Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Signal Processing: *Image Communication* 20 (2005) 728–742

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.com/locate/image

Network-embedded FEC for optimum throughput of multicast packet video ☆

Mingquan Wu, Shirish S. Karande, Hayder Radha*

Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA

Received 30 April 2005; accepted 2 May 2005

Abstract

Forward error correction (FEC) schemes have been proposed and used successfully for multicasting realtime video content to groups of users. Under traditional IP multicast, application-level FEC can only be implemented on an end-to-end basis between the sender and the clients. Emerging overlay and peer-to-peer (p2p) networks open the door for new paradigms of *network FEC*. The deployment of FEC within these emerging networks has received very little attention (if any). In this paper, we analyze and optimize the impact of network-embedded FEC (NEF) in overlay and p2p multimedia multicast networks. Under NEF, we place FEC codecs in *selected* intermediate nodes of a multicast tree. The NEF codecs detect and recover lost packets within FEC blocks at earlier stages before these blocks arrive at deeper intermediate nodes or at the final leaf nodes. This approach significantly reduces the probability of receiving undecodable FEC blocks. In essence, the proposed NEF codecs work as signal regenerators in a communication system and can reconstruct most of the lost data packets without requiring retransmission. We develop an optimization algorithm for the placement of NEF codecs within random multicast trees. Based on extensive H.264 video simulations, we show that this approach provides significant improvements in video quality, both visually and in terms of PSNR values.

Keyword: FEC; Network coding; Multicast; Peer–peer; Video

1. Introduction

A variety of forward error correction (FEC) frameworks have been proposed and employed for packet loss recovery over the Internet, and especially for multicast applications. Traditional multicast video applications employ FEC on an end-to-end basis between the sender and the clients. However, the reliability and efficiency of end-to-end FEC-based packet video could suffer

☆Part of this work has been presented as: Mingquan Wu, Shirish Karande, and Hayder Radha, "Network-Embedded Channel Coding for Optimum Throughput of Multicast Packet Video", International Packet Video Workshop (PV), December 2004. Hayder Radha and Mingquan Wu, "Overlay and Peer-to-Peer Multimedia Multicast with Network-Embedded FEC", IEEE International Conference on Image Processing (ICIP), October 2004.

*Corresponding author. Tel.: +1 734 7633178.

E-mail addresses: wumingqu@egr.msu.edu (M. Wu), karandes@egr.msu.edu (S.S. Karande), radha@egr.msu.edu (H. Radha).

0923-5965/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2005.05.002

significantly over large video distribution networks. In this paper, we explore a new alternative for improving the reliability and efficiency (in terms of throughput) of packet video applications by optimum placement of few FEC codecs within large packet-video distribution networks. We develop an optimization algorithm for the placement of FEC codecs within selected nodes of random packet-video networks. We show that this approach provides significant improvements in video quality, both visually and in terms of PSNR values. These significant improvements are achieved while (a) maintaining the desired source-video coding rate, and (b) avoiding any source-video rate shaping or complex transcoding within the network. Hence, our proposed approach is motivated, in part, by the following:

- First, for many practical realtime video applications, the sender needs to transmit and adhere to a minimum source-video rate. This, for example, could represent the bitrate of the base layer of scalable video, or the rate of a minimum-acceptable quality non-scalable video stream.
- Second, for many applications, including ones with a large number of receivers, performing complex rate-shaping or transcoding operations may not be desirable or even feasible.
- Third, emerging and new network paradigms (e.g., [1–3,13–15]), such as overlay and peer-to-peer (p2p) systems, can facilitate the proposed framework for placing FEC codecs within realtime video distribution networks.

It is well known that the overall video quality is directly related to the effective video packet throughput that can be achieved with a given FEC channel-coding rate. However, for a given FEC coding rate (e.g., based on the popular Reed–Solomon FEC method), the packet loss ratio experienced by an end-to-end FEC-based video application could become very high when the number of nodes in the distribution tree increases. This naturally leads to a reduction in video packet throughput. One alternative for improving the reliability of end-to-end FEC solution is to lower the FEC coding rate (i.e., use more redundant packets and less video packets within an FEC block). However, this approach could lead to a significant reduction in the effective source-video rate.[1] Furthermore, and as highlighted above, the video application may need to adhere to a minimum source rate. This constraint could be expressed in terms of a rate value k/n, i.e., the sender needs to maintain a transmission rate of k video packets over an n-packet transmission periods. Consequently, in the context of an FEC channel coding, a minimum of k message (video) packets must be included in an n-packet FEC block.

In this work, under a given FEC (n,k) block constraint (i.e., a k/n coding rate constraint), we seek to achieve optimum video-packet throughput by the placement of FEC codecs within selected (optimum) locations (nodes) of the video distribution network. In particular, we analyze and optimize the impact of network-embedded FEC (NEF) within realtime packet video networks. We develop a recursively optimum scheme for the placement of a small number of NEF codecs within any randomly generated multicast video network of known (yet random) link loss rates. In essence, the proposed NEF codecs work as signal regenerators in a communication system, and hence, they can reconstruct the vast majority (and sometimes all) of the lost data packets without requiring retransmission and complex rate shaping and/or transcoding operations. Our theoretical analysis and simulation results show that a relatively small number of NEF codecs placed in (sub-)optimally selected intermediate nodes of a network can improve the throughput and overall reliability dramatically. This leads to the dramatic improvements in the overall video quality observed at the receiving nodes. Fig. 1 shows an example of the proposed NEF framework. It is worth noting that in the proposed NEF framework, the number of added NEF codecs is inherently minimized. This could be important

[1] Here, we are presenting arguments in the context of non-scalable packet video or the base layer of a scalable video stream. However, the same arguments and motivations are applicable to each layer of a scalable video solution over packet networks.

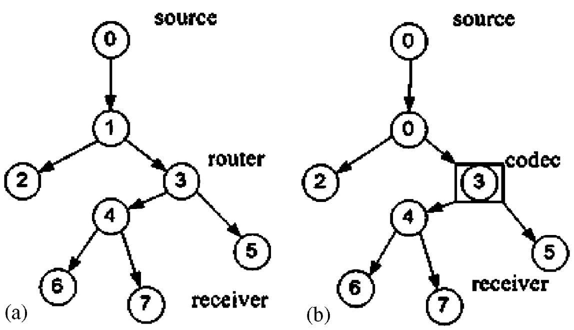


Fig. 1. (a) Under traditional realtime video multicast, intermediate nodes and routers do not perform any FEC functions. (b) A NEF codec in a video multicast tree can recover lost video and parity packets and transmit them downstream.

for applications that wish to minimize overall complexity and delay. For example, a particular service may tolerate a certain (maximum) number of NEF codecs between the sender and any receiver. This number can then be used by our NEF (sub-)optimization algorithm to place the desired FEC codecs.

