

# Optimal Packet Scheduling and Rate Control for Video Streaming

Eren Gürses<sup>a</sup>, Gozde Bozdagi Akar<sup>b</sup> and Nail Akar<sup>c</sup>

<sup>a</sup>Center for Quantifiable Quality of Service in Communication Systems<sup>†</sup>, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

<sup>b</sup>Department of Electrical and Electronics Eng., Middle East Technical University, Ankara, Turkey

<sup>c</sup>Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey

## ABSTRACT

In this paper, we propose a new low-complexity retransmission based optimal video streaming and rate adaptation algorithm. The proposed OSRC (Optimal packet Scheduling and Rate Control) algorithm provides average reward optimal solution to the joint scheduling and rate control problem. The efficacy of the OSRC algorithm is demonstrated against optimal FEC based schemes and results are verified over TFRC (TCP Friendly Rate Control) transport with ns-2 simulations.

**Keywords:** video streaming, rate-distortion optimization, rate control, packet scheduling, average reward optimality, Markov decision processes

## 1. INTRODUCTION

In the recent years, there has been a phenomenal growth in the amount of video data delivered over the Internet. However Internet is inherently a lossy packet network with a dynamic nature where channel statistics and available bandwidth change in all time scales. This led to the rise of network-adaptive video streaming applications which are capable of making bit rate adaptation while providing a distortion control mechanism in order to reduce the effect of packet losses. There has been a great research effort on distortion control mechanisms for video streaming over both wireless and wired channels which have inherently different error patterns and different distortion control requirements. In wired video streaming both forward error correction (FEC) and multiple description (MD) coding gained a great popularity in recent years, however they suffer from a rate-distortion penalty as a result of introduced coding redundancy. Disadvantages of FEC and even MD coding led to an increased interest in using intelligent ARQ (Automatic Repeat Request)<sup>1,5,8</sup> and hybrid FEC/ARQ<sup>7,9</sup> based streaming techniques for non-interactive applications. An intelligent ARQ scheme<sup>1</sup> that make use packet deadlines as a prioritization metric is proposed, nevertheless it is heuristic-based and no optimization is sought. The pioneering work by Podolsky et. al.<sup>4</sup> sought the optimal solution using Markov Decision Process (MDP), however it is far from practicality. Later in Chou's seminal paper<sup>5</sup> a very flexible and practical MDP framework based on the idea of transmitting packets in a rate distortion optimized (RaDiO) manner that decides which packet to transmit in order to minimize a Lagrangian cost function that is calculated by using rate and distortion information, channel statistics, packet deadlines and transmission history. In succeeding works, variants of RaDiO algorithm for path diversity<sup>6</sup> and hybrid FEC/ARQ<sup>7,9</sup> based streaming are proposed. The performance of retransmission based algorithms depends very much on the observability of the system state which is supplied by ACKs. In case of missing ACKs the sender may estimate the receiver status by using a probability distribution and calculates optimal policies using a partially observable Markov decision process (POMDP) model.<sup>8</sup> On the other hand adaptive FEC allocation has been widely investigated in the recent years. Cavusoglu et. al<sup>10</sup> propose a low complexity, content adaptive FEC allocation algorithm however optimal rate control is not sought. Hence it utilizes RTP/UDP transport and solves the congestion control problem by core based solutions, in particular using DiffServ. However in neither of these works the optimal rate control (adaptation) parameter is sought. In the literature the closest work to our algorithm is the proposed joint source/FEC rate selection technique.<sup>11</sup> They provide optimal trade-off between source and FEC code bits within a given total bandwidth constraint in a media

---

<sup>†</sup>Center for Quantifiable Quality of Service in Communication Systems, Center of Excellence, is appointed by The Research Council of Norway, and funded by the Research Council, NTNU and Uninett

Further author information: (Send correspondence to E. Gürses)

