OPTIMAL SCHEDULING OF MEDIA PACKETS WITH MULTIPLE DISTORTION MEASURES

Carri W. Chan, Nicholas Bambos

Stanford University
Stanford, CA

Susie Wee, John Apostolopoulos

Mobile & Media Systems Lab
Hewlett-Packard Labs, Palo Alto, CA

ABSTRACT

Due to the increase in diversity of wireless devices, streaming media systems must be capable of serving multiple types of users. Scalable coding allows for adaptations without re-encoding. To account for various viewing capabilities of each user, such as different spatial resolutions, multiple distortion measures are used [1]. In this paper, we examine the question of how to broadcast media packets with multiple distortion measures to multiple users. We cast the problem as a stochastic shortest path problem and use Dynamic Programming to find the optimal policy. We generate an offline algorithm to generate the optimal transmission policy for the general case. We then show the optimal policy can be done online via a simple threshold policy for the case of independent Bernoulli packet losses. Through experimental results, we show that our policy, which considers multiple distortion measures, achieves up to 2dB gains over conventional approaches.

1. INTRODUCTION

Media streams are often served simultaneously to many clients with different display capabilities and network connections. Scalable coding methods, such as JPEG2000 [2], allow a streamer to quickly adapt a media stream for different resolutions or bitrates by simply discarding the least important packets. With intelligent prioritization of scalable media packets, the varying needs of each user can be captured. We propose adaptive transmission policies which take a pre-coded media sequence and incorporates the different needs of all users in order to maximize the aggregate quality of service.

To measure the needs of each user, we look at the case of media packets with multiple distortion measures. Multiple distortion measures evolve from the varying capabilities and requirements of the different users. Suppose we transmit an image or video to multiple users with different display capabilities. One user may have a small cellphone display, while another may have a large monitor attached to his desktop computer. Because of the varying display capabilities, media packets will have various distortion values depending on the user who consumes them. A scheduling algorithm optimized for high resolution users may transmit high frequency edge information; however, transmitting the same data to low resolution users may be wasteful if these edges are not visible on a low resolution display. In prior work [1], we developed an algorithm to generate multiple-distortion-measure-aware embedded schedules which incrementally add packets so all packet in a schedule at rate R_1 are also included in a schedule at rate R_2 if $R_1 < R_2$. Embedded schedules make transcoding operations possible by simply truncating the codestream. Gains of up to 4dB were achieved. We also showed that, in fact, an important packet for a high resolution users does not necessarily correspond to an important packet for a low resolution user.

In this paper, we examine a different scenario. We wish to broadcast media streams to multiple users, each with a different distortion measure. A control dilemma arises. If a fraction of users receive a packet, one can either retransmit, benefitting only a fraction of users, or one can drop it and transmit the next packet, which may benefit all users. We propose an optimal offline transmission policy which incorporates multiple distortion values. Then we develop a simple, online policy for a specific case which outperforms the standard approach of using a single distortion measure for all users.

The rest of this paper proceeds as follows. In Section 2, we present our problem formulation and discuss the control dilemma that arises in the broadcast problem. Section 3 introduces a general offline algorithm to determine the optimal transmission policy. In Section 4, we examine the case of Bernoulli packet drops and present an optimal, online transmission algorithm. In Section 5 we compare our multiple-distortion-measure-aware algorithm to standard policies which only consider a single distortion measure. We show that accounting for multiple distortion measures can result in up to 2dB gains.

2. PROBLEM FORMULATION AND SETUP

We consider the following scenario depicted in Fig. 1. An encoded image or video frame is stored at the transmission buffer. In each time slot, the transmitter broadcasts a packet to multiple receivers over the same channel frequency/code/slot; however, the channel quality differs at each receiver due to varying path loss and fading. Therefore, the probability of successful reception varies for each user. At each time slot, the transmitter must determine whether to transmit/retransmit the Head of Line (HOL) packet or drop it in order to service the next packet in the queue. Dropping the HOL packet results in an increase in media distortion of the users that have not received the packet. We assume that distortions are additive across multiple packets.

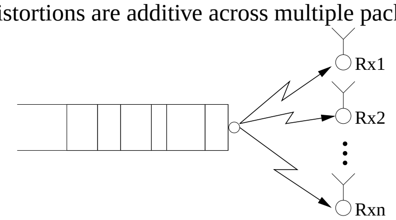

Fig. 1. Broadcasting media packets to multiple users.

