JOINTLY OPTIMAL REFERENCE FRAME & QUALITY OF SERVICE SELECTION FOR H.26L VIDEO CODING OVER LOSSY NETWORKS

Gene Cheung *Connie Chan*

Hewlett-Packard Laboratories, Japan

ABSTRACT

In new video coding standards such as H.26L, predicted frame has the flexibility to select its reference frame from a number of previous frames for motion prediction. In this paper, we propose an optimization algorithm using dynamic programming that jointly exploits this flexibility with available network QoS for optimal streaming performance. A rounding technique is employed to scale the complexity of the algorithm down at the expense of gracefully degrading solution quality. Results show significant streaming quality improvement over an ad-hoc scheme.

1. INTRODUCTION

This paper is concerned with the problem of optimal transport of standard-compliant video stream over lossy networks for real-time playback. In particular, we consider the scenario where the application is very delay-jitter sensitive, to the extent that it cannot tolerate even one end-to-end packet retransmission. One reason can be that a small initial playback buffer is employed at the client side, meaning any retransmitted packet upon client request will miss its playback deadline and hence be rendered useless. For quality of service (QoS) enabled networks [1], the conventional approach for this streaming scenario is to designate more important frames to better network service classes. (For example, designate I-frames to a better service class than P-frames.) For networks without QoS, end hosts can mimic a QoS network by applying forward error correction (FEC) of different strengths to different frames. We consider both cases under the same formulation in the paper.

The video coding standard we are focusing on is H.26L [2], a new standard that offers many coding flexibilities for better coding and streaming performance. One of these flexibilities is the multi-frame prediction support, where each P-frame can choose among a number of frames for motion prediction. At the cost of coding efficiency, using a frame further in the past for motion prediction can potentially avoid error propagation due to packet loss. Given the described streaming scenario and the chosen video coding standard, the research problem we are investigating is: what is the jointly optimal selection of reference frame and quality of service for optimal performance? After discussing related work in section 2, we formulate it as a formal optimization problem in section 3. Given the problem is NP-hard (proof in section 7.1), we present an approximate algorithm in section 4. Results and conclusion are presented in section 5 and 6 respectively.

2. PREVIOUS WORK

Streaming media is a well studied topic, and we refer readers to [3] [4] and their exhaustive references for general background. We instead focus on a few selected aspects in this section.

H.26L [2] is a new video coding standard that has demonstrably superior coding performance over existing standards such as MPEG-4 and H.263 over a range of bit rates. As part of the new standard definition is the flexibility of using any arbitrary frame to perform motion-prediction, originally introduced as Annex N in H.263+ and later as Annex U in H.263++. Early work on optimizing streaming quality using reference frame selection includes [3] [5]. We differ from these work in that we jointly optimize streaming using both reference frame (RF) and quality of service (QoS).

Unlike many previous rate-distortion optimization algorithms [4] [6] which rely on the use of Lagrange multipliers, our optimization is unique in that we use a rounding technique which trades off complexity with the quality of the obtained solution, relieving us of the necessity to find a suitable Lagrange multiplier.

3. PROBLEM FORMULATION

We formally formulate the RF / QoS selection problem as an optimization problem in this section. We first discuss the source model used for the encoded video stream, then the network model for QoS networks in our streaming scenario.

3.1. Source Model

We model the decoding dependencies of the encoded media source using a directed acyclic graph (DAG) model $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$, similar to one used in [4]. More concretely, the streaming media is represented by a collection of frames, F_i, $i \in \{1, \ldots, |\mathcal{V}|\}$. Each frame F_i, represented by a node i in G, has a set of outgoing edges $e_{i,j} \in \mathcal{E}$ to nodes j's, representing the possible RFs F_j's from which F_i can choose. We designate a 0-1 variable $x_{i,j}$ to be 1 if F_i uses F_j as RF:

$$x_{i,j} = \begin{cases} 1 & \text{if } F_j \text{ is selected as RF for } F_i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Because each P-frame F_i can only have one RF, we have the following *RF constraint*:

$$\sum_{j \mid e_{i,j} \in \mathcal{E}} x_{i,j} = 1 \qquad \forall i \in \mathcal{V} \tag{2}$$

We assume that only frames in the past are used for reference, i.e. $\forall e_{i,j} \in \mathcal{E}, i > j$. We also assume only frame 1 is intra-coded, and hence $\nexists e_{1,j} \in \mathcal{E}$. An example of a DAG model of a 4-frame sequence is shown in Figure 1.

