Content-Aware Distortion-Fair Video Streaming in Networks

Ying Li Zhu Li Mung Chiang A. Robert Calderbank

Abstract—Internet is experiencing an explosive growth of video traffic. Given the limited network bandwidth resources, how to provide Internet users with good video playback quality is a key problem. For video clips competing bandwidth, we propose an approach of Content-Aware distortion-Fair (CAF) video delivery scheme, which is assumed to be aware of the characteristics of video frames and ensures max-min distortion fair sharing among video flows. Different from bandwidth fair sharing, CAF targets video playback quality fairness for the reason that users care about video quality rather than bandwidth. The proposed CAF approach does not need an analytical rate-distortion function which is difficult to estimate, but instead, it uses the explicit distortion of every frame which is induced by frame drop. Our CAF approach is fast and practical with content-aware cooperation. Experimental results show that the proposed approach yields better quality of service when the network is congested compared with the approach not rate-distortion optimized, and it makes competing video clips help each other to get fair playback quality.

I. INTRODUCTION

The recent advances in bandwidth and storage capabilities in Internet and Internet devices have ushered an unprecedented growth of video traffic over Internet. According to some recent estimate, more than 70% bits carried in Internet are video. Popular video repository solutions like Youtube are driving up this video traffic demand on the network everyday. Unlike web and email traffic, video is characterized by its large bandwidth requirement and stringent quality of service (QoS) parameters, especially delays. This presents a challenge to network engineering different from early days of mostly text voice dominated traffic.

Video traffic is different from data traffic by the fact that it can be made “elastic”, i.e., certain packets can be dropped if the network is congested, with controllable decoding quality degradation. Recent advances in video coding [13] allows rate-distortion tradeoffs in video SNR quality, display size and temporal resolution. A cross-layer optimization in networking with content characteristics considered can bear fruits in end-to-end delivered content quality and network efficiency. A good summary and analysis of cross-layer optimization work can be found in [6].

Pictures (GoP) as in Fig. 2. If a B frame is dropped, e.g. \( f_{22} \) is dropped, the operational pair will only consists of bits for \( f_{22} \), and distortion incurred:

\[
R(f_k) = b(f_k), \quad D(f_k) = d(f_k, f_{k-p})
\]

where \( k - p \) is the frame index of the last frame not dropped, \( b(f_k) \) is the number of bits for frame \( k \), \( d(f_j, f_k) \) is the frame drop distortion metric, which will be discussed in more detail.

In this example, if \( f_{21} \) is not dropped, \( p = 1 \), and \( D(f_{22}) = d(f_{22}, f_{21}) \). If \( f_{21} \) is also dropped, \( p = 2 \). If a P-frame is...
dropped, the frames after the P frame and before the next I frame is considered lost,

\[ R(f_k) = \sum_{j=k}^{k+t-1} b(f_j), \quad D(f_k) = \sum_{j=k}^{k+t-1} d(f_{k-1}, f_j) \]

where \( k + t \) is the next I frame index. The I-frames are independently coded, and if an I-frame is dropped, the subsequent frames within the same GoP become useless, and the associated operating point will have distortion of frame drops for the whole GoP as,

\[ R(f_k) = \sum_{j=k}^{k+N-1} b(f_j), \quad D(f_k) = \sum_{j=k}^{k+N-1} d(f_{k-1}, f_j) \]

where \( N \) is the GoP size. This computation assumes that the player display the last available frame \( f_{k-1} \), until the next frame, \( f_{k+N} \) is available.

There are many different distortion metrics [11] to measure the impact of frame loss on visual perception. Simply computing MSE distance is not good, since even a one-pixel shift of frame results in a large distortion while perceptually it is negligible. In this work, we adopt the distortion based on principle component analysis (PCA), the same metric as in [11]. For a frame distortion metric reflecting the subjective quality of an image perception, we use the Euclidean distance in the principal component space of the frames. Video frames are scaled into smaller sizes (e.g., 8x6, 12x9 or 16x12), which is to reduce noise and local variance such that the frame distance evaluation is performed at a spatial resolution scale that probably better matches human perception.

