorithms that enhance QoP significantly, particularly for overloaded systems.*

Index Terms — MPEG video decoding, overloaded systems, QoP, real-time scheduling.

I. Introduction

With MPEG becoming the video compression standard, MPEG decoding algorithm has been embedded into many consumer electronic products other than DVD and video game players. For example, cell phones and PDAs (Personal Digital Assistants) are now designed to enable some multimedia applications to increase the market competitiveness; videophones do not exist only in scientific fictions.

MPEG was originally designed for efficiently storing video and audio on digital media; however, it is also suitable for transmitting video frames over computer networks in order to lower its demand on network bandwidth. The MPEG coding standard defines a lossy[1] compression technique that takes advantage of spatial and temporal correlation to achieve high compression ratios [1]. Specifically, MPEG video coding uses transform coding such as discrete cosine transform (DCT) to organize the redundancy in the spatial direction and motion compensation to predict motion from frame to frame in the temporal direction. A MPEG video sequence is made up of group of pictures (GOP). Each GOP is composed of an I frame and some P and B frames. I frame is *intra* coded frame which uses only spatial correlation and contains no references to other pictures of the stream. P frame is predictive coded frame that is coded more efficiently by also using temporal redundancies from a preceding I or P frame. B frame is bi-directional-predictive coded frame, which uses temporal redundancies from both preceding and succeeding frames.

When a system receives the MPEG video stream, before the video can be displayed, it has to decode (uncompress) them, which is often computationally intensive. As more and more applications are enabled, such system will be overloaded more frequently. For instance, when a DVD movie is being played, an incoming (video) phone call or a message with multimedia attachment may greatly increase system's processing load. In such occasion, the system may not be able to decode all the frames correctly in a timely fashion. Dropped or incorrectly decoded frames degrade the video's quality of presentation (QoP), which can be conveniently measured by the number of correctly decoded frames. This poses a unique challenge for video decoding: how to maximize QoP when the system is overloaded?

For the overloaded real-time systems, task completion [2], deadline miss-ratio [3] and loss-rate [4] have been widely used as the measurement of quality of service (QoS), which will be eventually mapped to QoP. Baruah et al. [2] study how to maximize task completion for overloaded systems. They conclude that any online algorithm may perform arbitrarily poorly as compared to a clairvoyant scheduler, but discuss competitive online schedulers for a few special cases. For the MPEG players system, Ng et al. [5] propose a QoS metric QoS-Human, which measures the human judgment of quality of video display against the number of frames displayed for different kinds of video. Comprehensive survey of QoS research in network and real-time system areas is given in [6].

Although MPEG video streaming can be considered as a soft real-time application in which occasional deadline misses can be tolerated, the user required guaranteed QoS or QoP leads the system to conduct the real-time scheduling in order to efficiently use the available computing resources. Because execution time analysis is the foundation for using real-time scheduling, there are some works on predicting the execution time of MPEG video decoding. In [7] Bavier et al. introduce a method for predicting execution times of MPEG-decoding by using information from previously decoded frames and the size and type of MPEG-encoded frames. In [8] Burchard et al. provide exact information about the execution time in the voice-on-demand scenario and two-phase approach that significantly reduces the gap between the worst-case and the actual execution times in live-video scenario.

In this paper, we propose a new QoP metric that reflects user perceived QoP much better than traditional completion ratio, which is a good metric for independent tasks and used by most online schedulers. Then, under the assumption that the actual execution time of MPEG-decoding can be known a priori by predicting, we present a set of on-line scheduling algorithms to maximize the proposed QoP measurement in order to achieve high performance at the user level.

S. Hua is with the Electrical and Computer Engineering Department, University of Maryland, College Park, MD 20742 USA (e-mail: shua@eng.umd.edu).

G. Qu is with the Electrical and Computer Engineering Department and Institute of Advanced Computer Studies, University of Maryland, College Park, MD 20742 USA (e-mail: gangqu@eng.umd.edu).

[1] Lossy compression scheme compresses a file by permanently eliminating certain information, especially redundant information. When the file is uncompressed, only a part of the original information is still there (although the user may not notice it).

