OVERLAY MULTI-HOP FEC SCHEME FOR VIDEO STREAMING OVER PEER-TO-PEER NETWORKS[1]

Yufeng Shan,[1] *Ivan V. Bajić,*[2] *Shivkumar Kalyanaraman,*[1] and *John W. Woods*[1]

1/ ECSE Department, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
2/ ECE Department, University of Miami, Coral Gables, FL 33146, USA

ABSTRACT

Overlay networks offer promising capabilities for video streaming, due to their support for application-layer processing at the overlay forwarding nodes. In this paper we propose a novel Overlay Multi-hop FEC (OM-FEC) scheme that provides FEC encoding/decoding capabilities at intermediate nodes in an overlay path. Based on the current network conditions, the end-to-end overlay path is partitioned into segments, and appropriate FEC codes are applied over those segments. We evaluate our work in a real-world scenario and illustrate that the proposed OM-FEC can outperform a pure end-to-end strategy by 10-15 dB in terms of video PSNR.

1. INTRODUCTION

Peer-to-peer (P2P) architectures and overlay networks [1] are recently gaining attention in the context of video multicasting [2] – [3]. Performance characteristics of a peer-based overlay network are likely to be very different and highly variable compared to the traditional Internet because packets may cross the Internet several times. However, the massive diversity, i.e. multiple peer-based overlay paths harnessed could compensate for the performance variability of any one path. In addition, lightweight application-layer support at intermediate nodes can improve the *single* path performance. In this paper, we focus on the latter problem and propose a novel "Overlay Multi-hop FEC" (OM-FEC) scheme for video streaming over peer-based overlay networks. The OM-FEC scheme partitions the end-to-end overlay path into segments according to the error characteristic of the overlay path, and provides various amounts of FEC (through Reed-Solomon codes) over those segments. We assume a fixed constructed peer-based overlay path and focus on how to efficiently utilize it. We will henceforth use the term "overlay path" to mean the constructed path over a peer-to-peer network. We assume that we can always construct an overlay path with higher bandwidth than default Internet route by using P2P techniques such as Chord [4], to obtain a set of intermediate forwarding nodes as shown in Figure 1. In the figure, the dashed lines represent the virtual paths between overlay nodes and the solid line represents the default Internet path. Quantities B_i, P_i, and RTT_i represent, respectively, the bandwidth, loss rate, and round trip time of the *i*-th *virtual* link.

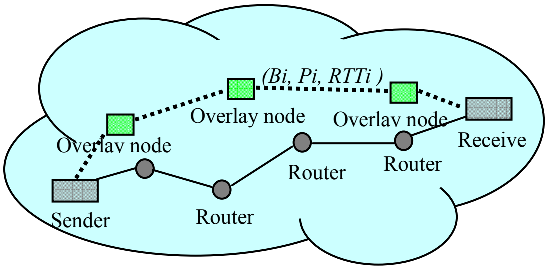

Figure 1: Streaming video using overlay network

The motivation behind OM-FEC is illustrated by the following example. In Table 1 we list a set of possible bandwidth and loss rates in a 6-hop overlay path.

Hop	1	2	3	4	5	6
B_i	300K	380K	550K	400K	600K	800K
P_i	0%	4%	3%	3.5%	1.5%	2%

Table 1: Possible bandwidth and loss rates of a 6-hop overlay path

In order to fully recover the lost packets, the end-to-end based FEC scheme would have to design its FEC based on the end-to-end available bandwidth 300Kbps and the total loss rate 14%. Thus, the good-put is reduced to 0.86×300 = 258Kbps. On the other hand, if a different amount of FEC is used at each hop, the end-to-end good-put can be engineered to be 300Kbps. Obviously, the hop-by-hop FEC scheme may induce more per-hop delay and use more

---

[1] This work was supported in part by the ARO grant number DAAD19-00-1-0559

0-7803-8554-3/04/$20.00 ©2004 IEEE.

computational power of the overlay nodes than necessary. To balance the delay and bandwidth efficiency with computational complexity, the proposed OM-FEC aims to maximize the video good-put over the overlay path, while minimizing the overall computational complexity at the intermediate nodes. This is achieved by partitioning the overlay path into sub-paths and performing FEC encoding/decoding at the end-nodes of these sub-paths.

The rest of our paper is organized as follows. In Section 2, we describe our protocol, rate allocation scheme, and algorithms for our proposed novel OM-FEC strategy. Next, we describe our Internet experiments and discuss the results in Section 3. Finally, in Section 4 we conclude our work and provide some ideas for possible extensions.

