NEAR-OPTIMAL MULTIPATH STREAMING OF H.264 USING REFERENCE FRAME SELECTION

Gene Cheung

Hewlett-Packard Laboratories, Japan

ABSTRACT

New video coding standards such as H.264 offer the flexibility to select from a number of reference frames for motion-estimation for a given predicted frame. In this paper, we propose an optimization algorithm using dynamic programming that exploits this flexibility for multipath streaming — simultaneously streaming over two transmission paths with different bandwidths and loss rates. A rounding technique is employed to scale the complexity of the algorithm down at the cost of degrading solution quality. Results show significant streaming quality improvement over a conventional multiple description scheme.

1. INTRODUCTION

This paper is concerned with the problem of optimal transport of standard-compliant video stream over networks with multiple transmission paths for real-time playback. Using multiple paths means larger combined transmission rate in the case when each path is rate constrained[1]. A network may have multiple transmission paths, for example, if a wireless client can simultaneously communicate with two nearby base-stations. Because the reserved network resources and physical properties such as distance from the transmitting base-station for each link are different, they will very likely have different packet loss rate and bandwidth constraint. We focus on such case when two transmission paths with heterogenous characteristics are simultaneously available.

For the application, we consider the scenario where the application requires lowest network delay possible, to the extent that it cannot tolerate even one end-to-end packet retransmission. One reason can be that a small playback buffer is employed at the client side, together with the relatively large transmission delay of wireless links such as 3G cellular links [1], means that any retransmitted packet upon client request will miss its playback deadline and hence be rendered useless. In such case, the optimizing strategy can only select the correct transmission path for each frame, subject to the two paths' rate constraints.

The video coding standard we are focusing on is H.264 [2], a new standard that offers many coding flexibilities for better coding and streaming performance. One of these flexibilities is flexible motion-estimation support, where each P-frame can choose among a number of frames for motion-estimation. At the cost of coding efficiency, using a frame further in the past for motion-estimation 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 transmission path for optimal performance? After discussing related work in section 2, we formulate it as a formal optimization problem in section 3. We present an approximate algorithm in section 4. Results and Conclusion are presented in section 5 and 6, respectively.

[1] In some cases, using two transmission paths simultaneously decreases overall performance because of mutual signal interference. We assume here that the paths are orthogonal and therefore additive.

2. PREVIOUS WORK

H.264 [2] is a new video coding standard that has demonstratably 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-estimation, 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] [4]. In contrast, we jointly optimize streaming using both reference frame (RF) and transmission path (TP) selection.

The most related work is [5], where the authors consider using rate-distortion optimized reference frame selection together with path diversity for optimized streaming. To match the two-state Gilbert model considered in [5], an ad-hoc path selection scheme is used *after* reference frame selection is done. In contrast, we jointly optimize the selection of both the reference frame and the transmission path simultaneously. Doing so means rate constraints for the two paths can be considered during optimization — this is not considered in [5].

A related research topic is multiple description (MD) [6], where video is encoded into two (or more) "descriptions", and each description can be decoded independently of the other. For example, a MD encoder encodes even frames independently as stream 1, and odd frames independently as stream 0. Customarily, each description is sent over one of two TPs, with the assumption that at any one time, transmission errors typically occur in one or the other TP but not both. We differ in assumption in that simultaneous failure in both paths is probable — as in the case of cellular links — and is taken into account in the optimization. Also note that typical frame-level MD-encoded streams, such as the even and odd frames encoded streams described above, is simply a special case of the many possibilities our optimization algorithm considers.

Unlike many previous rate-distortion optimization algorithms [7] [4] [5] which rely on the use of Lagrange multipliers, our optimization is unique in that we use a rounding technique that trades off complexity with the quality of the obtained solution. This relieves us of the burden of finding a suitable Lagrange multiplier, which is non-trivial.