Contributed Paper
Manuscript received June 22, 2003

The rest of the paper is organized as following. Section II defines our quantitative measurement for QoP that captures the firm/soft deadlines and frame dependency. Section III presents our online QoP-driven scheduling policies to maximize the defined QoP. In section IV, we apply the proposed online schedulers to a set of simulated MPEG movies and demonstrate that these scheduling algorithms with small run-time overhead are effective in improving not only our QoP measurement but also the actual QoP over classic scheduling policies such as EDF. The paper is concluded in Section V.

II. QoP Model

We consider a system that real-time decodes the arriving MPEG video. The video stream consists of a sequence of frames such as I frames, P frames and B frames, and each frame is characterized by *<a,d,e,f/s>*, where *a* is the arrival time, *d* is the decoding deadline which depends on the frame display rate, *e* is the execution time, and *f/s* specifies whether the deadline is firm or soft.

- A frame has a *firm* deadline if it must be completed before the deadline otherwise the system will not get the reward for serving the frame and the application.
- A frame has a *soft* deadline if the system can still benefit even if the deadline is missed, subjected to a deadline-miss penalty.
- A frame is *non-preemptive* means that once the frame gets the CPU, it will occupy the CPU until its deadline or completion, whichever comes earlier.
- A frame is *preemptive* means that the frame may lose control of the CPU during its execution, but when it gets the CPU back, it will restore the previous state and resume the interrupted execution.

Although in the MPEG video decoding, in fact all of the deadlines of the frames are soft regardless of the frame type, we intentionally set the deadline of the important frame such as I and P frame to be firm while setting the deadline of B frame to be soft. The reason is that because the system memory is finite, for the overloaded system that does not have enough computing capability to decode all frames, some frames have to be eventually dropped. In order to encourage the completion of the important frame, we assign them to be firm which is consistent with the common sense that firm-deadline task is more important than soft-deadline task.

An online scheduler will allocate the system resource to decode the frame it selects. The completion ratio is defined as the ratio of completed frames over the total number of frames according to the given scheduler. Although it has been widely used in real-time systems, completion ratio does not give an accurate measure for the QoP of MPEG video decoding due to the following reasons: 1) it does not distinguish frames which are completed before their soft deadlines and those that are completed but miss their soft deadlines; and 2) it does not reflect data dependency among frames because all deadline misses are treated in the same way. Based on these observations, we define our new QoP metric for MPEG video decoding as follows:

Suppose that a scheduler S completes K_f firm-deadline frames and K_s soft-deadline frames out of a total of N frames, the QoP provided by such scheduler is:

$$Q_{MPEG}(S) = \frac{K_s + K_f}{N} - \frac{\beta}{N}\sum_{i=1}^{K_s}\frac{\delta_i}{d_i - a_i} - \frac{\gamma}{N}\left(m_I\Delta_I + \sum_{i=1}^{n_P} m_{P,i}\Delta_{P,i}\right) \quad (1)$$

Where β is the penalty parameter or the tolerance factor for deadline missing; γ is the penalty parameter for frame dropping; δ_i is the difference between the frame's deadline and completion time when the soft deadline is missed (if the frame is completed before its deadline or eventually dropped, then δ_i is 0); d_i-a_i is the life time of the frame and is equal to the reciprocal of the frame display rate; Δ_I and $\Delta_{P,i}$ are the number of frames that will be affected if the I frame or P frame is dropped and they can be determined by the characteristic of the GOP pattern; m_I or $m_{P,i}$ is the number of dropped I or P frames; n_P is the number of P frames in a GOP pattern. In (1) the first term rewards frame completion; in the other terms, the first sum is taken over all the completed frames (B frames) that miss their soft deadlines; and the second sum is taken over all the dropped frames (I/P frames).

The QoP defined in (1) is a direct extension of completion ratio, in the case when there is no penalty for missing soft deadlines ($\beta = 0$) or dropping frames ($\gamma = 0$). Soft deadline missing is penalized by the relative amount that the deadline has been missed with the penalty factor β. Data dependency is captured in the last term by reducing QoP in the amount of frames depending on dropped frames with a penalty factor γ. From (1), we can see that although the completions of firm-deadline frame and soft-deadline frame have equivalent reward, i.e., 1/N, their deadline misses lead to different results. For the firm deadline frame, the system will get no reward and even negative reward because of its deadline missing. However, for the soft-deadline frame, the system may still have positive reward. Therefore we can conclude that in order to maximize $Q_{MPEG}(s)$ as defined in (1), the system prefers to decode firm-deadline frame than decode the soft-deadline one. This exactly matches the fact that the firm deadline frame is more important.