The objective is to minimize the total distortion of all receivers. In current media systems, packet scheduling only considers a single distortion measure for each packet–typically mean-squared error compared to the original, high resolution image. However, in some scenarios the amount of distortion incurred by the loss of the same packet varies for each user depending on its viewing capabilities–hence multiple distortion measures. Multiple distortion measures are calculated by evaluating packet importance compared to a benchmark image downsampled to different resolutions. Refer to [1] for more details on multiple distortion measures. We assume that these multiple distortion measures are known to the transmitter via information stored in the packet headers, as described in [3]. If each user incurs identical distortion for the loss of each packet, our problem reduces to the conventional approach and has been well studied. However, if a packet has different distortion values for each user, the decision is more complicated–especially if only a fraction of the users successfully receive the transmitted packet. The following control dilemma arises: should we retransmit the HOL packet so that all users may receive it, or should we drop it after some of them receive it, incur some distortion for those that do not, and transmit the next packet to mitigate delay? Clearly if we require all users to receive every packet, packet transmissions may be blocked by a single weak link.

We assume, after each transmission, acknowledgements are transmitted back to the server over some high bandwidth control channel. Therefore, the server knows which users have and have not yet received the HOL packet. Note that a packet is only useful to a receiver the first time it is received, i.e., distortion is not reduced further with the second reception of a packet. Based on the multiple distortion measures, the HOL packet reception information, and the transmission channel statistics, the transmitter must make the decision of whether to drop or (re)transmit the HOL. Our goal is to find an algorithm to resolve this control dilemma.

3. GENERAL OPTIMAL OFFLINE ALGORITHM

In this section, we analyze the control dilemma of whether to transmit or drop the Head of Line (HOL) packet by modeling the system as a stochastic shortest path problem. We use dynamic programming to find the optimal solution, which is stored in a lookup table [4]. In each time slot of a transmission session, the transmitter references the lookup table to make the optimal decision of whether to drop or (re)transmit the HOL packet.

We model the wireless channels by the probability of successful transmission of a packet in each time slot. Due to fluctuations in the channel due to user mobility, interference from neighbors, as well as varying path loss and fading, these success probabilities may be time varying. We denote by $\mathbf{s}$ the vector of success probabilities, where s_i is the success probability of the channel between transmitter and the i^{th} user. We can model the channel as a finite state Markov Chain, with transition probability $q_{ss'}$ from success probabilities $\mathbf{s}$ to $\mathbf{s}'$.

We assume M packets are to be transmitted to each receiver in some predetermined order, such as coding order. Packet m is the m^{th} packet to arrive at the transmitter. The distortion value, $d_i^m \geq 0$, for each user i is stored in the header of packet m. If packet m is not received by user i, a distortion cost d_i^m is incurred. Because each user may have varying viewing capabilities, $d_i^m \neq d_j^m$ for some $i \neq j$.

Delay and timing considerations of media content must be accounted for. To prevent lengthy transmission times, we introduce a backlog cost, $\alpha B(m)$, to capture the need to empty the transmission queue as well as to prevent the media application from blocking other traffic from being serviced by the transmitter. Note that we make no restrictions on the functional form of $B(m)$. Typically, $B(m)$ is decreasing in m because the backlog cost decreases as the number of packets left to be transmitted, $M - m + 1$, decreases. α is a weight which varies the importance of distortion cost versus backlog costs. If $\alpha = 0$, we are only concerned with distortion costs and in order to ensure successful reception of all packets by all users, the transmission time can be arbitrarily high. As $\alpha \to \infty$, we are only concerned with backlog costs, and all packets will be dropped. Implicitly, α is the tradeoff factor between distortion costs and time/rate consumption.