3. PROBLEM FORMULATION

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

0-7803-7750-8/03/$17.00 ©2003 IEEE

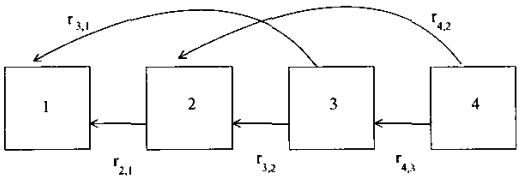

Fig. 1. Directed Acyclic Graph Source Model

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 [7]. Specifically, the streaming media is represented by a collection of frames, F_i's, $i \in \{1, \ldots, |\mathcal{V}|\}$. 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, and 0 otherwise:

$$x_{i,j} = \begin{cases} 1 & \text{if } F_i \text{ uses } F_j \text{ as RF} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

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

$$\sum_{j|e_{i,j} \in \mathcal{E}} x_{i,j} = 1 \qquad \forall i \in \mathcal{V} \quad (2)$$

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, for fixed quantization parameters, 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 integer $r_{i,j} \in \mathcal{I}$, 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.

3.2. Network Model

We assume the network provides two TPs with different packet loss rates. More specifically, we assume an independent and identically distributed (iid) loss model for each path. As a concrete example, consider the case where a wireless client uses two cellular links connected two different base-stations simultaneously. We henceforth assume each frame F_i can select a network path k, denoted by variable $q_i = k$, $k \in \{0, 1\}$.

For given selected TP_k and frame size $r_{i,j}$, it entails a frame delivery success probability $p_k(r_{i,j})$. $p_k(.)$'s are in general dependent on $r_{i,j}$ because a large frame size may negatively impact the delivery success probability of the entire frame as more data is pushed through the lossy path. Note, however, that the operation and the optimality of our algorithm are independent of how $p_k(.)$'s are defined.

3.2.1. Network Resource Constraints

We assume each path has a rate constraint due to limited network resources available. Mathematically, the constraints for the two paths are:

$$\begin{aligned} \sum_{i=1}^{|\mathcal{V}|} \sum_{j|e_{i,j} \in \mathcal{E}} x_{i,j} q_i r_{i,j} &\leq R_1^* \\ \sum_{i=1}^{|\mathcal{V}|} \sum_{j|e_{i,j} \in \mathcal{E}} x_{i,j} (1 - q_i) r_{i,j} &\leq R_0^* \end{aligned} \quad (3)$$

In practice, rate constraints are imposed by the network infrastructure — for cellular link, during session setup phase when physical resources are assigned.

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:

$$\max_{\{x_{i,j}\},\{q_i\}} \left\{ \sum_{i=1}^{|\mathcal{V}|} \prod_{j \preceq i} \sum_{k|e_{j,k} \in \mathcal{E}} x_{j,k}\, p_{q_j}(r_{j,k}) \right\} \quad (4)$$

The problem is then: given pre-computed rate matrix $\mathbf{r}$ and delivery success probability functions $p_k(r_{i,j})$'s, find variables $x_{i,j}$'s and q_i's that maximize (4) while satisfying the integer constraint (1), the RF constraint (2) and the network resource constraints (3). We formally denote this optimization the *RF / TP selection problem*.

4. DYNAMIC PROGRAMMING SOLUTION

A proof similar to the one in [8] can be easily constructed to show that the RF / TP selection problem is NP-hard. Given it is NP-hard, we first present a pseudo-polynomial algorithm that solves the optimization problem optimally but in exponential time. We then discuss how a rounding technique is used to trade off algorithm complexity with the quality of the resulting solution.

The optimization algorithm composes of two recursive functions, called $Sum(i, R_1, R_0)$ and $Prod(j, i, R_1, R_0)$ and are shown in Figure 2 and 3 respectively. $Sum(i, R_1, R_0)$ returns the maximum sum of products in (4) for frames F_1 up to F_i given rate R_1 in path 1 and rate R_0 in path 0 are available. $Prod(j, i, R_1, R_0)$ returns the maximum product term for F_j, given rate R_1 in path 1 and R_0 in path 0 are available for F_1 to F_i. A single call to $Sum(|\mathcal{V}|, R_1^*, R_0^*)$ will yield the optimal solution. We now examine $Sum(i, R_1, R_0)$ and $Prod(i, R_1, R_0)$ closely.