Note that this QoP measurement is calculated incrementally at run time and there are only a few arithmetic operations involved at each frame. The penalty parameter β and γ are *stream specific.* For example, the β and γ for decoding Cartoon Video (e.g., 0.8) should be smaller than those for decoding Action Video (e.g., 1.0) because the human being are less sensitive to the artificial movements in Cartoon Video, but are very sensitive to the smoothness of the motions in Action Video [5]. The values of these parameters can be stored as *user-defined data* within the stream. User-defined data is defined by the MPEG standard in order to place additional data, which could provide application specific information to aid software operation, within the stream.

In this paper, based on the assumption that the decoding time of each frame can be known a priori by predicting it based on the frame type, size and workload history etc. [7], [8], we study the following QoP-driven online scheduling problem: *for a set of real-time MPEG video frames with mixed firm and soft deadlines on a single processor system, determine an online schedulers S to maximize* $Q_{MPEG}(S)$.

III. Online Schedulers

Due to the uncertainty of the arriving frames and the nature of online scheduling, it becomes unavoidable to drop frames and hard to provide absolute QoP guarantees. Our objective is thus to develop online scheduling algorithms that give competitive average QoP. An online scheduling policy must have low complexity because it will be executed frequently on the fly. It should also specify its drop policy as the frame drop becomes inevitable. In this section, we first give the drop lemma and then explain a set of online scheduling heuristics based on the widely used EDF (Earliest Deadline First) and LETF (Least Execution Time First).

Lemma (Drop Lemma):

If a scheduler (online or offline) maximizes the QoP as defined in (1), then it must

1) drop frame $<a,d,e,f>$ at time $t > d - e^*$
2) drop frame $<a,d,e,s>$ at time $t > \frac{1+\gamma\times\Delta}{\beta}(d-a)+(d-e^*)$

where e^* is the frame's remaining execution time, and $e^* = e$ for non-preemptive frames, Δ is the number of frames that will be affected by the frame dropping.

[Proof] At time t, the earliest time that we can complete frame $<a,d,e,f/s>$ is $t+e^*$, where e^* is the frame's remaining execution time. If the frame has a firm deadline d, it cannot be completed and will not contribute for QoP at time t when $t+e^* > d$. If the frame has a soft deadline, we will execute it if and only if the benefit of completion (with deadline missing penalty if applicable) exceeds the penalty for dropping the frame, that is, $\frac{1}{N} - \frac{\beta}{N}\frac{\delta}{d-a} > -\frac{\gamma}{N}\Delta$ where $\delta = t + e^* - d$. A simple calculation leads us to 1) and 2) as above.

Intuitively, Drop Lemma suggests us to drop firm-deadline frames as soon as we discover that we are unable to finish on time. However, for soft-deadline frames, Drop Lemma implies that we should wait an extra period because soft deadline miss will still be beneficial to some extent. Clearly, the smaller is the deadline missing penalty parameter β and the larger is the weight of drop penalty γ, the longer we should wait.

A. S2F : Soft to firm deadline conversion

From Drop Lemma, we see that two frames $<a,d,e,s>$ and $< a, d + \frac{1+\gamma\times\Delta}{\beta}(d-a), e, f >$ will always be dropped at the same time. Notice that the first frame has a soft deadline and the second one has a firm deadline. Therefore, we propose the first QoP-driven online scheduling algorithm based on this soft deadline to firm deadline conversion:

Algorithm S2F:

1) *For each soft deadline frame* $<a,d,e,s>$
2) change its deadline from d to $d + \frac{1+\gamma\times\Delta}{\beta}(d-a)$;
3) change its deadline type from soft to firm;
4) apply EDF on the new set of firm real-time frames;

B. EDF* and LETF*

The EDF and LETF service strategies are among the most popular ones for real-time applications. On the completion of one frame, they aggressively schedule the next frame with the earliest deadline and the least execution time respectively. However, neither of them distinguishes firm deadlines and soft deadlines and they may decide to execute frame that should be dropped according to the Drop Lemma. We integrate the Drop Lemma into these two scheduling policies and propose scheduling algorithms EDF* and LETF*.