E.Gürses: E-mail: gurses@q2s.ntnu.no, Telephone: +47 73 59 27 23

G.B.Akar: E-mail: bozdagi@eee.metu.edu.tr, Telephone: +90 312 2102341

N.Akar: E-mail: akar@ee.bilkent.edu.tr, Telephone: +90 312 2902337

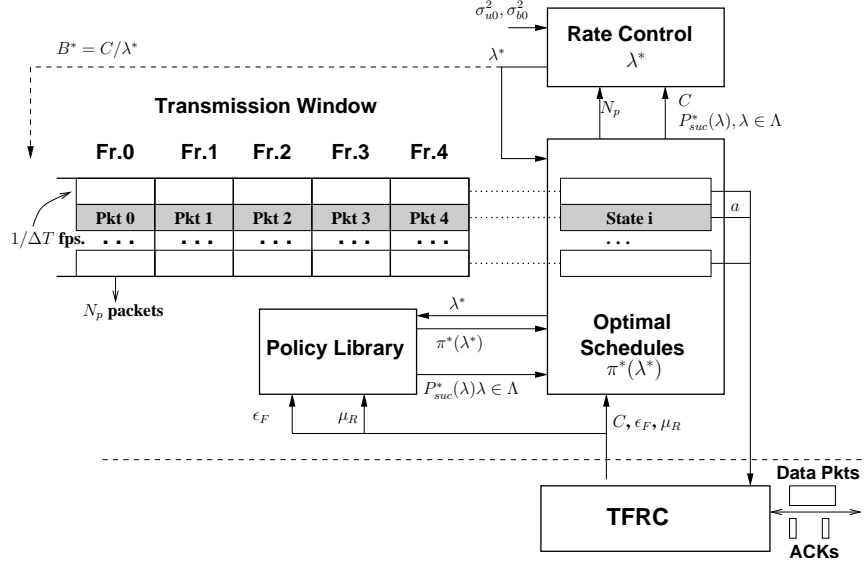


Figure 1. System architecture for the OSRC algorithm

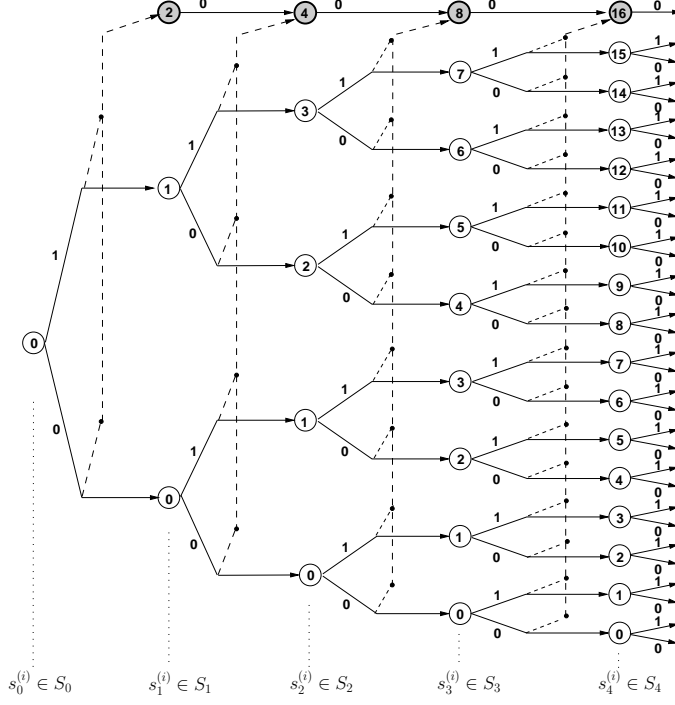
independent manner. However our approach is ARQ based which is known to be more efficient in delay tolerant application yet more complex. In this work, we focus on non-interactive, retransmission based video streaming applications where the proposed OSRC framework decouples the problem of calculating optimal packet schedules (OS) from the problem of making optimal rate control (RC) by using the analytical video distortion model<sup>2</sup> which is also utilized in recent works.<sup>3,8</sup> Some major features of the OSRC algorithm is given as follows where  $\lambda^*$  can be interpreted as the number of bits sent per 1 source bit: *i*) low streaming complexity due to use of a priori calculated optimal policy  $\pi^*(\lambda^*)$ , *ii*) increased granularity for the selection of redundancy  $\lambda^*$ , *iii*) capability of handling fragmentation of frames into packets, *iv*) no extra quality fluctuation introduced due OSRC distortion control, *v*) utilization of a realistic distortion model, *vi*) applicable to streaming of real-time encoded/transcoded and scalable coded video like FGS. In Section 2, we present the system architecture. Simulation results are given in Section 3 and finally we provide the concluding remarks and future research directions in Section 4.

## 2. SYSTEM ARCHITECTURE

The main goal of the OSRC algorithm is to calculate and apply the optimal transmission policy  $\pi^*(\lambda^*)$  which minimize the average distortion while regulating the video bit rate  $B^*$  by using the rate constraint  $\lambda^* = C/B^* \geq 1$  for bandwidth estimate  $C$ . Hence when the policy  $\pi^*(\lambda^*)$  is used the remaining bandwidth  $C - B^*$  is then utilized by packet retransmissions in order to control the distortion in case of packet losses. A general overview of the OSRC system architecture is given in Figure 1, where  $T_p$  is initial playout buffer time,  $1/\Delta T$  is video frame rate,  $M$  is the number observed frames in the transmission window and  $\epsilon_F, \epsilon_B$  are the estimates of forward and backward channel packet error rate respectively. In the OS part optimal schedule (policy)  $\pi^*(\lambda)$  that maximizes the success probability of packets and conforms to rate constraint  $\lambda$  are calculated such that the resulting optimal policy will yield the optimal success probability  $P_{suc}^*(\lambda)$ . This step is done a priori in an off-line manner for different values of channel error  $\epsilon_F, \epsilon_B$ , mean round trip time  $\mu_R$ , rate constraint  $\lambda$ . The resulting  $\pi^*(\lambda)$  and  $P_{suc}^*(\lambda)$  for  $\lambda \in \Lambda$  are stored in the Policy Library in Fig 1 for later use of the RC algorithm. On the other hand, optimal rate control RC is an online algorithm which calculates the optimal  $\lambda^*$  by using the optimal success probabilities  $P_{suc}^*(\lambda)$ , video distortion model parameters  $\sigma_{u0}^2, \sigma_{b0}^2$ , bandwidth  $C$  and number of packets  $N_p$  per frame.