As a frame F_i uses a RF F_j further in the past, the encoding rate of F_i is likely to increase since the temporal distance between the predicted frame and RF has increased. We model this change in encoding rate by scalar $r_{i,j}$, denoting the encoding rate of F_i if F_j is used as reference. $r_{1,1}$ denotes the rate of the starting I-frame. We will assume a *rate matrix* $\mathbf{r}$ of size $|\mathcal{V}| * |\mathcal{V}|$ is computed *a priori* as input to the optimization algorithm. We will discuss how $\mathbf{r}$ is generated in our experiment in section 5.1.

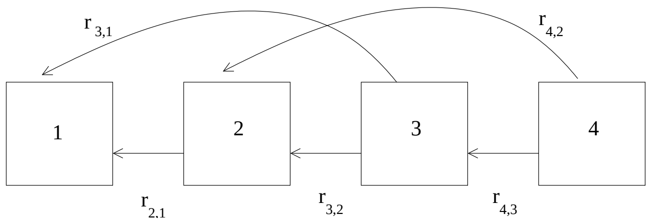

Fig. 1. Directed Acyclic Graph Source Model

3.2. Network Model

We assume the network provides a set of quality services (QoS), resulting in different packet loss rates. This can be achieved via actual network infrastructure support [1] (network-level), or end host induced such as applying different strengths of FEC (host-level).

We henceforth assume a QoS set $\mathcal{Q} = \{0, 1, \ldots Q\}$ for streaming media transport. Each frame F_i can select a QoS level q_i, $q_i \in \mathcal{Q}$, for packet loss protection. $q_i = 0$ denotes the case when F_i is not transmitted at all.

For given observable network condition, QoS level q_i and frame size $r_{i,j}$ will entail a frame delivery success probability $p(q_i, r_{i,j})$. $p()$ is dependent on $r_{i,j}$ because a large frame size will likely negatively impact the delivery success probability of the entire frame as more data is pushed through the network. While the operation and the optimality of our algorithm are independent of how $p(q_i, r_{i,j})$ is defined, we will illustrate an example how $p(.,.)$ can be evaluated in practice.

3.2.1. *Example Evaluation of* $p(q_i, r_{i,j})$

Suppose the transport layer provides 3 QoS levels using (n, k) Reed-Solomon code ($k = n, n-1, n-2$), mapping each data unit i at the application level to $q_i \in \mathcal{Q} = \{0, 1, 2, 3\}$. Given the raw packet loss rate of the network currently is α, we can evaluate the resulting packet loss rate given q_i, $\epsilon(q_i)$, as follows:

$$\epsilon(q_i) = \alpha \sum_{s=q_i-1}^{n-1} \binom{n-1}{s} \alpha^s (1-\alpha)^{n-1-s} \tag{3}$$

The frame delivery success probability $p(q_i, r_{i,j})$ depends on how many packets frame F_i needs for delivery. Hence:

$$p(q_i, r_{i,j}) = (1 - \epsilon(q_i))^{\left\lceil \frac{r_{i,j}}{MTU} \right\rceil} \tag{4}$$

where MTU is the maximum network transport unit.

3.2.2. *Network Resource Constraint*

Like any resource allocation problems, we impose a constraint on the amount of resource we can use, which in our case is the aggregate ability to protect the $|\mathcal{V}|$-frame sequence from network losses using QoS. Assume a QoS assignment q_i results in a cost of $c(q_i)$ per byte, then the constraint is:

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{\forall j | e_{i,j} \in \mathcal{E}} x_{i,j} c(q_i) r_{i,j} \leq R_t \tag{5}$$

In the case of network-level QoS, (5) represents a cost constraint, so that total cost to the user per $|\mathcal{V}|$-frame time does not exceed R_t units. In the case of host-level QoS, (5) represents a bit rate constraint, where constraint parameter R_t can be obtained using a congestion control algorithm like [7], so that the total output bytes for $|\mathcal{V}|$-frame time does not exceed R_t bytes.