4.1. Dissecting $Sum(i, R_1, R_0)$

The recursive case (line 10-22) is essentially testing every combination of RF and path for F_i for the maximum sum. The result of this search is stored in the $[i, R_1, R_0]$ entry of the 3 dynamic programming (DP) tables, $DPsum[\,]$, $DPpath[\,]$ and $DPind[\,]$ (line 23-25). 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 3-9) are the following: i) when

```
function Sum(i, R_1, R_0)
1. if (DPsum[i, R_1, R_0] is filled)     // DP case
2. {  return DPsum[i, R_1, R_0];  }
3. if (R_1 < 0) || (R_0 < 0)             //base case 1
4. {  return -∞;  }
5. if (i = 1)                            //base case 2
6. {  maxV_1 := (R_1 ≥ r_{1,1})? p_1(r_{1,1}) : -∞;
7.    maxV_0 := (R_0 ≥ r_{1,1})? p_0(r_{1,1}) : -∞;
8.    return max{maxV_1, maxV_0};
9. }
10. maxV := -∞;
11. for each j s.t. e_{i,j} ∈ E,          //recursive case
12. {  maxV_1 := Sum(i - 1, R_1 - r_{i,j}, R_0);
13.    maxV_1 += p_1(r_{i,j}) Prod(j, i - 1, R_1 - r_{i,j}, R_0);
14.    maxV_0 := Sum(i - 1, R_1, R_0 - r_{i,j});
15.    maxV_0 += p_0(r_{i,j}) Prod(j, i - 1, R_1, R_0 - r_{i,j});
16.    if(maxV < max{maxV_1, maxV_0})
17.    { if(maxV_1 > maxV_0)
18.      {  (maxV, maxQ, maxJ) := (maxV_1, 1, j);  }
19.      else
20.      {  (maxV, maxQ, maxJ) := (maxV_0, 0, j);  }
21.    }
22. }
23. store maxV in DPsum[i, R_1, R_0];
24. store maxQ in DPpath[i, R_1, R_0];
25. store maxJ in DPind[i, R_1, R_0];
26. return maxV;
```

Fig. 2. Defining $Sum(i, R_1, R_0)$

```
function Prod(j, i, R_1, R_0)
1. if (R_1 < 0)||(R_0 < 0)    //base case 1
2. {  return 0;  }
3. if (j = i = 1)             //base case 2
4.    return DPsum[1, R_1, R_0];
5. q^o := DPpath[i, R_1, R_0];
6. k^o := DPind[i, R_1, R_0];
7. if (j < i)                 //recursive case
8. {  if(q^o = 1)
9.      val := Prod(j, i - 1, R_1 - r_{i,k^o}, R_0);
10.   else
11.     val := Prod(j, i - 1, R_1, R_0 - r_{i,k^o});
12. }
13. else                      //j = i
14. {  if(q^o = 1)
15.    {  val := p_1(r_{i,k^o});
16.     val := val * Prod(k^o, i - 1, R_1 - r_{i,k^o}, R_0);  }
17.   else
18.    {  val := p_0(r_{i,k^o});
19.     val := val * Prod(k^o, i - 1, R_1, R_0 - r_{i,k^o});  }
20. }
21. return val;
```

Fig. 3. Defining $Prod(j, i, R_1, R_0)$

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 6-9) is much simpler.

The complexity of $Sum(|\mathcal{V}|, R_1^*, R_0^*)$ is bounded by the time required to construct the DP table of size $|\mathcal{V}| * R_1 * R_0$. To fill each entry, we call function $Sum(i, R_1, R_0)$ as shown in Figure 2, which has complexity $O(|\mathcal{E}|)$ to account for the `for` loop from line 11-22 in the recursive case. Therefore we can conclude the complexity of $Sum(|\mathcal{V}|, R_1^*, R_0^*)$ is $O(|\mathcal{V}||\mathcal{E}|R_1^*R_0^*)$.