As mentioned above, we envision the deployment of the proposed NEF framework in emerging networks such as overlay and p2p multimedia multicast systems (e.g., [1–3,13–15]). Overlay and p2p networks are becoming increasingly popular for the distribution of shared content over the Internet. Most of the studies conducted for these networks have focused on multicast tree building. Further, these studies assume that reliable transport and congestion control are performed by the underlying end-to-end transport protocol such as TCP. However, TCP favors reliable rather than on-time delivery. Under TCP, the source decreases the sending rate dramatically once congestion is detected, and this makes TCP not appropriate for realtime applications.[2] More importantly, the deployment of FEC within emerging networks for realtime multimedia applications has received very little attention (if any).

[2]For clients behind firewalls, there are several methods that allow non-TCP traffic to get through: (a) encapsulate non-TCP traffic in TCP at the last hop, (b) open specific ports on the firewall for non-TCP traffic, (c) set up proxy servers, etc.

Under the two types of networks considered here, "overlay" and p2p [1–3,13–15], multicast functions such as membership management and data replication are promoted to the application layer. Here, to distinguish it from a p2p network, an *overlay* network is equivalent to a *proxy-based* network[3] [3]. In a p2p multicast network, each node in the multicast tree can also be a multicast client (receiver). In a (proxy-based) overlay network, only the leaf nodes are clients. Within both networks, and at each intermediate node, data packets reach the application layer, and then get replicated and forwarded. Hence, in both cases (proxy-based or p2p), packet-loss recovery as an application level service can be placed in the intermediate nodes of the network.

The remainder of the paper is organized as follows. Section 2 presents an analytical model for rate-constrained video throughput using NEF within a multicast packet-video network. Section 3 describes and analyzes a recursive optimization NEF codec placement algorithm. Simulation results for reliability/throughput and for video quality measures are presented in Sections 4 and 5, respectively. Finally, we summarize the key conclusions of this work in Section 6.

2. Analysis of rate-constrained throughput using network-embedded FEC

Analyzing the impact of FEC on packet losses has been an active research problem that was addressed by previous efforts. In particular, previous studies analyzed the packet-loss model for FEC-enhanced multicast trees (e.g., [10,11]). These studies are based on the IP multicast model, in which intermediate nodes do not participate in FEC. Here, we study the packet-loss model of a multicast tree when FEC codecs are placed in the intermediate nodes of a tree. In our subsequent analysis and for the remainder of this paper we use the notations and definitions described in Table 1.

[3]Please note that both p2p and *proxy-based* networks are forms of *overlay* networks [3]. In this paper, we use the term *overlay networks* to refer to proxy-based networks.

Table 1
Important notations and definitions

T	A multicast tree with a root node r.
$\|T\|$	The size (in terms of the total number of nodes) of a multicast tree T.
T^c	A sub-tree rooted at some node $c \in T$ but does not include the node c.
T_l^c	The set of leaf nodes of T^c.
$\|T_l^c\|$	The total number of leaf nodes of the sub-tree T^c.
$P_v(i)$	Probability that node $v \in T$ receives exactly i packets.
$P_{v\|v-1}(i,j)$	Probability that node v receives i packets given that its parent $v-1$ sends j packets.
P	The packet loss probability between the link from $v-1$ to v.
(n,k)	The desired FEC block parameter pair (constraint) used by the system. n is the FEC block size, and k is the number of message (video) packets.
RS(n,k)	Reed–Solomon code with k video packets and $n-k$ parity packets.

We assume a binomial distribution for the packet losses.[4] For node v, if its parent $v-1$ sends j packets, the probability that it receives i packets is

$$P_{v|v-1}(i,j) = \binom{j}{i}(1-p)^i p^{j-i}. \quad (1)$$

When computing the probability $P_v(i)$ that a node v receives exactly i packets, we need to consider two cases; first, we consider the case when the parent node $v-1$ has no codec; second, we consider the case when the parent node $v-1$ has a NEF codec. If node v's parent does not have a codec, the probability that node v receives i packets is

$$P_v(i) = \sum_{j=i}^{n} P_{v-1}(j)P_{v|v-1}(i,j). \quad (2)$$

Note that $P_{v|v-1}(i,j) = 0 \ \forall j < i$. In other words, node v can receive i packets only when its parent $v-1$ sends at least i packets. For the root node (r) of the tree, we define

$$P_r(i) = \begin{cases} 0 & 0 \leqslant i \leqslant n-1, \\ 1 & i = n. \end{cases} \quad (3)$$

Eq. (2) is a recursive function, and hence with the initial condition from (3), we can calculate the probability $P_v(i)$, for any node v in the multicast tree, that it receives exactly i packets. When a node has a codec for a $R(n,k)$ block, and if that node receives less than k packets and cannot decode the FEC block, it will just forward the received packets as usual; if it receives k or more packets, the node can decode the block and reconstruct the original data. It can also reproduce the lost parity packets. In fact, a codec can produce more or less than $n-k$ parity packets if desired; however, in this paper, we assume that the NEF codecs reconstruct the original data and reproduce the lost parity packets using the same $R(n,k)$ code. These packets are then multicasted downstream. (The design of NEF codecs with an adaptive FEC erasure codes is a problem that we are currently pursuing, and it is beyond the scope of this paper.)

A node that has a NEF codec and which receives $k \leqslant j \leqslant n$ packets will send n packets. If v is the immediate child of a codec, the probability that it receives i packets becomes

$$P'_v(i) = \begin{cases} \sum_{j=k}^{n} P_{v-1}(j)P_{v|v-1}(i,n) & k \leqslant i \leqslant n, \\ \sum_{j=k}^{n} P_{v-1}(j)P_{v|v-1}(i,n) & \\ \quad + \sum_{j=k}^{n} P_{v-1}(j)P_{v|v-1}(i,j) & 0 \leqslant i \leqslant k-1. \end{cases} \quad (4)$$

Once a node c is assigned a NEF codec, the probability $P_v(i)$ for all $v \in T^c$ will change and need to be recomputed. We use (4) to calculate

[4]It is well known that losses over the Internet are bursty in nature (i.e., exhibit memory), and hence such losses do not necessarily follow a binomial distribution. Due to its simplicity though, many studies (including this paper) assumes a binomial distribution [9,10]. Evaluation of the performance of NEF under burst losses is part of our ongoing work.

$P_v(i)$ for the immediate children of the codec. For nodes that are not immediate children of a codec, the calculation of $P_v(i)$ is the same as Eq. (2).