Given the preceding formulation, we cast the control problem as an infinite horizon stochastic shortest path problem. The state is $(\mathbf{v}, m, \mathbf{s})$, where $\mathbf{v}$ is an indicator vector and $v_i = 1$ if packet m has not been received by user i and $\mathbf{s}$ is the vector of the current success probabilities. Let $J(\mathbf{v}, m, \mathbf{s})$ be the total expected cost associated with initial state $(\mathbf{v}, m, \mathbf{s})$ and the optimal policy is used. Then $J(\cdot)$ satisfies Bellman's equation which relates the optimal cost in the current state to the expected future costs given our scheduling decision:

$$J(\mathbf{v}, m, \mathbf{s}) = \alpha B(m) + \min\{E_{(\mathbf{v}',\mathbf{s}'|\mathbf{v},\mathbf{s})}[J(\mathbf{v}', m, \mathbf{s}')], \sum_i v_i d_i + E_{(\mathbf{v}',\mathbf{s}'|\mathbf{1},\mathbf{s})}[J(\mathbf{v}', m+1, \mathbf{s}')]\} \quad (1)$$

The optimal value corresponds to the scheduling decision with the minimum cost. The first term in the minimization refers to the decision to transmit the HOL packet. We take the expected value over all possible transitions of the channel state $\mathbf{s} \to \mathbf{s}'$ (with probability $q_{ss'}$) and reception vector $\mathbf{v} \to \mathbf{v}'$. The vector $\mathbf{v}$ is updated in this manner: with probability s_i the transmission is successful to user i and $v_i' = 0$; with probability $1 - s_i$, the transmission is unsuccessful to user i, so the reception indicator to user i does not change and $v_i' = v_i$. The second term in the minimization refers to the case when the HOL packet is dropped and packet $m + 1$ is transmitted. In this case, distortion is incurred for each receiver which has not yet received the HOL packet. Because no users have yet to receive packet $m + 1$, $\mathbf{v} = \mathbf{1}$. We transmit in this time slot, so $\mathbf{1} \to \mathbf{v}'$ and $\mathbf{s} \to \mathbf{s}'$in the same manner as before. Since there are M packets, our zero cost state is $J(\mathbf{0}, M+1, \mathbf{s})$ as no more packets are left to be transmitted. Therefore, no distortion cost can be incurred. Likewise, there are no packets in the transmission queue, so no backlog costs can be incurred. Hence, $J(\mathbf{0}, M+1, \mathbf{s}) = 0$. Solving the DP recursion in (1) will result in the optimal transmission policy of each HOL packet. The solution can be found using the *value iteration* method.

Proposition 1 *There exists a stationary optimal control solution to (1) which is obtainable via value iteration.*

Proof: The Bellman's recursion terminates when there are no more packets in the transmission queue left to transmit. i.e., $m = M + 1$. There is no cost for being in this state and the optimal policy will never leave this state once it reaches it. Any policy which does not empty the buffer in finite time will incur infinite cost due to the backlog pressure, $B(m)$. There exists a policy which will empty the transmission buffer and cause the Bellman's recursion to terminate in finite time. (i.e., we can drop *all* HOL packets and terminate in M time slots.) This guarantees the existence of a stationary optimal policy which is obtainable via value iteration [4]. ■

The optimal control for each possible system state, $(\mathbf{v}, m, \mathbf{s})$, is determined offline and stored in a table. In each time slot during the transmission session, the transmitter determines the system state and optimal control via a table lookup. The storage space necessary for this table is $M 2^x k^x$, where M is the number of packets to transmit, x is the number of users to transmit to, and k is the number of channel states per user. Clearly, the storage space grows very large making this difficult for implementation in many systems.

4. OPTIMAL ONLINE CONTROL IN BERNOULLI CASE

In this section we analyze a special case of the general DP formulation given above. We model the channel by i.i.d. packet losses, equivalently, by a single state Markov Chain. The success probabilities differ for each user, but are fixed across subsequent time slots. This assumption is justified in the case of slowly varying channels. In this case, the optimal control is a simple threshold policy which can be computed online. Hence, the offline computation and large storage space of the policy presented in Section 3 is unnecessary.

For ease of notation, we focus on the 2 user case. An extension to more users will follow similarly. For simplicity, let $\bar{s_i} = 1 - s_i$.

Because the success probabilities are fixed, we no longer have to track the state of the channel. The DP recursion in (1) can be rewritten as:

$$\begin{aligned} J(v_1, v_2, m) = \alpha B(m) + \min\{ \\ s_1 s_2 J(0,0,m) + \bar{s_1} s_2 J(v_1, 0, m) + \\ s_1 \bar{s_2} J(0, v_2, m) + \bar{s_1}\bar{s_2} J(v_1, v_2, m), \\ v_1 d_1 + v_2 d_2 + s_1 s_2 J(0,0,m+1) + \bar{s_1} s_2 J(1,0,m+1) + \\ s_1 \bar{s_2} J(0,1,m+1) + \bar{s_1}\bar{s_2} J(1,1,m+1)\} \end{aligned} \quad (2)$$

where the first term in the minimization corresponds to the cost of transmitting the HOL packet and the second term corresponds to the cost of dropping the HOL packet and transmitting the next packet.