### 2.1. Optimal Packet Scheduling Problem, $\pi^*(\lambda)$

The optimal policy is calculated by first developing an MDP model which captures the behavior of the transmission window on a per-packet basis as illustrated with darker color packets in Figure 1, and then solving it by means of constrained optimization techniques which maximize a reward function while conforming to a constraint. In the MDP formulation  $i = (s_0^{(i)}, s_1^{(i)}, \dots, s_{M-1}^{(i)})$  denotes the current system state of the transmission window. The state variable  $s_k^{(i)} \in S_k$



**Figure 2.** State variables ( $s_0^{(i)}, s_1^{(i)}, \dots, s_{M-1}^{(i)}$ ) of the OS algorithm for  $M = 5$

for  $k \in \{0, \dots, M-1\}$  represents the state (i.e. transmission and the observation history) of the  $k$  previous packet in transmission window as in Figure 2 where  $S_0 = \{0\}$ ,  $S_1 = \{0, 1, 2\}$ ,  $S_2 = \{0, \dots, 4\}$ ,  $S_3 = \{0, \dots, 8\}$ ,  $S_4 = \{0, \dots, 16\}$  are for state variables and  $\mathcal{I} = S_0 \times S_1 \times \dots \times S_{M-1}$  is for the overall state space. The labels 1 and 0 on the arrows respectively indicate that whether packet is transmitted or not at that transmission opportunity, and by following back to the initial state of the trellis, the transmission history of a packet can be pointed out. Hollow and darker circles in the trellis represent non-final and final states respectively where a final state means an acknowledgment is already received for that packet. On the other hand the action taken at state  $i \in \mathcal{I}$  is denoted as  $a = [a_0, \dots, a_{M-1}]$  where  $a_k \in \{0, 1\}$  is the action for the  $k$ th packet in transmission window and action 1 indicate a packet transmission and 0 otherwise. Finally state transition probabilities  $P_{ij}(a) = \prod_{k=0}^{M-1} P_{ij}(a, k)$  completes the definition of the underlying Markov chain where  $j$  is defined as the next state. Later with the definition of reward and cost functions, it would be possible to calculate the optimal policies that maximize the expected reward function while the expected cost is bound to a constraint over this Markov chain. Given  $(i, a)$  as a state action pair, cost function  $c(i, a) = \sum_{k=0}^{M-1} a_k$  denotes the number of transmitted packets at state  $i$  under action  $a$  and the expected value of it is constrained by  $\lambda$ . On the other hand reward function  $r(i, a)$  in Eqn. 1 is defined as the success probability of oldest packet (i.e.  $M-1$  th) which can be calculated by setting  $k = M-1$  and using expression for error probability in Eqn. 2. Finally to simplify the notation, for any non-final state  $s_k^{(i)}$  we define  $q_{s_k^{(i)}}(a, m)$  for  $m < k$  and  $q_{s_k^{(i)}}(a, k) = a_k$  to denote the transmission history and current action for packet  $k$  respectively.

$$r(i, a) = 1 - \epsilon_{s_{M-1}^{(i)}}(a, M-1) \quad (1)$$

$$\epsilon_{s_k^{(i)}}(a, k) = \begin{cases} \prod_{m=0, q_{s_k^{(i)}}(a, m)=1}^k P\{FTT > T_p - m\Delta T | RTT > (k-m)\Delta T\} & , s_k^{(i)} \text{ non-final} \\ 0 & , s_k^{(i)} \text{ final} \end{cases} \quad (2)$$

State transition probability  $P_{ij}(a, k)$  for packet  $k$  in the transmission window is given by Eqn. 3 and for non-final states  $s_k^{(i)}$  probability of not reaching a final state under action  $a$  is given by  $p_{s_k^{(i)}}(a, k)$  in Eqn. 4

$$P_{ij}(a, k) = \begin{cases} 1 & , s_k^{(i)} \text{ final} \\ 1 - p_{s_k^{(i)}}(a, k) & , s_k^{(i)} \text{ non-final, } s_{k+1}^{(j)} \text{ final} \\ p_{s_k^{(i)}}(a, k) & , s_k^{(i)} \text{ non-final, } s_{k+1}^{(j)} \text{ non-final} \\ 0 & , s_{k+1}^{(j)} \text{ not reachable from } s_k^{(i)} \end{cases} \quad (3)$$

The conditional probability in Eqn. 4 expresses the probability of not receiving an ACK during state transition  $s_k^{(i)}$  to  $s_{k+1}^{(j)}$  for the  $k$ th frame given that the current state  $s_k^{(i)}$  is not a final state. The condition  $RTT > (k - m)\Delta T$  ensures that no ACK is received for the  $m$ th transmission of  $k$ th frame until state  $s_k^{(i)}$  (i.e.  $q_{s_k^{(i)}}(a, m) = 1$ ).