Here we use P_v^{dec} to represent the probability that node v can decode a RS(n,k) block:

$$P_v^{\text{dec}} = P_v(i \geqslant k) = \sum_{i=k}^{n} P_v(i). \tag{5}$$

We define the average decodable probability of a tree T for p2p and proxy-based overlay networks, respectively, as

$$P_{\text{avg}}^{\text{dec}} = \frac{\sum_{v \in T-r} P_v^{\text{dec}}}{|T| - 1} \tag{6}$$

and

$$P_{\text{avg-leaf}}^{\text{dec}} = \frac{\sum_{v \in T_l^r} P_v^{\text{dec}}}{|T_l^r|} \tag{7}$$

If we use $r_{\text{d}}(v)$ to represent the number of received data packets (not including the parity packets received) of a FEC block at node v, then,

$$E[r_{\text{d}}(v)] = \sum_{i=k}^{n} kP_v(i) + \frac{k}{n}\sum_{i=0}^{k-1} iP_v(i). \tag{8}$$

Here we assume that for a RS(n,k) block, if a node receives i packets, on average only $(k/n)i$ are data packets. For a p2p and overly networks, we define the data throughput as

$$g = \frac{\sum_{v \in T-r} E[r_{\text{d}}(v)]}{(|T| - 1)k} \tag{9}$$

and

$$g_{\text{leaf}} = \frac{\sum_{v \in T_l^r} E[r_{\text{d}}(v)]}{|T_l^r| k}. \tag{10}$$

3. Optimum placement of network embedded FEC under a rate constraint

In this section, we develop a mechanism for placing NEF codecs within a given network topology. In a large topology, identifying the optimum locations for the NEF codecs is not a trivial task. One objective is to place codecs in the intermediate nodes of a topology to maximize the average throughput. Assuming that the loss rate for each link in the topology and the number of codecs to be placed are known beforehand,[5] the problem is similar to (but different from) the well-known P-median problem [4,7]. A P-median problem is to find P locations in the network to place facilities in order to minimize the overall cost for servicing all of the nodes. Generally, in a P-median problem, the cost to serve a node is determined by the *weight* at the node and the distance between the node and its nearest available facility. The P-median cost has nothing to do with other facilities placed in the network. As we have seen in the previous section, in order to calculate the decodable probability and throughput, we need to know the locations of the codecs that have been placed on that path, not just the immediate codec that serves the node.

As the throughput at a node in a NEF network is impacted by all the codecs placed along the path from that node to the source (root), the dynamic programming approaches that have been used in previous network-placement problems (e.g., [7]) cannot be used to solve the NEF codec placement problem. Thus, we use a greedy algorithm to place m codecs in the multicast tree.

The greedy algorithm finds the best location for the first codec, then the next best location for the second one, and so on. Once a node is selected, an FEC codec is added to regenerate any lost data or parity packets. Let $T^c \subset T$ be the subtree rooted at node $c \in T$ not including c. If c is set as a "codec node", only those nodes $v \in T^c$ will benefit from this selection; meanwhile, the "codec node" c itself will not be affected. For nodes $v' \in T - T^c$, everything remains unchanged. Let $E[r_{\text{d}}(v)]$ and $E'[r_{\text{d}}(v)]$ denote the average received packets for node $v \in T^c$ before and after node c is set as a codec node, respectively. We need to find $c \in T$ that maximizes the following:

$$\max_{c \in T} \left[\sum_{v \in T^c} (E'[r_{\text{d}}(v)] - E[r_{\text{d}}(v)]) \right]. \tag{11}$$

[5] Assuming that the multicast tree is relatively stable; each node can estimate the loss rate on the branch that connects its parents to itself. The loss rate and the connectivity information can be sent to a centralized place (e.g., the source) regularly. In one of our ongoing work, we have developed a distributed codec placement algorithm to cope with the dynamics of the network.

Table 2
Average throughput: comparison between optimal and greedy algorithm

Num of codecs	$p=3\%$		$p=4\%$		$p=5\%$	
	opt %	greedy %	opt %	greedy %	opt %	greedy %
2	98.5	98.4	93.9	91.9	87.8	87.8
3	99.1	99.1	95.8	95.4	90.4	90.4

A similar optimization objective function can be expressed for proxy-based overlay networks, except here the summation takes place over the leaf nodes only. Under the proposed greedy algorithm, we use an exhaustive search to find the best place for the first codec, after we find the optimum $c \in T$ node, we place the codec at that node. We use the same method to place the next codec; this process continues until all of the m codecs are placed.

The proposed greedy algorithm does not guarantee a global optimum solution for the placement of the m FEC codecs. Nevertheless, its performance has been very close to the global optimum. Table 2 shows the performance (in terms of throughput) resulting from the placement of $m=2$ and 3 FEC codecs (within 100-node multicast trees) based on the greedy algorithm, and compares these numbers with the throughput of the actual optimum placement under three (average) packet-loss ratios (p) over the multicast trees' links. (More details on the simulations are presented in the next section.) It is clear from the table that the greedy algorithm provides an excellent set of (sub-)optimum solutions in all 6 cases covered in this example.

4. Throughput analysis and simulation results

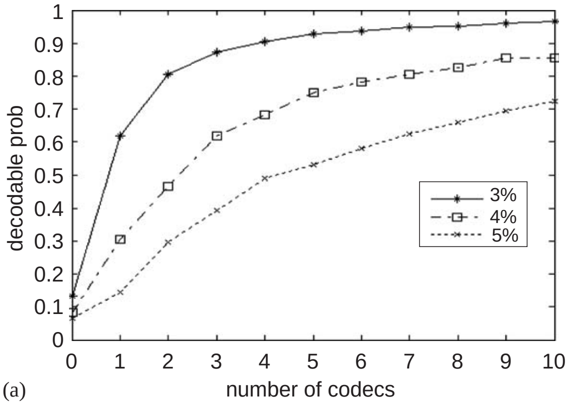
The throughput performance analysis presented above was applied to several random tree topologies. We use the popular Georgia Tech gt-itm [17] network topology generator to produce a set of 10 100-node transit-stub graphs. (Analysis and simulations with trees of larger sizes were also conducted. Here, we focus on the 100-node tree cases for brevity.) For each graph, we use Dijkstra's shortest path first (SPF) algorithm to produce a tree rooted at a randomly selected node. We used the greedy algorithm described in the previous subsection to place the NEF codecs in the multicast tree. The number of codecs was increased from 0 to 10. After each codec is placed, we calculate the improvement on average decodable probability and throughput. In addition to applying the above performance analysis on the 10 100-node trees, we used the ns2 [5] with some modifications for the support of the proposed NEF codecs in intermediate nodes. We modified the simulator to allow packets to reach the UDP and application layers. We have implemented a FEC UDP agent and a FEC application in the simulator. The analysis and simulation results were virtually identical. Below, and due to space limitations, we only present the analysis results.