We define d_i as the reduction in distortion to user i if the Head of Line (HOL) packet is received given the current system state. $\tilde{d}_i = 0$ if the packet has already been received, i.e., $v_i = 0$, and $\tilde{d}_i = d_i$ if the packet has not yet been received, i.e., $v_i = 1$.

Proposition 2 *The optimal policy for independent Bernoulli packet losses is a **Threshold** policy where the Head of Line packet is transmitted if $\alpha B(m) \leq s_1 \tilde{d}_1 + s_2 \tilde{d}_2$ and is dropped otherwise.*

Proof: The key element in this proof is to notice that if a packet is not successfully transmitted to all users, we return to the same state. The HOL packet is packet m. Suppose user 2 has already received the HOL packet, so $\tilde{d}_2 = 0$. Once user 1 receives that HOL packet, we should transmit the next packet. J is the expected total cost given a control policy which transmits the HOL packet in state $(1, 0, m)$ and uses the optimal policy in all other states. We decompose J into $\tilde{J} + K$, where K is the expected total cost given we use the optimal policy once the HOL packet is successfully transmitted and removed from the transmission queue and $\tilde{J}$ is the additional cost of transmitting the HOL packet. Then Bellman's equation reduces to:

$$\begin{aligned} \tilde{J} + K &= \alpha B(m) + s_1 s_2 K + \bar{s_1} s_2 (\tilde{J} + K) + \\ & \quad s_1 \bar{s_2} K + \bar{s_1}\bar{s_2}(\tilde{J} + K) \\ &= \frac{\alpha B(m)}{s_1} + K \end{aligned} \quad (3)$$

where the last equality follows after some algebra. The total expected cost of dropping the HOL packet is $d_1 + K$. If $\tilde{J} + K \leq d_1 + K$, the optimal policy is to transmit the HOL packet. Therefore, the optimal policy is to transmit when $\alpha B(m) \leq s_1 d_1$; otherwise, drop the HOL packet. Intuitively, we transmit when the expected reduction in distortion, $s_1 d_1$, is greater than the backlog cost, $\alpha B(m)$. Analogously, if user 1 has already received the HOL packet, we only transmit when $\alpha B(m) \leq s_2 d_2$. Based on the assumption that $d_i \geq 0$, if $\alpha B(m) \leq s_1 d_1$, the optimal policy is to transmit the HOL packet regardless if user 2 has received the HOL packet. This is because the reduction in total cost by transmitting the HOL packet will only increase when user 2 has not yet received it either.

The remaining question is what to do when both users have not received the HOL packet and $\alpha B(m) > s_i d_i$, for $i = 1, 2$. In this case, once any user receives the HOL packet, we drop it. Let $\tilde{J} + K$ denote the expected total cost given we transmit the HOL packet until at least one user receives it and then we use the optimal control for all future states. K is the expected total cost given we use the optimal policy after the current HOL packet is removed from the system, i.e., once at least one user receives it.

$$\begin{aligned} \tilde{J} + K &= \alpha B(m) + s_1 s_2 K + \bar{s_1} s_2 (d_1 + K) + \\ & \quad s_1 \bar{s_2} (d_2 + K) + \bar{s_1}\bar{s_2}(\tilde{J} + K) \\ &= \frac{\alpha B(m) + \bar{s_1} s_2 d_1 + s_1 \bar{s_2} d_2}{1 - \bar{s_1}\bar{s_2}} + K \end{aligned} \quad (4)$$

The total cost of dropping the HOL packet is $d_1 + d_2 + K$. Therefore, we transmit the HOL packet if $\tilde{J} \leq d_1 + d_2$; otherwise, we drop it. With some algebra, this reduces to transmitting when $\alpha B(m) \leq s_1 d_1 + s_2 d_2$. Similar to the previous case, the intuitive interpretation is that we transmit when the expected reduction in distortion for both users is greater than the backlog cost. This can be readily extended to encompass the case where one user, user i, has already received the HOL packet by replacing $d_i = 0$.