2. PROTOCOL OVERVIEW

Our protocol operates in two modes: (1) pure end-to-end mode; and (2) OM-FEC mode. The default mode is the end-to-end mode. In this mode, FEC is designed based on the end-to-end network characteristics, and the overlay nodes simply receive and forward data and FEC packets to the destination. If the network experiences congestion such that the end-to-end FEC scheme cannot recover the lost packets, the protocol transitions to the OM-FEC mode.

The video server sends out an active probe packet every Δt time units. Each overlay node measures the loss rate and *RTT* of its related virtual link in the overlay path using the probe packet. Thus obtained per-hop *RTT* and loss rate estimates are used to infer the TCP-friendly available bandwidth [6] of each virtual link. Using these bandwidth and loss rate estimates, the server runs a partitioning algorithm to calculate the optimal path partitioning consistent with the above FEC rate estimates, so that the overall computational complexity at intermediate hops is minimized without sacrificing the FEC-based resilience gains. Partitioning splits the overlay path into sub-paths, and FEC coding is employed over these sub-paths. Hence, only the boundary nodes between sub-paths are involved in FEC encoding/decoding. If the path partitioning algorithm produces a single sub-path (equivalent to the entire end-to-end overlay path), the system transitions back to the end-to-end mode. The decision made by the server is conveyed to every node by a small command packet from server, so each node knows what it should do after it receives the command packet. Our OM-FEC protocol is described in a) – c) below.

a) The server first calculates the parameters for the end-to-end based RS (n, k) code based on a given target loss probability for the video packets. Since the positions of the lost packets are known, the RS(n, k) code can correct up to $n-k$ erasures, so

$$P_{target} = \sum_{i=n-k+1}^{n} \binom{n}{i} P^{i} (1-P)^{n-i}, \qquad (1)$$

where P is the end-to-end packet loss rate. If k is fixed, n can be determined from (1) for a given P_{target}. According to [5], the viewing quality of MPEG-4 encoded video is acceptable at a loss rate of 10^{-5}, and good at a loss rate of 10^{-6}, so we choose $P_{target} = 10^{-6}$ in this paper. Let B be the estimated end-to-end TCP-friendly transmission rate, B_{data} be the rate of the encoded video, and B_{FEC} be the end-to-end transmission rate of FEC packets needed to provide the target loss probability. The total bandwidth B_{total} needed for transmitting both video and parity packets is

$$B_{total} = B_{data} + B_{FEC} = B_{data} + \frac{(n-k)}{k} B_{data} = \frac{n}{k} B_{data}. \quad (2)$$

b) If $B_{total} \le B$, the bandwidth needed for the video data and FEC is smaller than the available end-to-end bandwidth B of the overlay path. In this case FEC is deployed end-to-end. No intermediate overlay node is involved in the FEC encoding/decoding. The operating mode is therefore the end-to-end mode. An adaptive end-to-end rate allocation scheme is described in our prior work [8].

c) If $B_{total} > B$, the available end-to-end bandwidth B is not large enough for both the video data and end-to-end FEC overhead. If the current mode is end-to-end mode, then the protocol transitions to the OM-FEC mode, and the overlay path is partitioned into sub-paths according to the characteristics of each virtual link as shown in Figure 2. The figure shows an overlay path split into several sub-paths, so that the first J nodes are treated as as sub-path1, the next L nodes as sub-path2, etc. FEC scheme is deployed over each sub-path. The partition is dynamically determined by the OM-FEC partitioning algorithm described below.

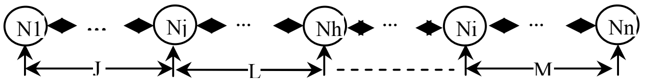

Figure 2: Partitioning of the overlay path into sub-paths.

Let N be the total number of overlay nodes in the overlay path, B_i be the estimated TCP-friendly transmission rate on a virtual link between the overlay nodes ($i-1$) and i, and $B_{FEC,(j-k)}$ for $j < k$ denote the FEC transmission rate between the overlay nodes j and k needed to satisfy the target loss probability. This rate is determined based on equation (1), but using the loss characteristics between the nodes j and k. The partitioning algorithm proceeds as follows.

```
Start = 0;          // begin calculation from the server
For (i = 1; i <= N; i++){
    // calculate the FEC bandwidth needed for the path
       between the start node and the i-th node
    Calculate $B_{FEC,(start\text{---}i)}$);
    // calculate the FEC bandwidth needed for the path
       between the start node and the (i+1)-th node
    Calculate $B_{FEC,(start\text{---}(i+1))}$;
    // determine if the i-th node is a suitable boundary
       node for the sub-path
    If ((min{$B_1$, …, $B_i$} $\geq$ ($B_{data}$ + $B_{FEC,(start\text{---}i)}$)) &&
       (min{$B_1$, …, $B_{(i+1)}$}< ($B_{data}$ + $B_{FEC,(start\text{---}(i+1))}$))){
       Declare the section from start node to i-th node as
       one sub-path;
       FEC is deployed over this sub-path, the FEC
       bandwidth allocated to this path is $B_{FEC,(start\text{---}i)}$;
       Start = i;  // start from the i-th node to partition
                      the rest of the path, the i-th node is
                      boundary node
    }
}
```