As mentioned above, in a p2p overlay multicast network, nodes in the multicast tree are also end users, which often are placed at the edge of the Internet. Each hop in the overlay network often consists of several underlying physical hops. This implies that the loss rate of each hop could be higher than the loss rate of a backbone link in an IP multicast model. There have been many studies on Internet packet losses [8,12,16]. Generally, the average packet loss rate differs dramatically depending on the application (e.g., unicast versus multicast), the time when these measurements were performed and the packet transmission rate at the source. Here, we show results when the loss rate[6]

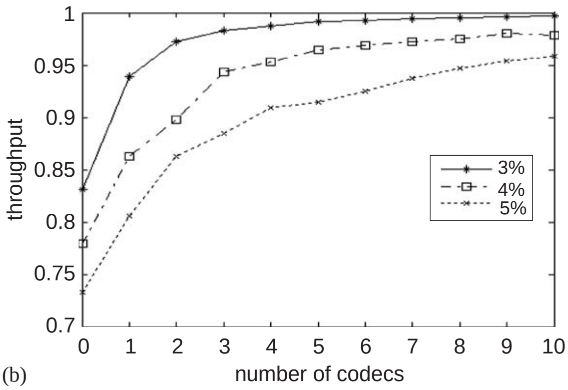
[6]These packet loss rates may be considered high for the current state of the Internet. However, it is important to note that multicast applications tend to have higher loss rates than the loss rates of unicast applications (in fact, it could be significantly higher). Furthermore, NEF could be employed among nodes at "the edge" of the Internet (e.g., p2p or proxy nodes) where packet losses may be high. Apart from these considerations, the impact of NEF will still be significant even under lower packet loss rates; since in this case, an end-to-end

per link is set to 3%, 4%, and 5%. We studied the performance improvement under each of these loss rates for a variety of RS codes. In this section, we present the results for RS(255, 223), which is a popular FEC code that has both hardware as well as software implementations. (The channel-coding rate[7] for RS(255, 223) is 87.5%).

The average FEC block decodable probability and data throughput for each tree were evaluated. The results are the averages over all of the 10 random trees that were analyzed. Fig. 2(a) shows the average decodable probability (over all nodes in a p2p tree) when the loss rates are set to 3%, 4% and 5%. (Similar results where obtained for proxy-based overlay networks. These results are not shown in this section for brevity. Video simulations for both cases are shown in the next section.) When no codecs are added, the FEC block decodable probabilities are very low for all three loss rates. For example, if the link loss ratio is 3%, the average decodable probability is just 18.6%. As the codec number increases, we see a dramatic increase in the decodable probability. It can be observed that a relatively small number of codecs can increase the decodable probability significantly. For a 3% per-link loss rate, the first codec increase the decodable probability from 18.6% to 76%; the first 3 codecs increase the decodable probability to above 95%. When the number of codecs increases to 10, the decodable probability reaches 99.9%; this implies that we can use NEF to achieve a very high level of reliability while using a very high (i.e., efficient) channel-coding rate. The results for the throughput are shown in Fig. 2(b). For an average p2p node, when no codecs are added, the throughput is about 85% with per-link loss rate of 3%. The first codec raises the throughput to over 95%. With only 3 NEF codecs, the throughput increases to 99%. For a typical video application, reducing the effective packet losses from 15% (85% throughput) to less



(a)



(b)

Fig. 2. (a) Average decodable probability over all nodes. (b) Average data packets throughput over all nodes.

(*footnote continued*)
application would use higher FEC coding rates, which lead to lower decodable probabilities. And therefore, NEF could help in improving the decodable probability and overall throughput as demonstrated in this paper.

[7]As we emphasized earlier, we target to achieve an optimum video-packet throughput solution under a given coding rate constraint. The above coding rate could represent one possible constraint value. Nevertheless, this channel coding rate may be high for some of the loss rates that are evaluated in this paper. However, it is important to note that the main conclusions of our study are valid regardless of the specific RS codes and the packet loss rates used in our simulations. In particular, the proposed NEF framework can be used in one of two ways. Under one approach, a *given* RS code is already being used (on an end-to-end basis) prior to adding any NEF codecs. In this case, NEF can significantly improve the overall throughput as shown extensively by our analysis and simulations in this paper. Under another approach, a reliable communication infrastructure is already in place. This reliable infrastructure would be normally based on using very conservative (low) FEC rates (i.e., much lower than the effective end-to-end channel capacity). In this case, NEF can be used to significantly improve the efficiency of the RS codes by increasing its rate while maintaining the same level of reliability provided by the original infrastructure. In this paper, we focused on the first scenario to illustrate the benefits of the proposed NEF-based framework in the context of rate-constraint video applications.

than 1% (higher than 99% throughput) will naturally have dramatic improvements in the decoded video quality, both in terms of PSNR and visual perception.

Fig. 3(a) shows the block decodable probability and packet throughput averaged only over the leaf nodes. The leaf nodes represent the set of nodes that receive the source data after the maximum number of relays and thus receive the data with more losses than the overall average. It can be observed that in the absence of a codec less than 20% packet blocks are received without any losses. However, from Fig. 3(a) and (b) it can be observed that introduction of embedded FEC can provide remarkable improvement in performance even for the leaf nodes.

As we eluded above, under high losses, traditional end-to-end FEC could resort to a significantly lower FEC coding rate (to lower the packet losses and achieve high reliability). However, this reduces the effective source video rate significantly. In this case, NEF could be used to maintain the high reliability performance while increasing the FEC rate significantly (i.e., increasing the effective source video bitrate). Either way, NEF provides salient and dramatic improvements in the delivery of realtime video over p2p and overlay networks. Thus in the next section we provide H.264 based video simulations which will further substantiate benefits of NEF codecs.