$$p_{s_k^{(i)}}(a, k) \triangleq \prod_{\substack{m=0, \\ q_{s_k^{(i)}}(a, m)=1}}^k P\{RTT > (k - m + 1)\Delta T | RTT > (k - m)\Delta T\} \quad (4)$$

Hence given the above definitions the average reward optimal policy  $\pi^*(\lambda)$  for rate constraint  $\lambda$  is calculated by using linear programming as formulated in Eqn.5 where the  $\pi_a(i)^*$  denotes the probability of selecting action  $a \in A(i)$  (i.e. the policy) at state  $i \in \mathcal{I}$  and  $x_{ia}$ 's represent the steady state frequency (probability) of visiting a state action pair  $(i, a)$ .

**Step 1.** Calculate optimal basic solution  $x_{ia}^*$  using the simplex algorithm.

$$P_{suc}^*(\lambda) = \max \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}(i)} r(i, a) x_{ia} \quad (5)$$

$$s.t. \begin{cases} \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}(i)} c(i, a) x_{ia} \leq \lambda \\ \sum_{a \in \mathcal{A}(j)} x_{ja} - \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}(i)} P_{ij}(a) x_{ia} = 0, j \in \mathcal{I} \\ \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}(i)} x_{ia} = 1 \\ x_{ia} \geq 0, \text{ for all } i, a \end{cases}$$

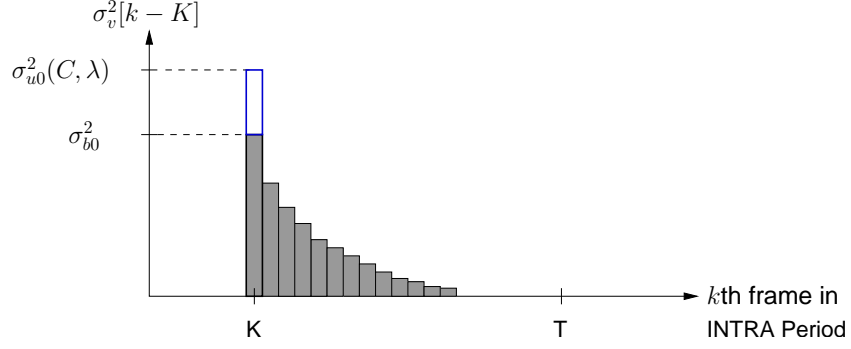
**Step 2.** For any state  $i \in I_0 = \{i | \sum_{a \in \mathcal{A}(i)} x_{ia}^* > 0\}$  calculate optimal policy  $\pi_a(i)^*$  by:

$$\pi_a(i)^* = \begin{cases} x_{ia}^* / \sum_{a' \in \mathcal{A}(i)} x_{ia'}^*, & a \in \mathcal{A}(i) \text{ and } i \in I_0 \\ \text{arbitrary,} & \text{otherwise} \end{cases} \quad (6)$$

## 2.2. Optimal Rate Control Problem, $\lambda^*$

Optimal rate control problem is denoted as the selection of the optimal rate constraint parameter  $\lambda^* \in \Lambda$  which minimizes the average video distortion by using a priori calculated average packet success rates  $P_{suc}^*(\lambda)$  corresponding to optimal policies  $\pi^*(\lambda)$  for  $\lambda \in \Lambda$  calculated in Section 2.1. Furthermore according to the assumption on fragmentation of frames into  $N_p$  network packets the effective frame success rate should be defined by using the packet success rate as follows,  $P_{f_{suc}}^*(\lambda) = P_{suc}^*(\lambda)^{N_p}$  and used in the distortion calculations conducted in this chapter. This relation between frame and packet success rate is acceptable since the transport layer discards the partially received transport layer packets due to fragmentation in order to preserve its data integrity. However more advanced distortion calculations that remove this constraint can also be performed.

The distortion model<sup>2</sup> assumes a simple error concealment scheme which replaces the lost frame with the previous one and expresses the total distortion  $D_{tot}$  for video over lossy channels as the sum of distortions  $D_e$  and  $D_v$  which are introduced by signal compression and packet losses respectively. In the case of streaming FGS coded video, the initial distortion (MSE) term  $\sigma_{u0}^2(C, \lambda)$  can be split into two parts such as  $\sigma_{u0}^2(C, \lambda) = \sigma_{b0}^2 + \sigma_{e0}^2(C, \lambda)$  as illustrated in Figure 3. The terms  $\sigma_{b0}^2$  and  $\sigma_{u0}^2(C, \lambda)$  can be calculated a priori for FGS encoded videos where they respectively correspond to the MSE of base layer (BL) and BL+EL when simple error concealment scheme is applied. Hence the remaining term is simply calculated as  $\sigma_{e0}^2(C, \lambda) = \sigma_{u0}^2(C, \lambda) - \sigma_{b0}^2$ . In case of a packet loss base layer distortion  $\sigma_{b0}^2$  appears at the