The server runs the above algorithm to partition the overlay path into sub-paths and deploys different amounts of FEC over different sub-paths. The decision is conveyed to all intermediate nodes by a small command packet. For each boundary node, the command packet contains a 3-byte field specifying the node ID, and the *n* and *k* parameters of the RS (*n*, *k*) code. The nodes whose ID is not listed in the command packet will simply forward all the packets they receive, without FEC coding/decoding. Thus, each node knows what it should do after it receives the command packet. The boundary nodes of these sub-paths are the only ones involved in the FEC encoding and decoding. Based on the OM-FEC strategy, the largest sub-path could include all the nodes of the overlay path (same as the end-to-end scheme), and the smallest sub-path could be one hop (i.e. hop-by-hop). In other words, OM-FEC is an adaptive strategy that tunes the architectural complexity between the extremes of end-to-end and hop-by-hop operation.

3. RESULTS

We implemented our protocol over the real Internet using the Planet-Lab infrastructure [7]. The implementation includes an overlay agent and the protocol itself. Our overlay agent can run at any Linux Planet-Lab node. The agent forwards video packet to the next node until the packet arrives at the destination. The experimental topology is shown in Figure 3 and the Planet-Lab nodes involved are listed in Table 2. We compare different schemes in terms of the PSNR of the video reconstructed at the receiver. We used the color *Foreman* sequence (QCIF resolution, 30 fps) encoded at 512 kbps using the H.263+ coder, with error-resilient option and Intra frame refresh at every second.

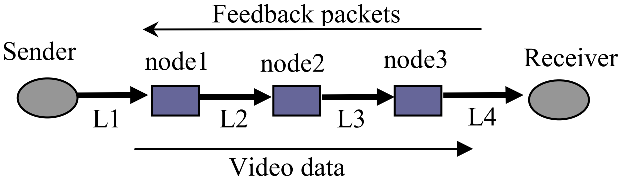

Figure 3: Simulation configuration.

Server	nima.eecs.berkeley.edu
Node1	planetlab1.flux.utah.edu
Node2	planetlab-1.cmcl.cs.cmu.edu
Node3	planetlab1.cs.cornell.edu
Receiver	video.testbed.ecse.rpi.edu

Table 2: Nodes involved in our experiments

Since there is virtually no congestion from UC Berkeley to RPI, packets are artificially dropped to simulate the congestion effect. The packet loss rate from Utah to CMU is set to 5%, other links are set to 1%. The available bandwidth from Utah to CMU is upper bounded to 550 kbps. In the above condition, the end-to-end scheme designs a FEC based on the 550 kbps bandwidth and total loss rate 8%. Our OM-FEC scheme identifies the bottleneck and partitions the overlay path into three sub-paths as follows: sub_path1 from Server to Node 1, sub_path2 from Node 1 to Node 2, and sub_path3 from Node2 to the receiver. Two intermediate nodes (Node1 and Node 2) are involved in FEC encoding/decoding. The OM-FEC designs FEC at the bottleneck for a bandwidth of 550 kbps and 5% loss rate. It can recover a larger number of lost packets than the end-to-end scheme, so the video quality is much higher as shown in Figure 4. The Y-component PSNR gains are on the order of 10-12 dB. For a given coded bit-stream, 3% percent random channel loss in a bit-stream can result in a much worse quality degradation than the reduction of the source coding rate by 3%, since the loss may happen in a very important part of the bit-stream. From Figures 4 and 5, we can see that all the I frames in the end-to-end scheme have losses and the errors propagate to the whole GOP, which results in poor video quality at the receiver.