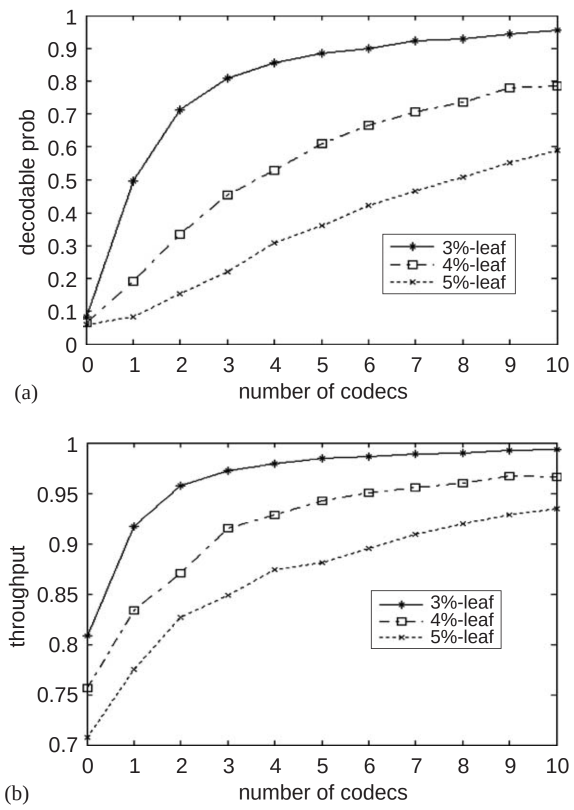


Fig. 3. (a) Average decodable probability over leaf nodes. (b) Average data packets throughput over leaf nodes.

4.1. Necessity of the greedy algorithm

One reason that we need a greedy algorithm is that we want to limit the number of codecs in a multicast session even if every node is willing to act as codec. Encoding and decoding are computation expensive operations, too many codecs may cause longer delay penalty and may transmit too many unnecessary packets into the network. In the above section, we have observed that when we embedded 10 codecs in a one-hundred node network, the maximum number of codecs per source-to-sink path is only two.

The other reason that we need the greedy algorithm is that not all possible codec arrangements necessarily lead to significant improvement in reliability and/or a near-optimal performance. We choose a proxy-based scenario to underline this argument. The results in this section are based on two arbitrarily chosen trees (for the sake of discussion we refer to these trees as "tree1" and "tree2") from the set of random trees considered in this paper.

It can be observed in Fig. 4(a)–(b) that placing NEF codecs in an arbitrary manner does not necessarily lead to an increase in reliability. It should be appreciated that trivial placements (e.g. embedding a NEF codec on a leaf node) have been excluded in Fig. 4(a)–(b). The design of our greedy algorithm is such that single NEF codec embedding is always optimal. However, even for multiple codec embeddings the placement given by the greedy algorithm is almost equal (if not equal) to the optimal solution. For 2 NEF codec placement for tree2 the solution given by the greedy algorithm was equivalent to the optimal solution

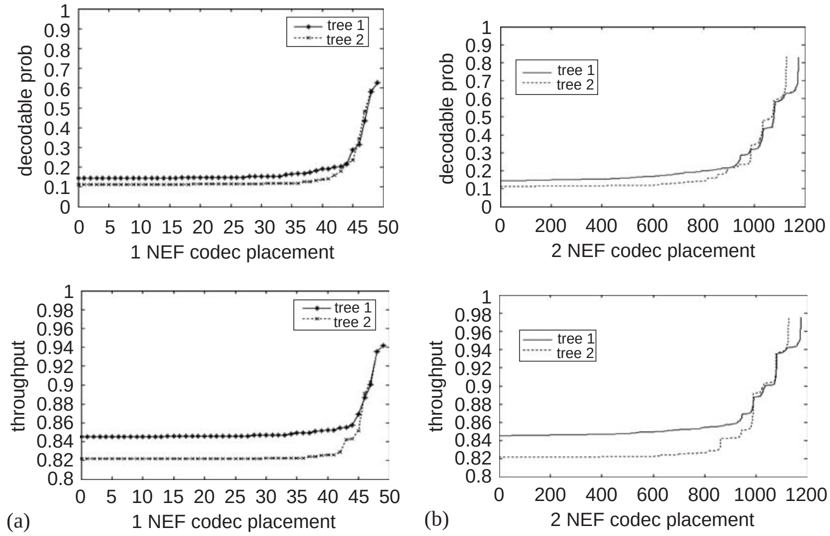


Fig. 4. The performance in terms of decodable probability and throughput for different NEF codec placements for two arbitrarily chosen topologies. The placements are ranked in a increasing order of efficiency. (a) Embedding a single NEF codec. (b) Embedding two NEF codecs.

and for tree1 in the sorted list of placements the greedy solution was just 1 index below the optimal solution. Furthermore, if all the considered solutions are assumed to be equally likely in a random NEF placement then:

- For a single NEF codec placement:
  - The decodable probability of a solution given by greedy algorithm is 62.57% for tree1 and 58.92% for tree2 as compared to the decodable probability of 18.54% for tree1 and 14.54% for tree2 obtained by averaging over all random placements.
  - The throughput of a solution given by greedy algorithm is 94.16% for tree1 and 93.57% for tree2 as compared to the throughput of 85.33% for tree1 and 82.93% for tree2 obtained by averaging over all random placements.
- For a placement of two NEF codecs:
  - The decodable probability of a solution given by greedy algorithm is 79.82% for tree1 and 83.39% for tree2, the results for the optimal solution are 83.25% for tree1 and 83.39% for tree2, respectively, as compared to the decodable probability of 22.87% for tree1 and 17.98% for tree2 obtained by averaging over all random placements.
  - The throughput of a solution given by greedy algorithm is 97.06% for tree1 and 97.46% for tree2, the results for the optimal solution are 97.57% for tree1 and 97.46% for tree2, respectively, as compared to the throughput of 86.14% for tree1 and 83.72% for tree2 obtained by averaging over all random placements.

Thus from the above discussion it should be clearly evident that an efficient NEF codec placement algorithm is necessary and the greedy algorithm that we have proposed is indeed such an algorithm. The proposed algorithm not only is near optimal but also improves the decodable probability and throughput by a significant margin when compared with a random placement. However, it should be noted that random algorithms where, all the solutions considered here are not equally likely, might provide better performance. However, as the solution given by the greedy algorithm is almost always equal to the optimal solution, we believe that it would be difficult to design a simple enough random algorithm that can provide performance comparable to the greedy algorithm.

5. Video simulations

Discussions in previous sections have concentrated on exhibiting the packet throughput improvements that can be achieved using NEF codecs. At this stage, it is necessary to clearly establish the advantage of using NEF in terms of the quality of video service available at the receivers. We use the emerging H.264/JVT [6] video standard for all the video simulations in this section. All the test sequences considered in this section have a ''cif'' frame size and are encoded at a frequency of 30 frames/sec. We use a constant quantization size of $QP = 16$ for all the video sequences. The results presented in this section are a subset of the examples we considered and thus it should be stated that the above choice of QP, frame frequency and frame size do not compromise on the generalness of the conclusions derived on the basis of the video analysis presented here. Unless specified no special error-resilience features are activated during the source encoding. Specifically, we do not turn on the ''RD-optimization in presence of losses'' feature in the JVT standard unless specified. Lastly, the encoded streams are made up of video packets of size 512 bytes each.