**Figure 3.** Illustration of the modified distortion model for FGS video

base layer as a result of error concealment process. Since base layer is compressed with a predictive encoder, the base layer distortion  $\sigma_{b0}^2$  propagates to the subsequent frames, whereas the remaining term  $\sigma_{e0}^2(C, \lambda)$  that appear due to the concealment of FGS data is constrained only to the lost frame. As a result of this fact the distortion signal due to the loss of  $K$ th frame is represented as  $\sigma_v^2[k - K]$  in Eqn. 7 where the impulse term  $\delta[k]$  represents the extra distortion introduced by the concealment of the FGS part of the video and the remaining term is the propagated error signal as given in.<sup>2</sup>

$$\sigma_v^2[k] = \sigma_{e0}^2(C, \lambda)(1 - P_{fsuc}^*(\lambda))\delta[k] + \sigma_{b0}^2(1 - P_{fsuc}^*(\lambda))\frac{1 - \gamma k}{1 - \beta k}, \text{ for } k \in [0, T - 1] \quad (7)$$

The parameters  $\beta = 1/T$  and  $\gamma$  in Eqn. 7 correspond to the intra frame rate and leakage factor, respectively, where  $\beta, \gamma \in [0, 1]$ . Hence  $D_v(C, \lambda)$  can be obtained by the superposition of  $T$  shifted error signals in Eqn. 7 under the assumption that the decoder is linear and error signals are uncorrelated. On the other hand  $D_e(C, \lambda)$  can be obtained from the rate-distortion curve of the given video.

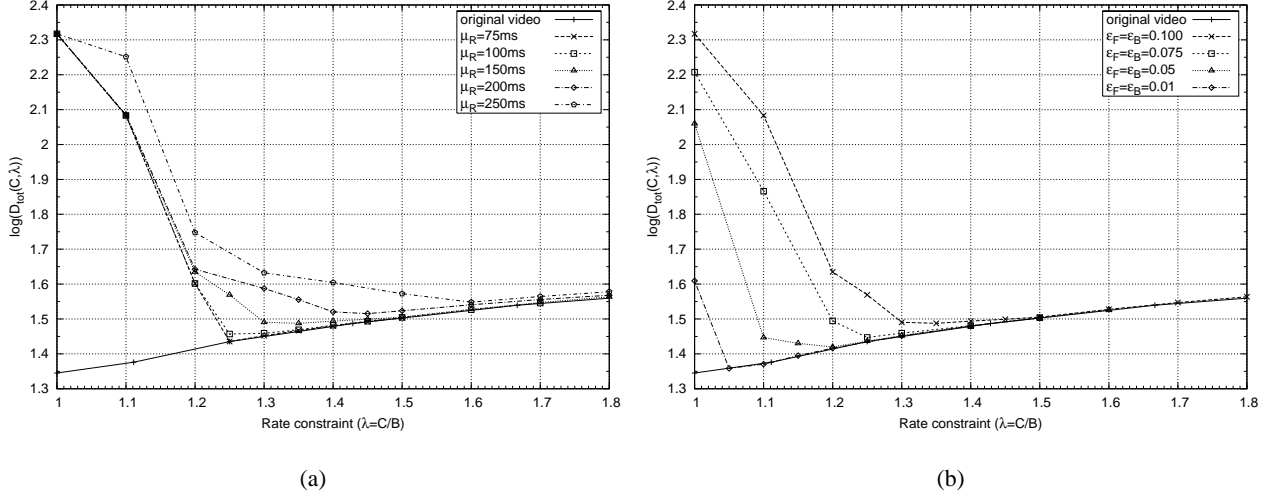
$$D_v(C, \lambda) = \sigma_{e0}^2(C, \lambda)(1 - P_{fsuc}^*(\lambda)) + \sigma_{b0}^2(1 - P_{fsuc}^*(\lambda)) \sum_{t=0}^{T-1} \frac{1 - \gamma t}{1 - \beta t} \quad (8)$$

Finally, the optimal rate control (RC) is simply achieved by finding the optimal rate constraint  $\lambda^*$  which minimizes the total distortion  $D_{tot}(C, \lambda) = D_e(C, \lambda) + D_v(C, \lambda)$  or maximizes the  $PSNR(C, \lambda) = 10 \log(255^2 / D_{tot}(C, \lambda))$  as given in Eqn. 9. However the number of fragments  $N_p$  is depending on the  $B^* = C/\lambda^*$ , MTU (Maximum Transmission Unit) size and frame rate. Since MTU size and frame rate is fixed for a streaming session and  $C$  is given, the value of  $N_p$  is determined only by the  $\lambda^*$ . Hence the minimization procedure of the RC algorithm in Eqn. 9 is iterated until the value for  $\lambda^*$  converges.

$$\lambda^* = \arg \min_{\lambda \in \Lambda} D_e(C, \lambda) + D_v(C, \lambda) = \arg \max_{\lambda \in \Lambda} PSNR(C, \lambda) \quad (9)$$