3.3. Integer Programming Formulation

The objective function we selected is the expected number of correctly decoded frames at the decoder. Each frame F_i is correctly decoded iff F_i and all frames F_j's it depends on ($\forall j \preceq i$) are delivered drop-free. Mathematically, we write it as:

$$\max_{\{x_{i,j}\},\{q_i\}} \left\{ \sum_{i=1}^{|\mathcal{V}|} \prod_{\forall j \preceq i} \sum_{\forall k | e_{j,k} \in \mathcal{E}} x_{j,k}\, p(q_j, r_{j,k}) \right\} \tag{6}$$

The problem is then: given pre-computed rate matrix **r**, delivery success probability function $p(q_i, r_{i,j})$ and cost function $c(q_i)$, find variables $\{x_{i,j}\}$ and $\{q_i\}$ that maximize (6) while satisfying the integer constraint (1), the RF constraint (2) and the network resource constraint (5). This formally defined optimization is called the *RF / QoS selection problem*.

4. DYNAMIC PROGRAMMING SOLUTION

Given the RF / QoS selection problem is NP-hard (proof in section 7.1), we first present a pseudo-polynomial algorithm that solves the optimization problem optimally but in exponential time. We then discuss how a rounding technique can be used to trade off algorithm complexity with the quality of the resulting solution.

The optimization algorithm composes of two recursive functions, called $Sum(i, R)$ and $Prod(j, i, R)$, and are shown in Figure 2 and 3 respectively. $Sum(i, R)$ returns the maximum sum of products of sums in (6) for frames F_1 to F_i given R network resource units are available. $Prod(j, i, R)$ returns the inner product of sums term in (6) for F_j — probability that F_j is *decoded* correctly — given R network resource units are optimally distributed from F_1 to F_i. A call to $Sum(|\mathcal{V}|, R_t)$ will yield the optimal solution. We now examine $Sum(i, R)$ and $Prod(i, R)$ closely.

4.1. Dissecting $Sum(i, R)$

The recursive case (line 13-21) is essentially testing every combination of RF and QoS for F_i for the maximum sum. The result of this search is stored in the $[i, R]$ entry of the 3 dynamic programming (DP) tables, $DPSum[\ ]$, $DPQoS[\ ]$ and $DPInd[\ ]$ (line 22-24). DP tables are used so that if the same subproblem is called again, the already computed result can be simply returned (line 1-2). The two base cases (line 3-12) are the following: i) when the resource constraint is violated, in which case we return $-\infty$ to signal the violation; and, ii) when the root node (I-frame) is reached. Because root node has no RF to choose from, the search for optimal solution (line 6-10) is much simpler.

The complexity of $Sum(|\mathcal{V}|, R_t)$ is bounded by the time required to construct the DP table of size $|\mathcal{V}| * R$. To fill each entry, we call function $Sum(i, R)$ as shown in Figure 2, which has complexity $O(|\mathcal{E}||\mathcal{Q}|)$ to account for the two `for` loops from line 14-21 in the recursive case. Therefore we can conclude the complexity of $Sum(i, R)$ is $O(|\mathcal{V}||\mathcal{E}||\mathcal{Q}|R_t)$.

4.2. Dissecting $Prod(j, i, R)$

From line 16-17 of Figure 2, we see that $Prod(j, i, R)$ is called after $Sum(i, R)$ has been called, so we will assume entry $[i, R]$ of the DP tables are available during execution of $Prod(j, i, R)$.

```
function Sum(i, R)
if (DPSum[i, R] is filled)      // Note: DP case
{   return DPSum[i, R];   }
if (R < 0)                       // Note: base case 1
{   return −∞;   }
if (i = 1)                       // Note: base case 2
{   maxV := 0;
    for each q ∈ Q,
    {   if (R ≥ c(q)r_{1,1})
        {   maxV := max (maxV, p(q, r_{1,1}));   }
    }
    return maxV;
}
maxV := 0;                 // Note: recursive case
for each j such that e_{i,j} ∈ E,
{   for each q ∈ Q,
    {   temp = Sum(i − 1, R − c(q)r_{i,j});
        temp += p(q, r_{i,j}) Prod(j, i − 1, R − c(q)r_{i,j});
        if (temp > maxV)
        {   (maxV, maxQ, maxJ) := (temp, q, j);   }
    }
}
store maxV in DPSum[i, R];
store maxQ in DPQoS[i, R];
store maxJ in DPInd[i, R];
return maxV;
```