5.1. Performance for diverse video sequences

The test sequences that we consider are *mobile*, *stefan* and *carphone*. It should be mentioned that all the simulations are actually based on sequences that are multiple repetitions of the original test sequences. Our choice of test sequences represents a diverse set of source features, e.g. stefan is a sports sequence, carphone has comparatively high temporal correlation and mobile has multiobject motion. Thus based on the above-described encoding parameters, a lossless streams of mobile, stefan, carphone exhibit psnr values of 33.97, 35.47, and 37.75 dB, respectively.

Fig. 5 shows the video quality[8] of the above three sequences for a link loss probability of 0.03.

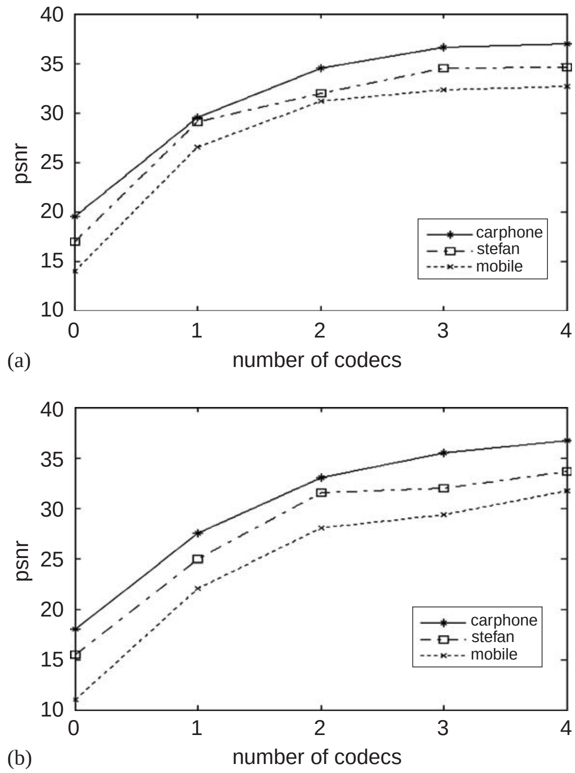


Fig. 5. (a) Average psnr over all nodes. (b) Average psnr over leaf nodes. The simulations are based on a link loss probability is 0.03.

It can be seen that significant improvement in video quality can be achieved by embedding codecs in the network. It can be observed that just adding 1–2 codecs can improve the performance by over 10 db. The distortion levels continue to decrease as more codecs are introduced in the network. However, the distortion

[8] Although it might be important to measure the performance guaranteed to the worst case receivers, in this section we present the results in terms of *average* video quality because our greedy algorithm optimizes the average video quality. The nature of our greedy algorithm is such that the embedded NEF codecs improve the *overall* quality of all receivers. Furthermore, as the number of codecs is increased, the variance in video quality provided to different users keeps reducing.

It should be noted that a greedy algorithm that tries to improve the performance only for the worst node might provide local optima and thus provide a worse solution than the proposed greedy algorithm when multiple codecs have to be embedded. Experimental evaluation of this phenomenon is not conducted as yet and could be a part of some future work.

levels do not decrease by equal amounts on addition of a new codec. Thus in a practical scenario the number of codecs can be determined on the basis of the required quality of service guarantees. For example, in our simulations, although 4 codecs are unable to guarantee a 100% lossless transmission, the distortion level is reduced to within 1 dB of lossless when averaged over all nodes. As the leaf nodes represent peers that receive the data after maximum impairments, naturally the distortion observed by these receivers is more than the overall average. However, compared with the quality of service received by leaf nodes in the absence of codecs, the improvement can still be termed as dramatic.

From Fig. 5 it can be observed that the stefan sequence represents in some way the averaged behavior of mobile and carphone. Thus from this point onwards we primarily concentrate all the analysis on the stefan sequence.

5.2. Dependence on link-loss probability

As done in previous sections, we evaluate the NEF performance under different link-loss probabilities, by evaluating the distortion in quality as seen by the receivers. Only results for the stefan sequence are presented. In order to describe the insight provided by Fig. 6 we consider the following notation. Let $Q(m,p)$ be the average video quality seen by the end user for a given link-loss probability p, when m codecs are embedded in the network. Thus let $\Delta Q(m,p) = Q(m,p - Q(m-1,p)$ represent the incremental utility of the mth codec. It can be observed that ΔQ is a monotonically decreasing function with respect to m. In other words, the quality improvement on account of adding a new codec decreases as the number of already embedded codecs increase. Moreover, it should be noted from Fig. 6 that the distortion decrement that can be achieved by a small number of codecs is more when the link-loss probability is low, e.g. $\Delta Q(1,0.03) > \Delta Q(1,0.04)$ for all nodes. However, as n increases, the distortion improvement for low link-loss probability saturates, but for high link-loss probabilities performance improvement on account of adding a new codec can still be substantial. e.g. $\Delta Q(4,0.03) < \Delta Q(4,0.04)$. Thus it can be concluded that utility

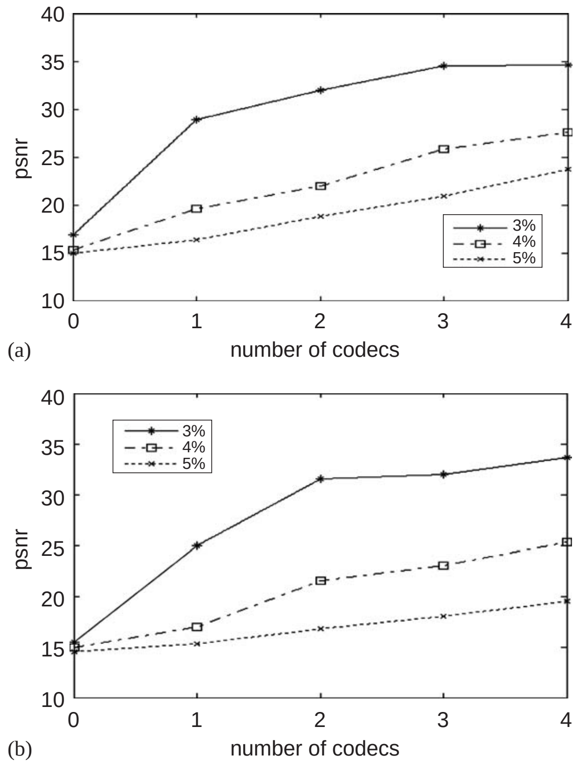


Fig. 6. (a) Average psnr over all nodes. (b) Average psnr over all leaf nodes. All simulations are based on stefan test sequence.