Algorithm EDF or LETF*:*

1) *On the completion of the current frame* τ *(or on the arrival of a new frame if preemption is allowed)*
2) *if preemption is assumed*
3) *replace the execution time of frame* τ *by its remaining execution time;*
4) *drop all the frames that meet the condition in Drop Lemma;*
5) *schedule the remaining frames using EDF or LETF.*

Non-preemptive execution stops only at the completion of the current frame. We are guaranteed that this completion will either meet the frame's deadline or still gives positive contribution to the QoP even its soft deadline is missed. The reason is that the current frame is the winner of all the frames in the previous round, which means that it survives the drop policies. During the drop policy checking in step 4, unlike the original schedulers, EDF* and LETF* will treat firm and soft deadline frames differently to maximize QoP. Finally, we argue that the drop policy checking takes only constant time. For example, in the implementation, we can first choose the frame picked by EDF or LETF and check whether it meets the drop policies. If the Drop Lemma is satisfied, we drop the frame and ask EDF or LETF for their next choice. Therefore, EDF* or LETF* will have the same run-time *complexity* as the original one.

C. IFF: Important frame first

Based on the MPEG standard, we know that I frame is the most important and P frame is more important than B frame. From (1), we see that dropping the important frame such as I/P frame will affect the frames that depend on them. Furthermore, we can see that missing firm deadline immediately erases the efforts that we have already put on the frame completely. However, when we miss the soft deadline, we still get the chance to improve the QoP by finishing the frame in a reasonable amount of extra time. Thus, in order to maximize QoP, we should assign frames with firm deadline higher priority than those with soft deadlines. On the other hand, as we mentioned before we assign firm deadline to important frame such as I or P frame and assign soft deadline to B frame. Based on these observations, we propose the important frame first online scheduling algorithm, which is a variation of EDF, to maximize the QoP.

Algorithm IFF:

1) *On the completion of the current frame τ (or on the arrival of a new frame if preemption is allowed)*
2) *if preemption is assumed,*
3) *replace the execution time of frame τ by its remaining execution time;*
4) *drop all the frames that meet the condition in Drop Lemma;*
5) *select the frame τ' with the earliest deadline in the ready list;*
6) *if τ' is not the most important frame, i.e., I frame,*
7) *check the drop policy at time* *t = current time* + execution time of frame τ';
8) *if any frame that is more important than frame τ' will be dropped*
9) *unselect τ' and goto step 5;*
10) *decode the selected frame.*

IFF is similar to EDF* with special treatment to important frames. In particular, if the frame is not the most important frame, i.e., I frame, we check whether there will be any more important frame dropping because we execute this frame first. In another word, a frame will be processed only if its execution will not cause any other more important frame drops. The *complexity* of IFF is approximately the same as EDF*.

IV. Experimental Results

We have implemented the proposed QoP-driven online schedulers and applied them to a set of simulated MPEG movies [9]. In this section, we report the setup of our simulation on MPEG movies and the comparison of the completion ratio (CR), our proposed new QoP metric, and user perceived QoP, which can be conveniently measured by the number of *correctly* decoded frames by using different online schedulers. Our results indicate that while these schedulers have similar performance for completion ratio, they behave very differently under the new QoP metric. Compared with EDF, most proposed algorithms not only improve our new QoP metric but also improve real QoP dramatically especially IFF, which gives the most improvement.

A. Simulation of MPEG streams

We have tested the proposed algorithms on MPEG video streams decoding at the frame level. Standard MPEG encoders generate three types of compressed frames: I frames (intra-pictures), P frames (predicted pictures) and B frames (bi-directional predicted pictures). In general, encoders use a fixed GOP (Group of Pictures) pattern when compressing a video sequence. A typical GOP in display order and decoding order is shown as in Fig. 1.