4.2. Dissecting $Prod(j, i, R_1, R_0)$

From line 12-15 of Figure 2, we see that $Prod(j, i, R_1, R_0)$ is called after $Sum(i, R_1, R_0)$ has been called, so we will assume entry $[i, R_1, R_0]$ of the DP tables are available during execution of $Prod(j, i, R_1, R_0)$.

The recursive case has two sub-cases: i) when $j < i$, in which case we recurse on $Prod(j, i-1, ., .)$ given we know the optimal resource r_{i,k^o} is used for node i; and, ii) when $j = i$, in which case we know term i of the product term, which is either $p_1(r_{i,k^o})$ or $p_0(r_{i,k^o})$. The maximum product will be this times the recursive term $Prod(k^o, i-1, R_1 - r_{i,k^o}, R_0)$ or $Prod(k^o, i-1, R_1, R_0 - r_{i,k^o})$. The two base cases are similar to the two base cases for $Sum(i, R_1, R_0)$.

4.3. Trading off Complexity with Solution Quality

As previously derived, the complexity of $Sum(|\mathcal{V}|, R_1^*, R_0^*)$ is $O(|\mathcal{V}||\mathcal{E}|R_1^*R_0^*)$, which is pseudo-polynomial[2]. Instead of solving the original RF / TP selection problem instance I for optimal solution s, we solve a modified problem instance I' for solution s' with complexity reduced by a factor K^2 at the cost of decreasing solution quality. To accomplish that, we simply rewrite the network resource constraints by dividing and rounding *up* each rate term $r_{i,j}$ by factor K and dividing and rounding *down* the constraint parameters R_1^*, R_0^* by the same K. The new network constraints become:

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{j|e_{i,j}\in\mathcal{E}} x_{i,j} q_i \left\lceil \frac{r_{i,j}}{K} \right\rceil \leq \left\lfloor \frac{R_1^*}{K} \right\rfloor \quad (5)$$

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{j|e_{i,j}\in\mathcal{E}} x_{i,j} (1-q_i) \left\lceil \frac{r_{i,j}}{K} \right\rceil \leq \left\lfloor \frac{R_0^*}{K} \right\rfloor \quad (6)$$

Using the same $Sum(i, R_1, R_0)$ and $Prod(j, i, R_1, R_0)$, the complexity of I' is now $O(|\mathcal{V}||\mathcal{E}||\mathcal{Q}|\frac{R_1^*}{K}\frac{R_0^*}{K})$.

It can be easily shown (See [8]) 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 instance I'', where the network resource constraints are now:

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{j|e_{i,j}\in\mathcal{E}} x_{i,j} q_i \left\lfloor \frac{r_{i,j}}{K} \right\rfloor \leq \left\lceil \frac{R_1^*}{K} \right\rceil \quad (7)$$

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{j|e_{i,j}\in\mathcal{E}} x_{i,j} (1-q_i) \left\lfloor \frac{r_{i,j}}{K} \right\rfloor \leq \left\lceil \frac{R_0^*}{K} \right\rceil \quad (8)$$

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

$$|\text{obj}(s) - \text{obj}(s')| \leq |\text{obj}(s'') - \text{obj}(s')| \quad (9)$$

where obj(s) is the objective function using solution s. The proof of this bound is similar to one in [8].

5. EXPERIMENTAL RESULTS

5.1. Experimental Setup

To test the performance of the proposed optimization algorithm for the RF / TP selection problem, we selected network simulator 2 (ns-2 [9]) as our testing environment. We constructed two

[2]This essentially means the complexity looks polynomial but is not. In this case, because R_1^*, R_0^* are encoded in $\lceil \log_2 R_1^* \rceil$, $\lceil \log_2 R_0^* \rceil$ bits as input, $O(R_1^*R_0^*)$ is exponential in the size of the input parameters.