Fig. 2. Defining $Sum(i, R)$

```
function Prod(j, i, R)
if (R < 0)              // Note: base case 1
{   return 0;   }
if (j = i = 1)          // Note: base case 2
{   return DPSum[1, R];   }
q^o := DPQoS[i, R];
k^o := DPInd[i, R];
if (j < i)              // Note: recursive case
{   val := Prod(j, i − 1, R − c(q^o)r_{i,k^o});   }
else                    // Note: j = i
{   val := p(q^o, r_{i,k^o});
    val := val ∗ Prod(k^o, i − 1, R − c(q^o)r_{i,k^o});
}
return val;
```

Fig. 3. Defining $Prod(j, i, R)$

The recursive case has two sub-cases: i) when $j < i$, in which case we recurse on $Prod(j, i-1, .)$ given we know resource $c(q^o)r_{i,k^o}$ is optimally used for node i; and, ii) when $j = i$, in which case we know term i of the product term — $p(q^o, r_{i,k^o})$. The maximum product will be this times the recursive term $Prod(k^o, i-1, R - c(q^o)r_{i,k^o})$. The two base cases are similar to the two base cases for $Sum(i, R)$.

4.3. Trading off Complexity with Solution Quality

As previously derived, the complexity of $Sum(|\mathcal{V}|, R_t)$ is $O(|\mathcal{V}||\mathcal{E}||\mathcal{Q}|R_t)$, which is pseudo-polynomial[1]. Instead of solving the original RF / QoS selection problem instance I for optimal solution s, we solve a modified problem instance I' for solution s' with complexity reduced by a factor of K at the cost of decreasing solution quality. To accomplish that, we simply rewrite the network resource constraint by dividing and rounding up each rate term $r_{i,j}$ by factor K and dividing and rounding down the constraint parameter R_t by the same K. The new network constraint becomes:

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{j \,|\, e_{i,j} \in \mathcal{E}} x_{i,j} c(q_i) \left\lceil \frac{r_{i,j}}{K} \right\rceil \leq \left\lfloor \frac{R_t}{K} \right\rfloor \tag{7}$$

Using the same $Sum(i, R)$ and $Prod(j, i, R)$, the complexity of I' is now $O(|\mathcal{V}||\mathcal{E}||\mathcal{Q}|\frac{R_t}{K})$.

It can be easily shown (See section 7.2) that s' is feasible in I. Moreover, we can bound the performance difference between s' and s by first obtaining a super-optimal solution s'' in a new problem instance I'', where the network resource constraint is now:

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{j \,|\, e_{i,j} \in \mathcal{E}} x_{i,j} c(q_i) \left\lfloor \frac{r_{i,j}}{K} \right\rfloor \leq \left\lceil \frac{R_t}{K} \right\rceil \tag{8}$$

After obtaining optimal solution s'' to I'', we can bound our approximate solution s' from the optimal s in original problem instance I as follows:

$$\left| \text{obj}(s) - \text{obj}(s') \right| \leq \left| \text{obj}(s'') - \text{obj}(s') \right| \tag{9}$$

where $\text{obj}(s)$ is the objective function using solution s. The proof of this bound is in section 7.2.

5. EXPERIMENTAL RESULTS

5.1. Experimental Setup

To test the performance of the proposed optimization algorithm for the RF / QoS selection problem, we selected network simulator 2 (ns-2 [8]) as our testing environment. We constructed three independent paths from streaming source to client with three different loss rates to emulate three classes of network service. Class C has a packet loss rate of α, and class B and A employs Reed Solomon code of strength $(10, 9)$ and $(10, 8)$ respectively, resulting in smaller packet loss rates computed using (3).

The H.26L video software we use is version JM 4.2 and is accessible at [2]. The video sequence we selected for experimentation is the first 100 frames of the QCIF 176x144 `news` sequence, sub-sampled in time by 2 (i.e. we encoded every other frame). The quantization parameters are kept unchanged throughout at 31 and 30 for I-frames and P-frames respectively. We forced an I-frame into the sequence every 10 frames, meaning we optimize a group of 1 I-frame plus 9 P-frames at a time. We assume a playback speed of 10 frames per second at the client.