0	1	2	3	4	5	6	7	8	9	10	11	12	
I_0	P_1	B_2	B_3	P_4	B_5	B_6	P_7	B_8	B_9	I_{10}	B_{11}	B_{12}	decoding order
I_0	B_2	B_3	P_1	B_5	B_6	P_4	B_8	B_9	P_7	B_{11}	B_{12}	I_{10}	display order

Fig. 1. A typical GOP pattern (I-to-I=12, I-to-P=3)

On average, I frames are the largest in size (since they are self-contained), followed by P frames and B frames. Krunz and Tripathi present a comprehensive model for MPEG video streams [9]. This model captures the bit-rate variations at multiple time scales. Long-term variations are captured by incorporating scene changes, which are noticeable in the fluctuations of I frames. Three models are introduced to simulate the frame sizes of different types of frames, and the complete model is finally obtained by intermixing these three sub-models according to a given GOP pattern. Statistically, the generated MPEG streams fit the empirical video and are sufficiently accurate in predicting the queuing performance for real video streams. We simulate the frame information for movies, *Wizard of OZ, Star Wars, Silence of the Lambs, and Goldfinger,* from the parameters reported in [9].

Based on the frame size and type, we generate the normalized execution time for each frame using a linear model of MPEG decoding [7]. In the simulation we assume that the frames arrive in the decoding order and their inter-arrival time are independent with exponential distribution. The mean of the exponential distribution is approximately equal to the reciprocal of frame display rate (in terms of fps or frame per second) to generate a balanced loaded system. We simulate underloaded and overloaded systems by varying the fps requirement. The absolute deadline of each frame is monotonically increasing in its arrival time. We use several standard display rates (in terms of fps) in our simulation: 15, 30 (standard for computer video and graphics), 45 and 60 (suitable to sports and other fast-action programs). As we mentioned before the deadline type is assigned to each individual frame based on the dependency of different frames. I frame is the most important because the correct processing of all the P frames and B frames in the same GOP depends on the completion of the corresponding I frame. P frame is also important because it is required by the following P and B frames in the same GOP. We assign I and P frames firm deadlines rather than giving them soft deadlines. We also assign soft deadlines to B frames to create frames with mixed type of deadlines.

Each GOP can be viewed as one "application" independent of others as the correct decoding of all the frames in one GOP depends on the leading I frame. Each "application" consists of a set of tasks (frame decoding) and the drop of firm deadline I and P frames will cause the incorrect decoding of the remaining frames in this "application". To better model the data dependency among "tasks", we assign different values Δ_I and $\Delta_{P,i}$, which are corresponding to the number of frames that will not be decoded correctly because of a dropped frame, to frames with firm deadlines. For example if I-to-I, the number of frames between two consecutive I frames (see Fig. 1), is 12, then we assign Δ_I =11; $\Delta_{P,i}$ are assigned 10, 7, and 4 for the three P frames in the GOP pattern based on Fig. 1; and $\Delta_B = 0$ because there is no frame depends on the B frame. As a result, I frames have higher priority than P and B frames; P frames have higher priority than B frames. This exactly matches the MPEG decoding mechanism. In the simulation, the penalty parameter β and γ are both set to be the default value 1.

B. Simulation Results

There are two goals in our simulation: demonstrating that our new QoP metric reflects user perceived QoP better than the completion ratio (CR), and showing that our proposed online scheduling algorithms can improve dramatically our defined QoP metric and real QoP for the users.

First we have implemented popular online scheduling algorithms such as EDF and LETF and applied them to the

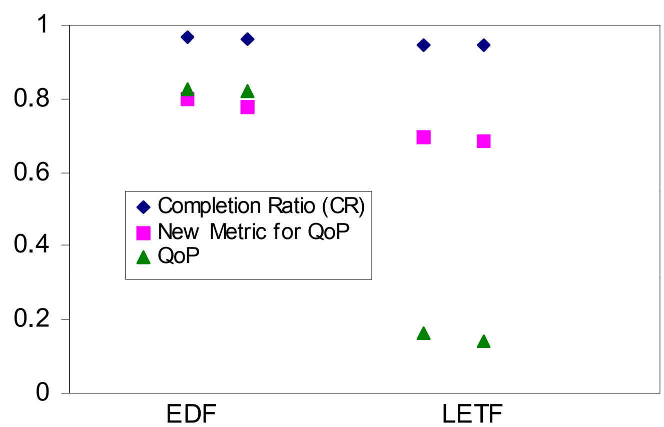


Fig. 2. Comparison of some widely used online schedulers on movie "Goldfinger" in the frame rate of 30 fps in the case of, from left to right, non-preemptive and preemptive.