$\tilde{d}_i = d_i$ if user i has not yet received the HOL packet and $\tilde{d}_i = 0$ if it has. Then the optimal policy is to transmit the HOL packet if:

$$\alpha B(m) \leq s_1 \tilde{d}_1 + s_2 \tilde{d}_2 \quad (5)$$

and to otherwise drop the HOL packet. This policy can easily be extended to the case of more users and distortion measures. Suppose there are N users, then the expression on the right of the inequality would be a summation of N terms: $\sum_{i=1}^{N} s_i \tilde{d}_i$. ■

This threshold policy is a simple, optimal control for scheduling media packets with multiple distortion measures. These properties are ideal for real world implementation. Note that, unlike the offline policy in the general case, this optimal policy is independent of the transmission order of the packets. During a transmission session, the transmitter simply examines the distortion measures for the HOL packet, the backlog in the transmission buffer, and the reception information for each user. Based on this information, in each time slot, the transmitter checks if the threshold condition in (5) is satisfied; if it is, the HOL packet is transmitted, otherwise it is dropped.

5. RESULTS

In this section, we present performance results for the policy proposed in Section 4. We find the threshold policy achieves gains of over 2dB over standard benchmarks. We present our results for the JPEG2000 standard test image, Cafe. The 640x512 image is encoded using JPEG2000 in TRLCP coding order [2] with 1 tile, 3 resolutions, 1 quality layer, 3 color components, and 6 precincts for a total of 36 packets. We use the 13-tap downsampling filter developed by Scalable Video Coding effort [5] to calculate the distortion values for the low resolution viewer. Each HOL packet corresponds to a JPEG2000 packet. We assume packets arrive to the transmitter in coded order.

We compare 2 different benchmark policies. If we must guarantee delivery of each packet transmitted, a persistent policy must be used which transmits each HOL packet until all receivers successfully receive the packet. We refer to this policy as the *Persistent* policy. A major drawback to this policy is that a poor channel can lead to a bottleneck, preventing transmissions to users who have already received the HOL packet. A more intelligent strategy would be to employ an optimization framework, such as the one discussed in Section 4. However, standard policies assume a single distortion metric, so for this policy we assume that the scheduler believes both users are high resolution viewers, i.e., $d_1 = d_2$. We refer to this policy as the *T-SDM (Single Distortion Measure)* policy. We refer to our threshold policy from Section 4 which incorporates multiple distortion measures as *T-MDM (Multiple Distortion Measure)*.

We can also determine a loose *Upperbound* for our performance. Given the channel success probability and the number of transmission time slots, N, we know the expected number of packets that will be successfully transmitted to each user. We assume the high resolution user receives the packets which will reduce his distortion the most and the low resolution user receives the packets which will reduce his distortion the most. This is clearly a highly optimistic upperbound for our performance, because, unless both users transmit over correlated channels and unless the most important packets are identical for both users, it is impossible to achieve this bound. However, the channels are independent and there tends to be a large discrepancy between low resolution and high resolution packets [1].

We present performance results in terms of PSNR= $10\log(\frac{255^2}{D})$, where D is distortion in mean-squared error. In Fig. 2, we see the performance, in terms of average PSNR of both users, of our policy as compared to the benchmarks and upperbound. For our simulations, we assume the probability of successful transmission to the high resolution user is $.9$ and $.4$ to the low resolution user. We also assume a linear backlog cost, $B(m) = M - m + 1$. *T-MDM* out performs the other benchmarks by over 2dB and approaches the upperbound performance more rapidly. We can see *Persistent* performs very poorly due to the bottleneck of the low resolution user.

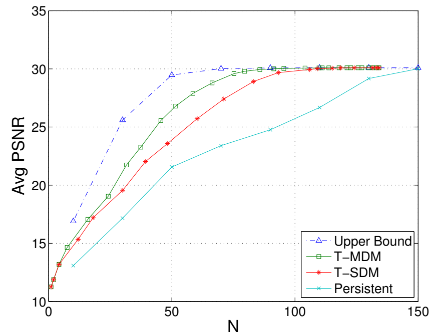

Fig. 2. Transmitting to two users with different channel qualities. Average PSNR vs. number of time slots (N).