Imagine at certain bottleneck link, the video queue needs to be reduced. Armed with the knowledge of R-D operational pair for each frame, we can have a quick sorting of the distortion \( D(f_k) \), or distortion/rate ratio, \( D(f_k)/R(f_k) \), and come up with optimized solutions for either QoS fair, or total utility maximization. In a network with multiple hops and multiple flows, we develop a utility fair solution in following sections.

### III. Content-Aware Distortion-Fair Video Streaming

#### A. Max-min Utility Fair Share

Consider a communication network with \( L \) logical links, each with a fixed capacity of \( c_l \) bps, and \( S \) source-destination pairs, each transmitting at a rate of \( x_s \) bps. Each source \( s \) emits one flow, using a fixed set \( L(s) \) of links, and has a utility function \( U_s(x_s) \). Each link \( l \) is shared by a set \( S(l) \) of sources. Since the publication of the seminal paper [9] in 1998, the framework of network utility maximization (NUM) has found many applications in network resource allocation algorithms and the design of protocol stacks [6].

NUM, in its basic version is the problem of maximizing the total utility over the source rates \( x \), subject to linear flow constraints \( \sum_{s \in S(l)} x_s \leq c_l \), for every link \( l \). The basic NUM is for the social efficiency in utility, but efficiency may not mean fairness.

A problem of fairness in QoS is of our interest here. Utility max-min fair bandwidth sharing is the following problem of maximizing the minimum utility, over the source rates \( x \), subject to linear flow constraints for all links,

\[
\text{maximize } \min_{s \in S(l)} U_s(x_s) \\
\text{subject to } \sum_{s \in S(l)} x_s \leq c_l, \quad \forall l \\
\text{x} \geq 0.
\]

#### B. Utility and Distortion

The utility function here, for the case of video delivery with loss, can be minus end-to-end distortion. Hence max-min utility fair is equivalent to min-max distortion fair.

#### C. Content-Aware Distortion Fair Share

There exists fairness in terms of the bandwidth sharing. Why do we need distortion fairness, not the bandwidth fairness? Users care about utility (QoS) rather than bandwidth, but bandwidth fair sharing may not always give us utility fairness because the same utility may need different bandwidth according to different content. Consider two users, each sending a video clip over a common link. When they compete for the bandwidth, by the bandwidth fair sharing, each user gets half of the link capacity. But they may not have the equal happiness. Say, User 1 is transmitting a video with a lot of motion, like commercial, User 2 is transmitting a video with less motion, like a man sitting there reading news. The rate-distortion is illustrated in Fig. 3. If the link is congested and some frames of the video should be dropped, perceptually User 1 is unhappy about the dropping because the content is sensitive to the loss, but User 2 can be happy even if a lot of frames are dropped because the content is not loss sensitive. With distortion fairness, User 1 can get more bandwidth than User 2, but perceptually both users are equally happy. Hence the sharing should be content-aware and distortion fair.

#### D. Difference from Related Work

There are existing works on max-min QoS fair sharing, such as [5] [14]. Authors of [5] [14] assume an explicit utility function of rate and the algorithms are based on the utility function, not really based on content. The work in [5] [14] are...
not specifically for video and they do not take into account the special characteristics of video as discussed in Section II. For video, the utility function is time-varying and it is hard to estimate the utility function accurately and on the fly. In addition, the distributed algorithm in [14] may need a lot of iterations and the rate may fluctuate dramatically such that the video quality fluctuates and the perceptual quality may be bad. As Section II discusses, the video delivery over networks is actually to shape the rate-distortion to a good operational point fitting the network condition. Different from [5] [14], we do not assume any explicit utility function, but instead we use the utility of every frame which can be easily and explicitly calculated. In addition, our scheme is fast in terms of the number of iterations.

Our scheme is per-flow performance guaranteed. Different from Intserv [4] architecture which also offers per-flow performance guarantees, our scheme has content awareness which is signaled over links. Our work is also different from DiffServ [3] which manages resources at the granularity of traffic classes.