simulated MPEG movies. For each movie, we simulate underloaded, balanced, and overloaded systems by changing the frame rate from 15 fps, to, 30, 45, and 60 fps. And for each case, we consider the case of non-preemptive and preemptive. Fig. 2 is the typical relationship of completion ratio, our proposed new QoP metric and user perceived QoP, which considers the actual number of correctly decoded frames, under different online scheduling policies (EDF and LETF) on movie "Goldfinger" in the frame rate of 30fps. The other popular scheduling algorithm FCFS (first come first serve) in our simulation is actually same as EDF because the system has monotonic absolute deadlines. From Fig. 2 we can see that our new QoP metric is much closer to the real QoP. So it is necessary to develop low overhead online scheduler to maximize this new QoP metric in order to eventually improve user perceived QoP without using extra hardware.

Then we implemented the proposed QoP-driven online scheduling algorithms and applied them to the simulated MPEG movies under different frame rates and different preemptive types. For underloaded system with a frame rate of 15 fps, the deadlines are relatively loose and we observe that almost all the algorithms achieve the maximal QoP in the amount of 1 without the frame drop and deadline missing. However, when the computation load increases, the system becomes balanced and overloaded eventually. Then we see, for instance in the movie of "Goldfinger" as shown in Fig. 3, different online schedulers provide very different QoP which have the same trends as the new defined QoP metrics. In general we can rank them in the increasing order of QoP: LETF*, EDF, EDF*, S2F, and IFF. When the system goes to overload state (such as 45 fps and 60 fps), the algorithm IFF achieves significant higher QoP compared to other algorithms.

It is of our particular interest to study overloaded systems

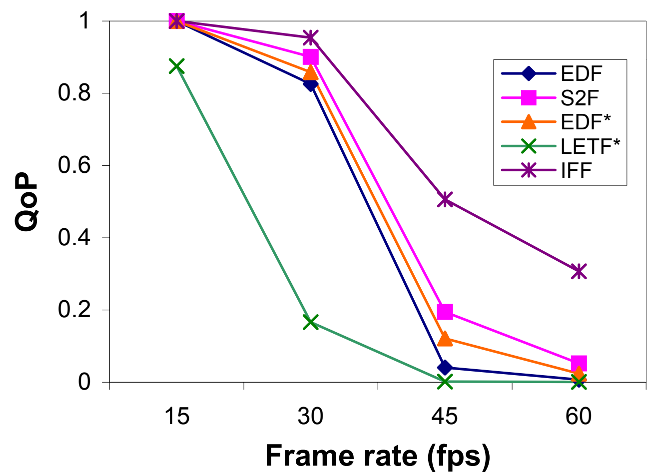


Fig. 3. Comparison of QoP under different online schedulers on movie "Goldfinger" in the case of non-preemptive with different frame rates (15, 30, 45, 60 fps).

where frame drop and deadline missing become unavoidable. Fig. 4-7 give the detailed report on the new QoP metric as defined in (1), completion ratio and real QoP, achieved by different schedulers at certain frame rate. We mention that the negative QoP measurement comes from the fact of frame drop and deadline missing as well as their associated penalties. It is possible to give a more accurate modified measure of QoP in this case to keep QoP positive. For example, in the fast-forward mode, the frame drop penalty should be much less, as is the soft deadline missing penalty. From these figures we can see that almost all the schedulers achieve similar completion ratio, however, their new QoP metric and real QoP are quite different. The conclusion is that regardless of the preemptive type, it is crucial to finish important frames as many as possible, not the raw counter of frame completions. It is mentioned that although LETF algorithm is 1/2–competitive in the completion ratio on our monotonic-absolute-deadline task system [2], LETF*, which is better than LETF in QoP, achieves very bad performance because, in general, the execution time of B frame is shorter than that of I or P frame, therefore, it will prefer to select B frame which actually is the least important frame.