The blocking effect is more noticeable in Fig. 3, which plot the PSNR performance for the same experiment but for the high and low resolution viewers separately. *Persistent* performs very poorly for the high resolution user, but beats all policies for the low resolution user. The low success probability to the low resolution user causes a bottleneck and blocks further transmissions to the high resolution user. *T-SDM* and *T-MDM* can overcome the blocking effect caused by the poor channel. Because *T-MDM* is more intelligent about the multiple distortion measures, with little degradation in performance for the low resolution user, the performance of the high resolution user is vastly improved. By accounting for multiple distortion measures, up to 2dB gains in average PSNR can be achieved and these gains are up to 5dB when PSNR is examined per user.

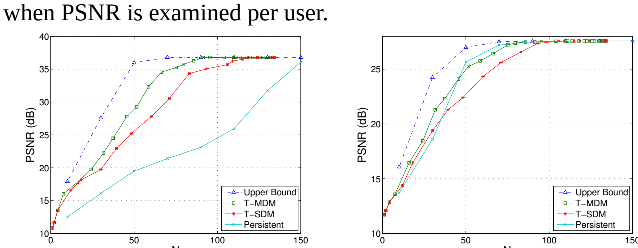

Fig. 3. Transmitting to two users with different channel qualities. PSNR vs. number of time slots (N) for high resolution viewer (left) and for low resolution user (right).

In Fig. 4, we see the average PSNR across both users as we vary s_L. We fix the number of time slots $N = 54$ and the probability of success for the high resolution user, $s_H = .9$. We vary the probability of success for the low resolution user, $.1 \leq s_L \leq .9$. When s_L is very large, most transmissions are successful to both users, so accounting for multiple distortion measures does not give significant gains. However, as the disparity between channel quality increases and blocking effects come into play, the gains of accounting for multiple distortion measures increases. When s_L is very low, the channel is so bad that both *T-SDM* and *T-MDM* ignore the low resolution user, so their performance is again similar.

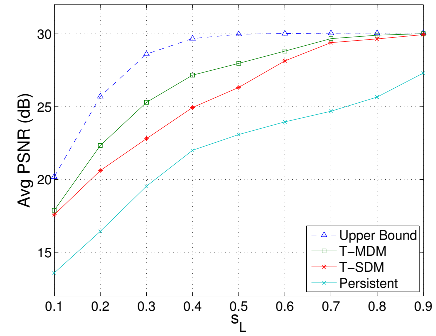

Fig. 4. Transmitting to two users with different channel qualities for a fixed number of time slots. PSNR vs. Probability of successful transmission to the low resolution user (s_L).

One's intuition may suggest that the performance of each policy is highly dependent on the transmission order of the media packets. Namely, performance should improve by transmitting packets in decreasing order of the high resolution distortion measure, or perhaps, the average of the low and high resolution distortion measures. A transmission order based on high resolution distortion measures can result in up to 1.5dB difference in average PSNR for *T-SDM* and *Persistent*. Transmitting in order of low resolution measures seriously degrades the average PSNR performance of these policies. However, these changes result in only a few tenths of dB for *T-MDM* for all packet orders: prioritization based on high or low resolution distortion measure, coding order, even random orders. This illustrates the robustness of the proposed policy to different packet orderings.

We have examined the performance gains of the threshold policy presented in Section 4. Through simulation results, we have shown that by incorporating multiple distortion measures when making scheduling decisions, up to 2dB gains can be achieved over optimal scheduling policies which only consider a single distortion measure. This emphasizes the importance of considering multiple distortion measures.

6. CONCLUSION

In this paper, we examine the transmission scenario of a server broadcasting media packets to multiple users with independent channels and varying viewing capabilities. We present a general offline algorithm to determine the optimal transmission policy. When considering the special case of Bernoulli packet losses, the optimal transmission policy can be performed online via a simple threshold policy. By accounting for multiple distortion measures, improvements of up to 2dB over standard approaches are achieved.

7. REFERENCES

[1] C. Chan, S. Wee, J. Apostolopolous. "Multipled Distortion Measures for Scalable Streaming with JPEG2000", to appear, *IEEE ICASSP* 2007

[2] ISO/IEC 15444 JPEG-2000 Image Coding System, Part 1: Core coding system.

[3] J. Apostolopoulos, "Secure Media Streaming & Secure Adaptation for Non-scalable Video", in *IEEE ICIP*, Oct. 2004.

[4] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1 & 2, 2nd. Ed., Athena Scientific, 2000.

[5] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, "AHG Report on Spatial Scalability Resampling," Document: JVT-Q007, October 2005.