A lot of other works utilizes content-awareness in other scenarios, such as in P2P streaming [2], in wireless networks [10], in content delivery networks [1], and in image authentication [17]. In [10], the content awareness is particularly a metric of the motion characteristics of different scenes. The content awareness in [2] is similar to our work, but in [2] the main focus is how to reduce the delay in wireless setting, but our work is to reduce the distortion fairly in a wired setting. In [1], the content awareness lies in the knowledge of the distortion level of dropping frames for all users, such that the total traffic through the link is less than the link capacity. The same threshold of dropping frames makes users experience a min-max fair distortion.

In a general network, the end-to-end path of a user may consist of several links, each of which may be shared by different users. If each link decides a common threshold of dropping frames for its users, since links are heterogeneous, on its end-to-end path a user (flow) may see the links each with a different threshold of dropping frames, and naturally the distortion level of dropping frames for a user should be decided by the most strict one, say, the highest distortion level over all the links on its path. If a link contains some user who is bottlenecked at other links, the link should know this user’s frame dropping level, reduce its capacity by this user’s flow, and re-allocate a common threshold of dropping frames for its other users with the remaining capacity, leading a lower common distortion threshold, or equivalently, getting through more traffic for its other users. In this way, each user gets distortion as low as possible and fairly, and the distortion of the user who experiences most stringent bottleneck is minimized, achieving the min-max distortion fair share.

B. Algorithm for the Optimal Threshold of Dropping Frames

We propose the following algorithm to find the optimal threshold of dropping frames for each user.

Algorithm

For a given interval \( W \), the links and users do the following.

Initialization: All links set ‘Done=0’ and \( v_l = \max U \); All users set ‘Done=0’ and \( z_s = \max U \), where \( \max U \) is a given upper bound of the frame importance.

Iterations:

1. If link \( l \) ‘Done=0’,
   - \( l \) reduces its capacity \( C_l \) by the rate (determined by \( z_s \)) used by users marked ‘Done=1’.
   - \( l \) calculates a common threshold \( v_l \) for the remaining users within the remaining capacity.
   - If link current \( v_l = \) previous \( v_l \)
     - \( l \) set itself ‘Done=1’ and the link sets all its users ‘Done=1’ and \( z_s = v_l \)
   - End If
   - End If

2. If user \( s \) ‘Done=0’,
   - \( s \) set the frame dropping level \( z_s = \max v_l \) over all the links on its path.
   - Need a backward message passing.
   - End If

In the algorithm, each link finds an equal distortion level to its users not marked ‘Done’; each user sets its frame

Fig. 3. Illustration of distortion fair sharing.
dropping level as the most stringent one over all the links on its path; afterwards, if the link is bottlenecked based on its users updated dropping level, the link marks itself and the users on it ‘Done’, and other links reduce its bandwidth by the flow of the users marked ‘Done’; then the iteration goes to another round. The algorithm iterates among links and then flows, until the thresholds can not be reduced further. The criterion to decide whether a link is bottlenecked is that if the link is fully loaded or if the dropping level is zero at this link in the current iteration round. Note that the concept of the bottleneck is extended to the bottleneck for the case that the flows are fully elastic, which means the flow can always increase if the link permits. In our case for the video clip which is already coded, the flow has a fixed maximum rate, hence if the link has plenty of bandwidth which can support the maximum transmission rate of video, the frame dropping level is zero, but if the flow is fully elastic (no upper bound), the link will be fully loaded.

C. Number of Iterations of the Algorithm

The following Proposition is straightforward.

**Proposition 1:** The number of iterations of the algorithm is at most the number of bottlenecks, where the bottleneck is counted in the same network where all the flows are fully elastic without upper bound.

**Proof:** In each iteration, there will be at least a link whose current threshold \( v_l \) will not change in the next iteration, and such link is a link which would be fully utilized if the flows on it are fully elastic without upper bound.

V. SIMULATIONS

Here we show some simulation results. These results are only meant to demonstrate the effectiveness of the proposed scheme, not a full scale simulation with real network.