### 3. SIMULATION RESULTS

The simulation results for the OSRC framework are obtained by using both a *model based simulator* and standard *ns-2 simulator*. The model based simulator simply applies packet error and delay using a probabilistic channel model, and is extremely useful for showing the relations between optimization parameters such as, packet error rate, mean RTT and  $\lambda$ . However model based simulator doesn't reflect the dynamics of a real network, for this reason in the last part of this section, we present ns-2 simulation results in order to show the adaptivity and TCP friendliness of the proposed OSRC algorithm. Both model based and ns-2 simulations are conducted by using an I-P-P coded H.264 video with an MPEG-4 FGS encoded bit stream as enhancement layer from FOREMAN, CARPHONE sequences for four times sampled at a



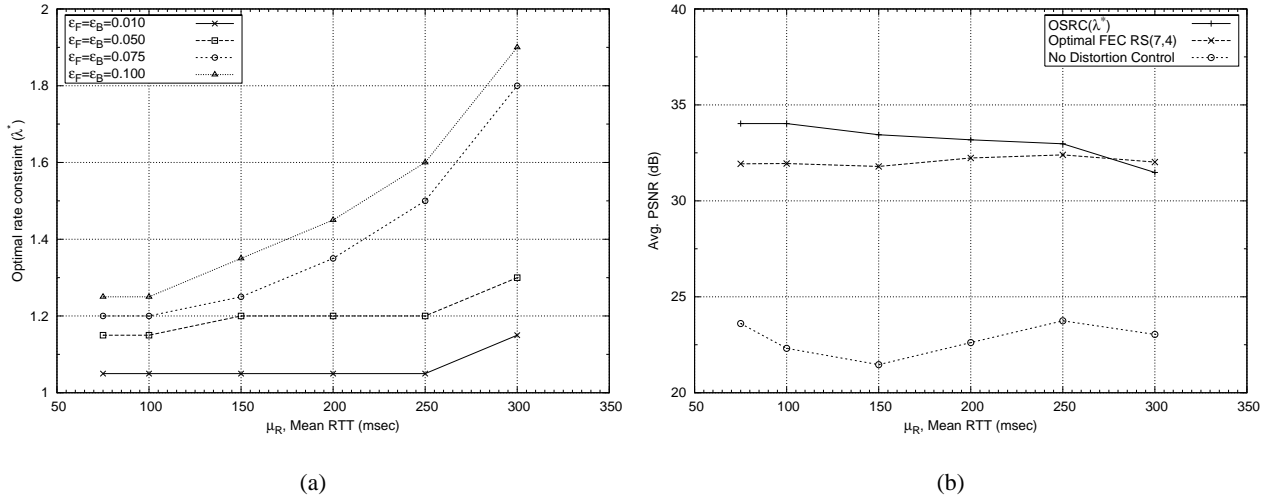
**Figure 4.** For  $C = 20000$  bytes/sec plots of  $\log(D_{tot}(C, \lambda))$  for (a) different  $\mu_R$  at a given  $\epsilon_F = \epsilon_B = 0.1$ , (b) different  $\epsilon_F = \epsilon_B$  at a given  $\mu_R = 150$  msec.

frequency of 10 fps which correspond to 1044 frames where  $T = 50$ ,  $\gamma = 0.55$ ,  $M = 5$  and  $T_p = 1000$  msec. The policy library is constructed for the following sets where  $\mu_R \in \{75, 100, 150, 200, 250, 300\}$ ,  $\epsilon_F = \epsilon_B = \{0.01, 0.05, 0.075, 0.1\}$  and  $\Lambda = \{1.00, 1.05, 1.10, 1.15, 1.20, \dots, 2.00\}$ .

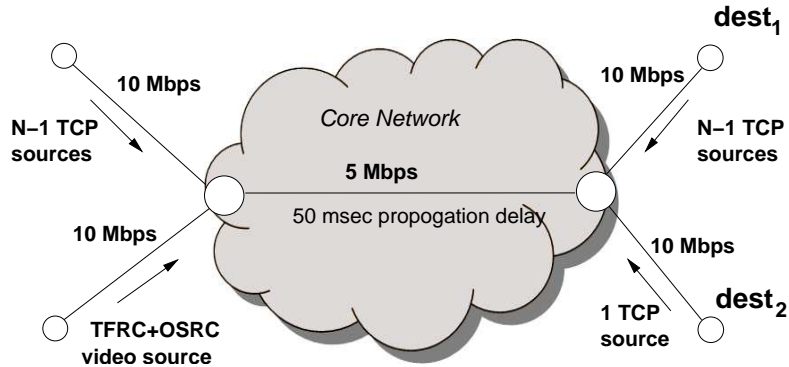
In the model based simulations we fixed the available bandwidth estimate to  $C = 20000$  bytes/sec. However fixing  $C$  is just done for reducing the number of unknowns so that the reader can observe the relations between  $\lambda^*$  and the channel error  $\epsilon_F$ , mean delay  $\mu_R$  throughout Figures 4-5. Later in ns-2 simulations of this section, OSRC algorithm is run by adapting to available network rate  $C$  which is determined by the network dynamics. Given  $C = 20000$  bytes/sec,  $\log(D_{tot}(C, \lambda))$  is plotted for  $\lambda \in \Lambda$  for different  $\mu_R$  values in Figure 4(a) while fixing  $\epsilon_F = \epsilon_B = 0.1$ . Similarly  $\log(D_{tot}(C, \lambda))$  is observed for different  $\epsilon_F = \epsilon_B$  values in Figure 4(b), while fixing  $\mu_R = 150$  msec. Hence the values of the optimal rate constraint  $\lambda^*$  that minimize  $D_{tot}(C, \lambda)$  are calculated by RC algorithm and easily observable in both plots in Figure 4 for different parameter sets. In Figure 4(a) since loss rate  $\epsilon_F$  is fixed, for case  $\lambda = 1$  the observed distortion  $\log(D_{tot}(C, \lambda))$  happens to be similar for all  $\mu_R$  cases. On the other hand in Figure 4(b) all curves correspond to different channel errors  $\epsilon_F$  and as a result their initial values at  $\lambda = 1$  happen to be different.