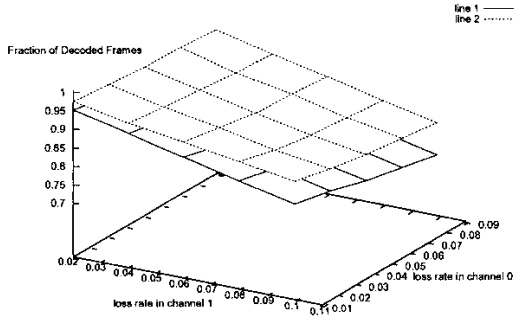

Fig. 4. Comparison for Different Loss Rates

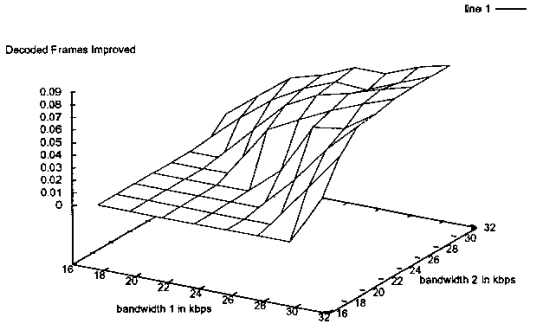

Fig. 5. Comparison for Different Bandwidth

independent paths from streaming source to client with two different loss rates. Path 1 has a frame delivery success probability $p_1(r_{i,j}) = 1 - \alpha_1$, and Path 0 has $p_0(r_{i,j}) = 1 - \alpha_0$.

The H.264 video software we use is version JM 4.2 [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 encoded 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 assumed a round off factor $K = 100$. We compared our optimization algorithm to a MD-encoded scheme which encoded even and odd frames independently into two streams for every 10 frames, i.e. frame F_i always used frame F_{i-2} for motion-estimation, with the exception of F_1, which used F_0, and F_0, which was an I-frame. Stream 1 uses path 1 and stream 0 uses path 0. Note that unlike our proposed algorithm, this MD-encoded scheme has the disadvantage that it is channel-blind.

Fixing the bandwidths of the two paths at 18kbps and 36kbps, the encoding rate of the MD-encoded streams, the performance of the two schemes are shown in Figure 4 for varying α_1 and α_0. The metric is the fraction of correctly decoded frames, where each frame F_i is correctly coded iff all frames F_j's it depends on, $j \preceq i$, are delivered drop-free. We see that the near-optimal scheme (line 2) consistently outperformed the MD-encoded scheme (line 1), and in poor network conditions, the near-optimal scheme decoded close to 10% more frames than the MD-encoded scheme.

For the second set of experiment, we fixed packet loss rates at $\alpha_1 = 0.1$ and $\alpha_0 = 0.05$, and we varied TPs' bandwidths. Figure 5 shows the fraction of correctly decoded frames of proposed scheme over MD-encoded scheme. By fixing quantization parameters, encoding rates of the two MD-encoded streams are fixed at 18kbps and 36kbps as in the previous experiment. We notice that as bandwidth increases in either path 0 or path 1, the proposed scheme is able to take advantage of the increased bandwidth by selecting earlier RFs to break error propagation and increase the number of correctly decoded frames.

6. CONCLUSION

In this paper, we consider optimized streaming for H.264 video over multiple transmission paths of different loss characteristics. In particular, we presented an optimization scheme that maximizes the expected number of correctly decoded frames at the receiver by selecting the near-optimal reference frame and transmission path for each predicted frame. The optimization is novel in the sense that unlike convention Lagrangian approaches, it uses a rounding technique instead to gracefully trade off complexity with the quality of the obtained solution.

7. REFERENCES

[1] G. Montenegro et. al., "Long thin networks," January 2000, IETF RFC 2757.

[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] 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.

[5] Y.J. Liang et. al., "Channel-adaptive video streaming using packet path diversity and rate-distortion optimized reference picture selection," in *IEEE Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, December 2002.

[6] J.G. Apostolopoulos, "Error-resilient video compression via multiple state streams," *Proc. International Workshop on Very Low Bitrate Video Coding (VLBV'99)*, pp. 168–171, October 1999.

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

[8] G. Cheung and Connie Chan, "Jointly optimal reference frame & quality of service selection for h.26l video coding over lossy networks," in *IEEE International Conference on Multimedia and Expo*, Baltimore, MD, July 2003.

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