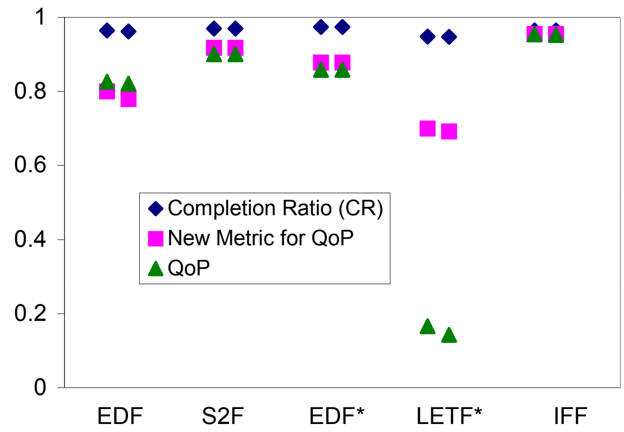


Fig. 4. Comparison of different online scheduling policies on movie "Goldfinger" in the frame rate of 30fps in the case of, from left to right, non-preemptive and preemptive.

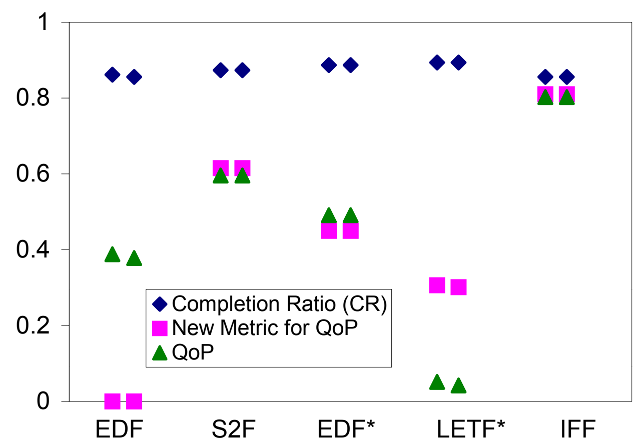


Fig. 5. Comparison of different online scheduling policies on movie "Wizard of OZ" in the frame rate of 30fps in the case of, from left to right, non-preemptive and preemptive.

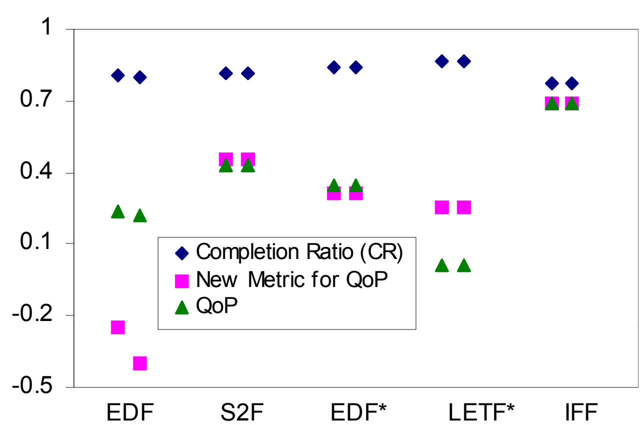


Fig. 6. Comparison of different online scheduling policies on movie "Silence of Lambs" in the frame rate of 45fps in the case of, from left to right, non-preemptive and preemptive.

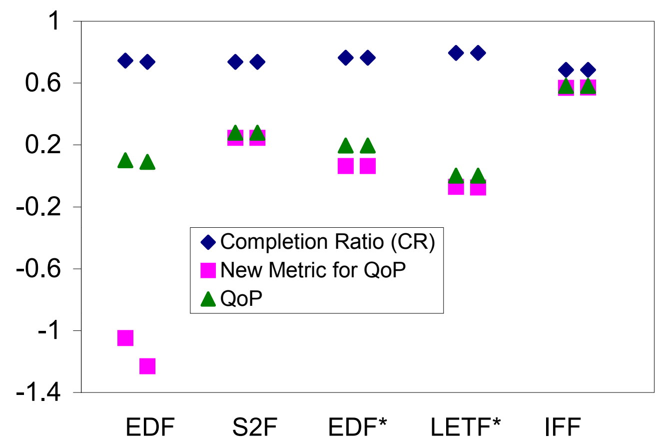


Fig. 7. Comparison different online scheduling policies on movie "Star Wars" in the frame rate of 45fps in the case of, from left to right, non-preemptive and preemptive.