On the other hand Figure 5(a) illustrates the optimal rate control parameter  $\lambda^*$  calculated by RC algorithm for different error and channel delay conditions. This figure simply reveals that with the increase in packet error rate  $\epsilon_F$  and/or average network delay  $\mu_R$ , the required amount of redundancy increases. Later in Figure 5(b) we compare the performance of optimal FEC allocation scheme and OSRC algorithm for a fixed channel error  $\epsilon_F = 0.10$  while varying the mean channel delay  $\mu_R$ . We use optimal FEC code rate allocation scheme in our simulations since it inherently enables a rate control algorithm where the rate is determined by code rate  $k/n$  (like Reed-Solomon  $RS(n, k)$ ). FEC code rate  $k/n$  would be analogous to OSRC with  $\lambda = n/k$  in terms of the amount of redundancy introduced. In our FEC results we fix the number of source packets to  $k = 4$  and determine the optimal number of total packets  $n^* = 7$ . First  $D_{tot}(C, n/k)$  can simply be calculated by replacing  $P_{f_{suc}}^*(\lambda)$  in Eqn. 8 with the success probability of corresponding FEC code rate and then  $n^*$  can be found by using Eqn. 9. Hence, in our simulations OSRC( $\lambda^*$ ) can achieve lower distortion levels (i.e. higher PSNR) at the same bandwidth  $C$  for  $\mu_R \in [0, 300]$  msec which is illustrated in Figure 5(b). On the contrary as  $\mu_R$  gets closer to  $M \cdot \Delta T$  retransmission based schemes like the proposed OSRC, yield worse results than FEC.

In ns-2 simulations we use OSRC( $\lambda^*$ ) streaming which adapts to network conditions by calculating the optimal rate constraint  $\lambda^*$  according to the observed parameters  $C$ ,  $\mu_R$  and  $\epsilon_F, \epsilon_B$ . In ns-2 simulations we conduct adaptive OSRC( $\lambda^*$ ) simulations over TFRC transport for the topology given in Figure 6 where  $N \in \{10, 15, 20, 25, 30, 35, 40\}$  sources sharing a bottleneck link of 5 Mbps. with 50 msec end-to-end propagation delay. In ns-2 simulations, channel error, RTT and bandwidth dynamically change and the OSRC framework switches between the optimal policies  $\pi^*(\lambda^*)$  corresponding to optimal rate control parameter  $\lambda^*$  which changes adaptively due to changing channel statistics and bandwidth estimate.



**Figure 5.** For  $C = 20000$  bytes/sec. (a) optimal rate constraint  $\lambda^*$  for different  $\mu_R$  and  $\epsilon_F = \epsilon_B$ , (b) Comparison for OSRC( $\lambda^*$ ) and optimal FEC (i.e. RS(7,4)) at a fixed channel error rate of  $\epsilon_F = \epsilon_B = 0.1$  for different  $\mu_R$