To generate the rate matrix $\mathbf{r}$, we executed the encoder while forcing all frames F_i's to select F_{i-t}'s for RFs. The resulting coding rates are entries $r_{i,i-t}$'s. We repeated this procedure for $t = 1, \ldots, 5$.

5.2. Numerical Results

For the experiment, we assume a raw packet loss rate $\alpha = 0.1$ and round off factor $K = 100$. We compare our optimization algorithm to an ad-hoc water-filling algorithm which uses exclusively previous frame $i-1$ for motion prediction for frame i, and spends network resources to increase protection for the earliest frames in the sequence as much as possible until network resource runs out. The results are shown in Figure 4 for a range of bandwidths. The metric is the percentage of correctly decoded frames. We see our proposed algorithm (line 1) constantly out-performs the ad-hoc scheme (line 2) by a wide margin over the tested range of bandwidth. This is particularly apparent when bandwidth is small, where our algorithm results in more than 12% correctly decoded frames over ad-hoc scheme.

[1] This essentially means the complexity looks polynomial but is not. In this case, because R_t is encoded in $\lceil \log_2 R_t \rceil$ bits as input, $O(R_t)$ is exponential in the size of the input parameters.

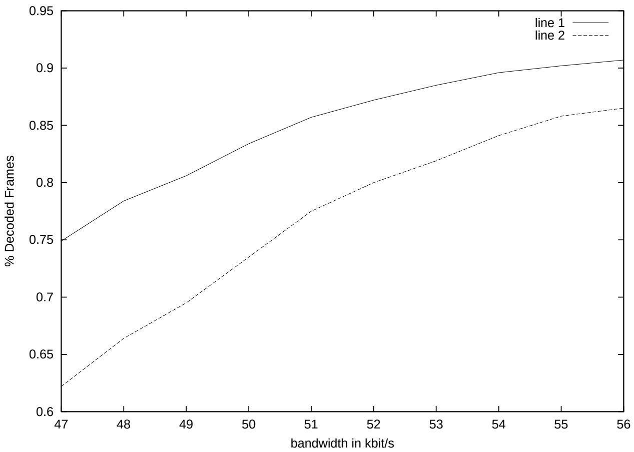

Fig. 4. Performance of Near-optimal vs. Ad-hoc Scheme

6. CONCLUSION

In this paper, we presented a novel video streaming optimization by jointly optimizing reference frame and network QoS selection. The solution is unique in that the complexity of the optimization algorithm is bounded and can be traded off with the quality of the obtained solution. Results show the obtained solution has as much as 12% improvement in correctly decoded frames over an ad-hoc reference frame / QoS selection scheme.

While the optimization algorithm was developed for a specific streaming scenario in this paper, We conjecture that the rounding technique — trading off complexity with quality of the obtained solution — can be applied to a much wider range of resource allocation problems in signal processing. This is an exciting avenue for future research.

7. APPENDIX

7.1. Proof of NP-Completeness

In this section, we prove that the RF / QoS selection problem is NP-hard by proving the corresponding binary decision problem — does there exist a solution such that the objective value is larger than some constant C? — is NP-complete. We accomplish that via a reduction from a well-known NP-complete problem, *Knapsack* problem (pg.247 [9]). For completeness sake, the Knapsack problem is repeated from [9] below:

> INSTANCE: Finite set $\mathcal{U}$, for each $u \in \mathcal{U}$ a size $s(u) \in Z^+$ and a value $v(u) \in Z^+$, and positive integers B and C.
> QUESTION: Is there a subset $\mathcal{U}' \subseteq \mathcal{U}$ such that $\sum_{u \in \mathcal{U}'} s(u) \leq B$ and $\sum_{u \in \mathcal{U}'} v(u) \geq C$?

The problem remains NP-complete if $v(u) = s(u)$.