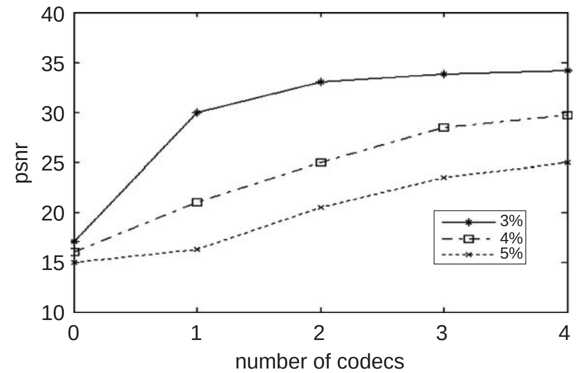


Fig. 7. Average psnr over all nodes for a robust source encoding of the Stefan sequence.

of NEF even in very poor channel conditions can also be significant. However, as the channel conditions deteriorate the number of codecs have

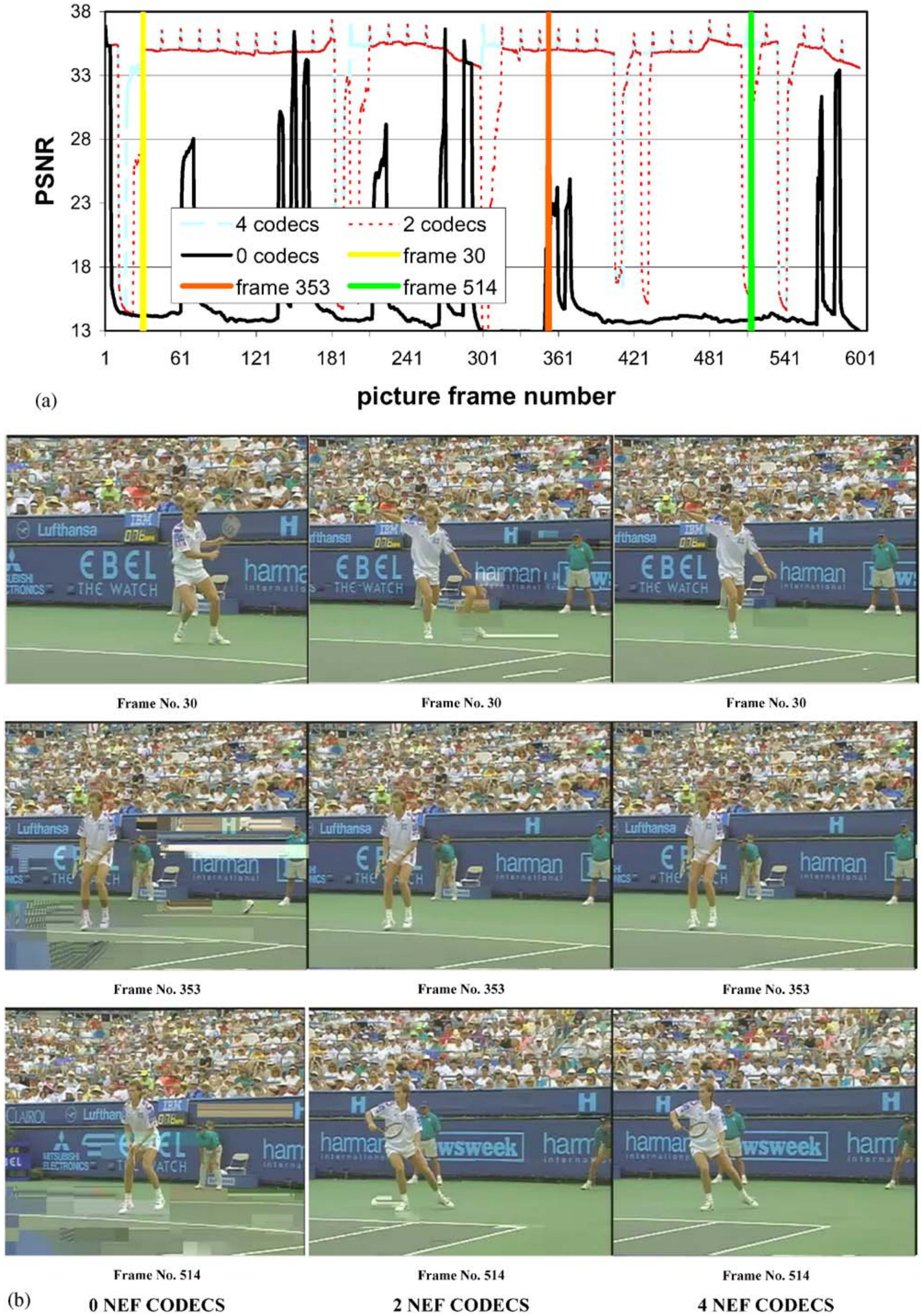


Fig. 8. (a) Picture framewise PSNR for a stefan sequence with robust source encoding. (b) A subjective comparison of distortion level with 4 codecs (right column), 2 codecs (center column) and without any codecs (left column).

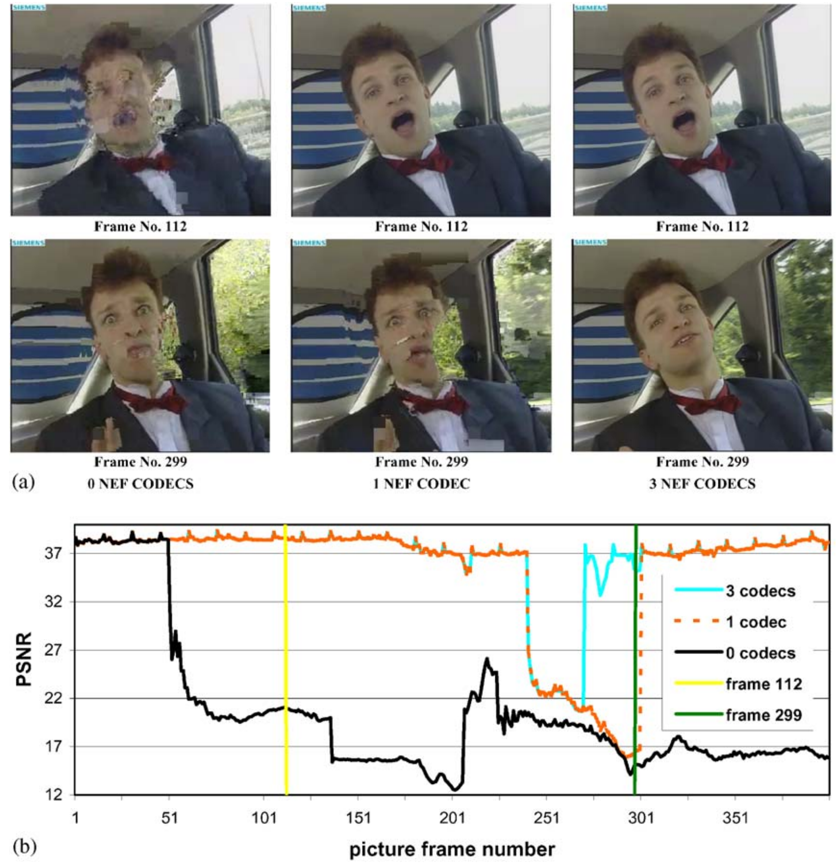


Fig. 9. (a) Picture framewise PSNR for a carphone sequence. (b) A subjective comparison of distortion level with 3 codecs (right column), 1 codecs (center column) and with out any codecs (left column).

to be increased in order to maintain the quality of service.