V. CONCLUSION

In this paper we consider the problem of how to improve the user perceived QoP for MPEG video decoding when the system is overloaded or lacks of computation power. We present a new QoP metric that captures the frame dependency. We show that the new defined QoP metric can reflect real QoP much more precisely than the traditional completion ratio. We then find that the most commonly used online scheduling policies do not achieve good performance for MPEG video decoding when the system is overloaded. Based on the proposed quantitative metric, we have developed a set of new online scheduling algorithms to maximize QoP. Simulations on popular MPEG movies show that most of them provide better QoP, particularly on overloaded systems, with about the same run time complexity and no extra hardware. These scheduling algorithms can be embedded in consumer electronic products to efficiently decode the MPEG video stream.

REFERENCES

[1] L. Teixeira and M. Martins, "Video compression: the MPEG standards", *Proceedings ECMAST'96*, pp. 615-634, May 1996.

[2] S. K. Baruah, J. Haritsa, and N. Sharma, "On-line scheduling to maximize task completions", *IEEE Real-Time Systems Symposium,* pp. 228-236, Dec. 1994.

[3] C. Lu, J. A. Stankovic, T. F. Abdelzaher, G. Tao, S. H. Son and M. Marley, "Performance specifications and metrics for adaptive real-time systems", *IEEE Real-Time Systems Symposium,* pp. 13-23, Nov. 2000.

[4] R. West, K. Schwan and C. Poellabauer, "Scalable scheduling support for loss and delay constrained media streams", *IEEE Real-Time Technology and Applications Symposium,* pp. 24-33, June 1999.

[5] J. K. Ng, K.R. Leung, W.Wong, V. C. Lee and C. K. Hui, "Quality of service for MPEG video in human perspective", *Proceedings of the 8th International Conference on Real-Time Computing Systems and Applications (RTCSA)*, pp. 233-241, March 2002.

[6] C. Aurrecoechea, A. T. Campbell, L. Hauw, "A survey of QoS architectures", *Multimedia Systems*, vol. 6, no. 3, pp. 138-151.

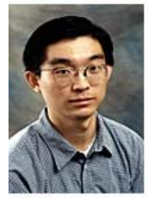
[7] A.C.Bavier, A. B. Montz, and L. L. Peterson, "Predicting MPEG execution times", *ACM SIGMETRICS'98*, pp. 131-140, June 1998.

[8] L. –O. Burchard and P. Altenbernd, "Estimating decoding times of MPEG-2 video streams", Proceedings of the International Conference on Image Processing, pp. 560-563, Sep. 2000.

[9] M. Krunz and S. K. Tripathi, "On the characterization of VBR MPEG streams", *ACM SIGMETRICS'97,* pp. 192-202, 1997.

Shaoxiong Hua received the B.S. and M.S. degrees in Instrument Science from Zhejiang University, Hangzhou, China, in 1992 and 1995, respectively, and the M.S. degree in computer engineering from University of Maryland, College Park, MD, in 2003. Currently, he is a Ph.D. candidate in computer engineering at University of Maryland. From 1995 to 1998, he was an Assistant Professor in the Department of Scientific Instruments, Zhejiang University.

His research interests include embedded/real-time systems, electronic design automation, hardware/software co-design and low power system design.



Gang Qu received the M.S. and Ph.D. degrees in Computer Science from University of California, Los Angeles, CA, in 1998 and 2000. Previously, he had studied Mathematics in University of Science and Technology of China and the University of Oklahoma. Since 2000 he has been an assistant professor in the Department of Electrical and Computer Engineering and Institute for Advanced Computer Studies at University of Maryland, College Park, MD.

His research interests include intellectual property reuse and protection, low power system design, applied cryptography, computer-aided synthesis, and sensor networks.