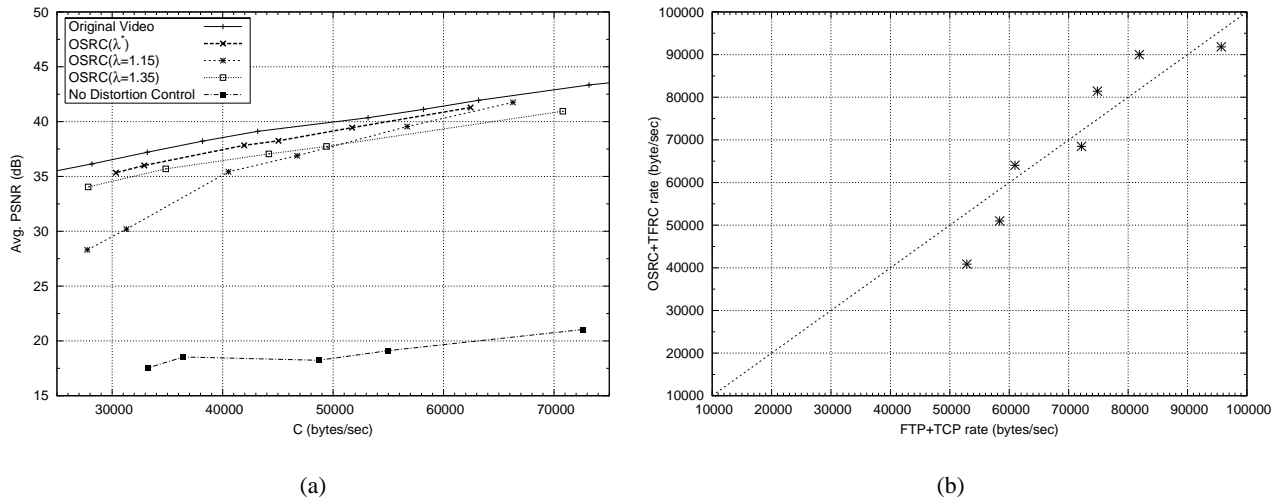
**Figure 6.** Topology used in ns-2 simulations

Figure 7(a) compares the efficacy of adaptive OSRC( $\lambda^*$ ) with *no-distortion* control schemes and the static versions of OSRC( $\lambda$ ) for  $\lambda \in \{1.15, 1.35\}$  where no optimal rate control is sought.

Finally we compare the bandwidth fairness of an OSRC+TFRC source directly with an infinite TCP source (i.e. FTP+TCP). In this experiment, rather than fixing the capacity of the Core Network to 5 Mbps. as given in Figure 6 and varying  $N$ , we fixed  $N = 2$  and changed the link's capacity for values  $\{0.750, 0.875, 1.0, 1.125, 1.250, 1.375, 1.5\}$  Mbps. The fairness between two source OSRC+TFRC and FTP+TCP is demonstrated in Figure 7(b) by comparing the throughput of these sources which respectively means OSRC schedules multimedia packets over a TFRC based transport and FTP transmits infinite data over a TCP transport.

#### 4. CONCLUSION

In this work we propose a low complexity and retransmission based optimal video streaming and rate adaptation algorithm. Efficacy and the adaptivity of the OSRC algorithm is shown against optimal FEC and static version of the OSRC algorithm using ns-2 simulations. Furthermore we conclude that the use of OSRC over TFRC based transport has no adverse effect on the TCP-friendliness. As a future work, MDP based framework is applicable to hybrid wired/wireless networks where packet drops due to bit errors are minimized by adaptive techniques<sup>12</sup> or adaptive coded modulation techniques. Furthermore with the aid of more accurate RTO<sup>13</sup> (Retransmission Timeout) estimations, the complexity of the MDP problem can be reduced with the removal of unnecessary transmission opportunities.



**Figure 7.** (a) ns-2 simulation results for OSRC( $\lambda^*$ ) with policy switching (b) Fairness comparison between the OSRC+TFRC and the FTP+TCP sources

## REFERENCES

1. S. H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," in *PV 2002*, Pittsburgh, 2002.
2. K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, June 2000.
3. J. Chakareski et. al., "Low-Complexity Rate-Distortion Optimized Video Streaming," *IEEE ICIP*, October 2004.
4. M. Podolsky, S. McCanne and M. Vetterli, "Soft ARQ for layered streaming media," Technical Report UCB/CSD-98-1024, University of California, Computer Science Division, Berkeley, November 1998.
5. P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 390-404, April 2006.
6. J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proc. IEEE Data Compression Conference*, Snowbird, UT, April 2003.
7. J. Chakareski and P. A. Chou, "Application layer error correction coding for rate-distortion optimized streaming to wireless clients," *IEEE Transactions on Communications*, vol. 52, no.10, Oct. 2004.
8. D. Tian, X. Li, G. Al-Regib, Y. Altunbasak and J. R. Jackson, "Optimal packet scheduling for wireless video streaming with error-prone feedback," in *Proc. IEEE WCNC*, Atlanta, March 2004.
9. H. Seferoglu, Y. Altunbasak, O. Gurbuz and O. Ercetin, "Rate distortion optimization with adaptive FEC for real-time wireless multimedia," in *Proc. IEEE ICC*, May 2005.
10. B. Cavusoglu, D. Schonfeld, R. Ansari and D. K. Bal, "Real-time low-complexity adaptive approach for enhanced QoS and error resilience in MPEG-2 video transport over RTP networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, November 2005.
11. P. Frossard and O. Verscheure, "Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery," *IEEE Transaction on Image Processing*, vol. 10, no. 12, December 2001.
12. T. Yoo, R. J. Lavery, A. Goldsmith and D. J. Goodman, "Throughput optimization using adaptive techniques," submitted to *Communication Letters*, January 2006.
13. A. C. Begen and Y. Altunbasak, "An adaptive media-aware retransmission timeout estimation method for low-delay packet video," in *IEEE Transactions on Multimedia*, accepted for publication July 3, 2006.