5.3. Dependence on source encoding robustness

It could be argued that if a robust enough source code is used, then the advantage of using a NEF codec might not be significant. Our results show that this is not at all the case. We use a robust encoding of the stefan sequence to present our results. We use the H.264 RD optimization feature to optimize the source encoding for a loss rate of 30%. The rest of the simulation setup is maintained as before. Observing Fig. 7 it can be seen that NEF codecs continue to provide improvement over 10–15 db s. However by comparing Fig. 7 with Fig. 6 it can be observed that the improvements are not as dramatic as in the case of a non-robust source encoding.

5.4. Subjective video evaluation

In this section we present results based on stefan and carpone sequence to facilitate a thorough subjective evaluation of the improvement in video quality on account of embedding NEF codecs. As the relative improvement of the NEF scheme for the robust source encoding is less than that for non-robust encoding, we intentionally first chose the "robust" stefan sequence to present our subjective results. This represents a minimal distortion reduction (i.e., minimum advantage) that can be achieved by using NEF. Fig. 8(a) is temporal video quality plot. When losses are incurred, the distortion in a video sequence suddenly increases, this is represented by the downward spikes in Fig. 8(a). It can be seen that as the number of codecs are increased the frequency of these downwards spikes are decreased. In fact, the overall video quality in absence of video codecs is so low that we have upward spikes of quality improvements on account of intra-refresh or some other error resilience feature, instead of downward spikes.

Fig. 8(b) compares actual video frames to facilitate a clear subjective comparison. It is important to note that the choice of frames is not at all biased in favor of NEF (in fact it is slightly biased against). Again it can be clearly seen that the block distortion level or loss of motion prediction is very high in the absence of codecs. As error concealment features use data from previous frames to reduce block distortion the primary artifact causing loss in video quality is not block distortion but in fact jerkiness. The video was observed to be very discontinuous in absence of codecs and this discontinuity decreases as codecs are increased. As psnr is primarily determined by block distortions, it should be appreciated that improvement in video quality in terms of the viewing experience is even greater than as predicted by the psnr plots.

Results presented in Fig. 9(a) and (b) based on the carphone sequence further substantiate the above deductions.

6. Conclusion

In this paper, we explored a new approach for improving the reliability and throughput of packet video by optimum placement of FEC codecs within large packet video distribution networks. We developed an optimization algorithm for the placement of FEC codecs within selected nodes of random packet-video networks. We also demonstrated that this approach provides significant improvements in video quality, both visually and in terms of PSNR values. Our proposed approach has been motivated by (a) the need to adhere to a minimum source-video rate constraint that many practical video application require, (b) the need for avoiding complex transcoding operations that may not be feasible over large video networks, and (c) exploiting new and merging peer-to-peer and overlay networks that facilitate the proposed framework for placing FEC codecs within realtime video distribution networks.

References

[1] M. Castro, P. Druschel, A.-M. Kermarrec, A. Rowstron, Scribe: a large scale and decentralized application-level multicast infrastructure, in: IEEE JSAC 20 (8) (October 2002).

[2] M. Castro, M.B. Jones, A.-M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, A. Wolman, An evaluation of scalable application-level multicast built using peer-to-peer overlays, Infocom 2003, San Francisco, CA, April 2003.

[3] Y.-H. Chu, S.G. Rao, H. Zhang, A case for end system multicast, in: Proceedings of ACM Sigmetrics, June 2000.

[4] M.S. Daskin, Network and Discrete Location: Models, Algorithms, and Application, Wiley, New York, 1995.

[5] http://www.isi.edu/nsnam/ns/

[6] ISO/IEC JTC 1/SC29/WG11 and ITU-T SG16 Q.6, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), Doc. JVT-G050, March 2003.

[7] B. Li, M.J. Golin, G.F. Italiano, X. Deng, K. Sohraby, On the optimal placement of web proxies in the internet, in: IEEE INFOCOM'99, March 1999, pp. 1282–1290.

[8] D. Loguinov, H. Radha, Large-scale experimental study of internet performance using video traffic, ACM SIGCOMM Comput. Commun. Rev. (CCR) 32 (1) (January 2002).

[9] M. Mosko, J.J. Garcia-Luna-Aceves, An analysis of packet loss correlation in FEC-enhanced multicast trees, ICNP 2000, 14–17 November 2000, Osaka, Japan, pp. 151–161.

[10] J. Nonnenmacher, E. Biersack, D. Towsley, Parity-based loss recovery for reliable multicast transmission, IEEE Trans. Networking (August 1998) 349–361.

[11] J. Nonnenmacher, L. Martin, J. Matthias, E. Biersack, G. Carle, How bad is reliable multicast without local recovery? Proceedings of the First IEEE INFOCOM'98, vol. 3, March 1998, pp. 972–991.

[12] V. Paxson, Measurement and analysis of end-to-end internet dynamics, Ph.D. Dissertation, University of California, Berkeley, 1997.

[13] D. Pendarakis, S. Shi, D. Verma, M. Waldvogel, ALMI: an application level multicast infrastructure, Proceedings

of the Third USNIX Symposium on Internet Technologies and Systems, March 2001.
[14] A. Rowstron, P. Druschel, Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems, in: Proceedings of Middleware, November 2001.
[15] I. Stoica, R. Morris, D. Karger, F. Kaashoek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup service for internet applications, in: Proceedings of ACM SIGCOMM, August 2001.
[16] M. Yajnik, S. Moon, J. Kurose, D. Townsley, Measurement and modeling of the temporal dependence in packet loss, IEEE INFOCOM, March 1999.
[17] E.W. Zegura, GT-ITM: Georgia tech internetwork topology models (software), http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm.tar.gz, 1996.