In the second set of experiments we add one overlay node (Node 4: planet1.ecse.rpi.edu) to the path at last hop with 1% loss rate. In this case, for the end-to-end scheme, the FEC is designed based on the bandwidth of 550 kbps and loss rate 9%. Our OM-FEC scheme still partitions the overlay path into sub-paths as before, and the FEC at the bottleneck is still designed for the bandwidth of 550 kbps and 5% loss rate. Still, two nodes are involved in FEC encoding/decoding. The PSNR results are shown in Figure 5. The advantage of our OM-FEC scheme over the end-to-end scheme is now increased compared to Figure 4. Here, the PSNR gains are on the order of 14 dB. As the number of nodes involved in the transmission increases, our OM-FEC scheme performs dramatically better than the end-to-end scheme. Sample frames from these experiments are shown in Figure 6.

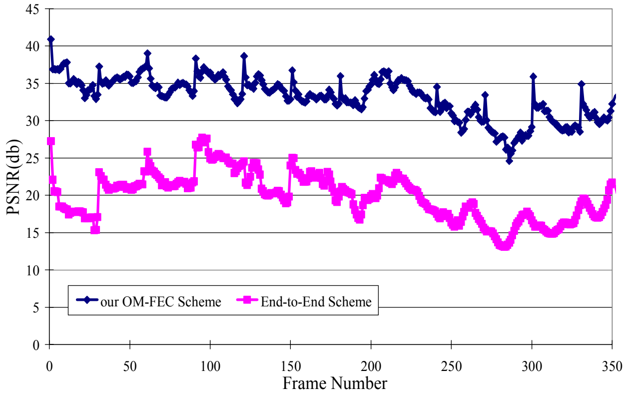

Figure 4: Y-component PSNR of our OM-FEC scheme vs. End-to-End scheme (4 virtual links).

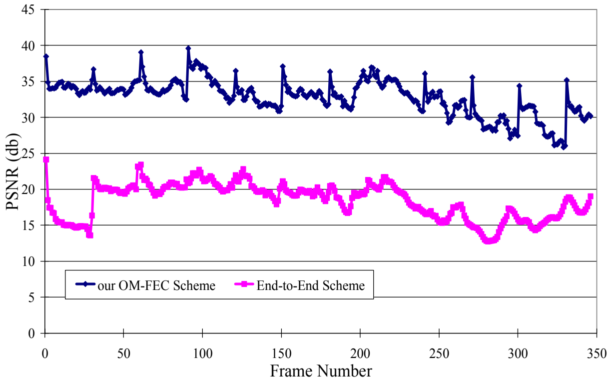

Figure 5: Y-component PSNR of our OM-FEC scheme vs. End-to-End scheme (5 virtual links).

4. CONCLUSIONS AND FUTURE WORK

We have proposed a novel Overlay Multi-hop FEC scheme for streaming video over peer-to-peer networks, which automatically adapts its architectural complexity between the extremes of pure end-to-end or pure hop-by-hop operation. We propose a rate allocation scheme and an OM-FEC scheme in order to achieve higher good-put of the constructed peer-based overlay transmission path. We have shown that video streaming using our approach outperforms the naive end-to-end FEC approach scheme without incurring per-hop complexity like in the hop-by-hop strategy. In our current work, we are incorporating ARQ techniques to combat overlay node failures, buffer management, and multi-path routing to build up an overall network service abstraction for video streaming and conferencing over peer-to-peer networks. We are also evaluating the extra computational burden of overlay nodes, and timing issues.

Figure 6: Sample frames from the video streaming experiments: OM-FEC (left) vs. End-to-End FEC (right) over 4 virtual links (top) and 5 virtual links (bottom).

5. REFERENCES

[1] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, “Resilient overlay networks,” *Proc. ACM SOSP’01*, pp. 131-145, Banff, Canada, October 2001.

[2] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, “Distributing Streaming Media Content Using Cooperative Networking,” Microsoft Technical Report MSR-TR-202-37, April 2002

[3] K. Yeo, B. S. Lee, and M. H. Er, “A Peering Architecture for Ubiquitous IP Multicast Streaming,” *ACM SIGOPS Operating Systems Review*, vol. 36, no. 3, pp 82-95, July 2002.

[4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications,”. *Proc. SIGCOMM’01,* pp 149-160, San Diego, CA, August 2001.

[5] S. Gringeri, R. Egorov, K. Shuaib, A. Lewis, and B. Basch, “Robust compression and transmission of MPEG-4 video,” *Proc. ACM MM 2000*, June 2000.

[6] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” ICSI Tech. Report TR-00-03, March 2000.

[7] http://www.planet-lab.org/

[8] Y. Shan and A. Zakhor “Cross Layer Techniques for Adaptive Video Streaming Over Wireless Networks," *Proc. IEEE ICME’02*, pp. 277-280. Lausanne, Switzerland, August 2002.