For the reduction, we construct a corresponding RF / QoS selection problem instance as follows. We construct a $|\mathcal{U}| + 1$-frame sequence, each frame F_i, $i > 1$, having one possible RF, which is F_1. Each frame F_i has a rate $r_{i,1} = s(u_{i-1})$. We construct the QoS set to offer only two services: $\mathcal{Q} = \{0, 1\}$. The resulting rate matrix $\mathbf{r}$ is:

$$\mathbf{r} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ s(u_1) & 0 & \dots & 0 \\ s(u_2) & 0 & \dots & 0 \\ \vdots & & & \\ s(u_{|\mathcal{U}|}) & 0 & \dots & 0 \end{bmatrix} \tag{10}$$

The corresponding construction for a 5-frame sequence is shown in Figure 5. The resulting RF / QoS selection problem under this construction

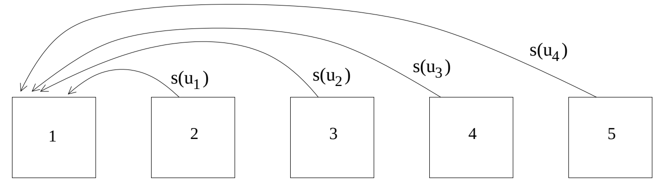

Fig. 5. Construction for NP-Completeness Proof

mathematically becomes:

$$\max_{\{x_{i,1}\}} \left\{ \sum_{i=2}^{|\mathcal{V}|} x_{i,1} \frac{r_{i,1}}{B} \right\} \quad \text{s.t.} \quad \sum_{i=2}^{|\mathcal{V}|} x_{i,1} r_{i,1} \leq B \tag{11}$$

where we set $p(1, r_{i,1})$ and R_t in (6) to be $r_{i,1}/B$ and B respectively. The corresponding binary decision problem is: does there exist a RF / QoS selection — $x_{i,1} \in \{0, 1\}$ — such that the objective value is ≥ 1?

It is clear from (11) that the binary decision problem of the constructed RF / QoS selection problem is equivalent to the original Knapsack problem instance when $s(u) = v(u)$. Hence the RF / QoS selection problem is as least as hard as the Knapsack problem. Therefore the RF / QoS selection problem is NP-hard.

7.2. Proof of Approximation Bound

We first show that $s' = \left(\{x'_{i,j}\}, \{q'_i\} \right)$ is feasible in I.

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{j \,|\, e_{i,j} \in \mathcal{E}} x'_{i,j} c(q'_i) \left\lceil \frac{r_{i,j}}{K} \right\rceil K \quad \leq \quad \left\lfloor \frac{R_t}{K} \right\rfloor K \tag{12}$$

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{j \,|\, e_{i,j} \in \mathcal{E}} x'_{i,j} c(q'_i) \frac{r_{i,j}}{K} K \quad \leq \quad \frac{R_t}{K} K \tag{13}$$

where (13) holds since $\frac{r_{i,j}}{K} \leq \left\lceil \frac{r_{i,j}}{K} \right\rceil$ and $\frac{R_t}{K} \geq \left\lfloor \frac{R_t}{K} \right\rfloor$. Using similar argument, one can show that s is feasible in I''. By optimality of s'' in I'', we have:

$$\text{obj}(s'') \geq \text{obj}(s) \tag{14}$$

Subtract obj(s') from both side and we get (9).

8. REFERENCES

[1] S. Blake et. al., "An architecture for differentiated services," December 1998, IETF RFC 2475.

[2] "The tml project web-page and archive," http://kbc.cs.tu-berlin.de/ stewe/vceg/.

[3] T. Wiegand, N. Farber, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," in *IEEE J. Select. Areas. Comm.*, June 2002, vol. 18, no.6, pp. 1050–1062.

[4] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," in *submitted to IEEE Trans. MM*, February 2001.

[5] Y.J. Liang et. al., "Low-latency video transmission over lossy packet networks using rate-distortion optimized reference picture selection," in *IEEE International Conference on Image Processing*, Rochester, NY, September 2002.

[6] G. Cheung and W. t. Tan, "Directed acyclic graph based source modeling for data unit selection of streaming media over qos networks,," in *IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 2002.

[7] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," in *IEEE/ACM Trans. Networking*, August 1999.

[8] "The network simulator ns-2," June 2001, release 2.1b8a, http://www.isi.edu/nsnam/ns/.

[9] M. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1979.