When the network capacity is limited, if we do not allow frame dropping, bottlenecks will appear at certain link, with resulting delays creating playback “freeze ups” for certain flows. This is not desirable from end users’ point of view. Instead, we allow certain frames to be dropped to clear up bottlenecks and to meet the stringent delay QoS requirements.

Obviously, most routers nowadays have packet drops in a content-blind fashion, for example, if a link is a first-in-first-out (FIFO) queue with a fixed maximum queue length limit, and if the queue meets the maximum length, the incoming frames are dropped. When there is congestion, if the packet drops are content blind, an I frame could be dropped with resulting catastrophic loss of quality for the whole GoP.

The above approaches are obviously not good when the links are congested. Due to the space limitation, we omit the comparison of CAF with these approaches. Instead, we show the comparison with the following baseline approach.

A baseline approach for comparison is that the frames from users are served at each link in a FIFO fashion, and assume for time interval \( W \) (seconds) of one GoP, link \( l \) can only serve frames with total length of \( W c_l \) (kbits) and the remaining frames in the GoP of the users on this link are lost, for the purpose of stabilizing the queueing. This also guarantees that the most important frame, I frame, in each GoP will be served if link capacity permits, and all the I frames will be arriving in time.

A. A Two-Link Three-User Network

First, we investigate a network with two links and three users, as shown in Fig. 4.

![Fig. 4. A two-link three-user network.](image)

Case a: User 1,2,3 all send video clip Foreman.
Case b: User 1,2 send Foreman, while user 3 sends Akiyo.

The link capacity: \( c_1=150\text{Kbps}, c_2=250\text{Kbps} \). (Same results if \( c_1 \) and \( c_2 \) are swapped.) Video Foreman is encoded as vbr 144Kbps, and Akiyo is encoded as vbr 112Kbps. Each GoP has 15 frames, IBPBPBP BP. Figure 5 shows the utility of each frame (taking into account of the frame dependency) of the first 150 frames of the video. It can be seen that Akiyo has smaller distortion than Foreman because Akiyo has less motions.

![Fig. 5. Video sequences and their utility of each frame.](image)

For Case a, link 1 is congested and user 1,3 have the same frame dropping. Link 2 is not congested and user 2 does not have frame dropping. We compare CAF with the baseline approach, where for CAF, the interval \( W \) is one GoP. Figure 6(a) shows the perceptual distortion after frame dropping w.r.t. playback time. It can be seen that CAF approach can achieve much smaller distortion. Figure 6(b) shows the queue length at the congested link 1. CAF and the baseline approach have the similar queue length, and each can stabilize the queue. Note that for the purpose of comparison, we align I frames of user 1 and 3, but in practice, CAF does not need any alignment of users. The computing of threshold of dropping frames does not need any alignment and it can be done on-the-fly.

We compare Case a and b in Fig. 7 to show the advantage of CAF on content-aware distortion fair sharing. Figure 7(a)-(c) show the frame dropping, where the boundary is the frame-by-frame distortion \( d(f_k, f_{k-1}) \) without considering the frame...
dependency, which gives an indication of activity levels in the video sequence. When two Foreman clips share link 1 as in Case a, the frames are dropped a lot due to the congestion. When Foreman and Akiyo share link 1 as in Case b, they have a common threshold of dropping frames because user 3 is bottlenecked by link 1. Since Akiyo does not have many motions, the distortion is smaller, a common threshold makes more frames dropped in Akiyo video clip, but Foreman video of user 1 benefits less frame dropping. In this case, both user 1 and user 3 are fairly happy about the video quality. From Fig. 7(d), the fair happiness about the video quality can be seen.

In the following we discuss how the interval of video for computing the threshold of dropping will affect CAF approach. For Case a, Fig. 8 shows that if the time interval $W$ is larger, the distortion can be smaller. This is because that a large $W$ makes it possible for more frames to participate the cooperation which lets the important frames get through. A large $W$ can also reduce the computation frequency for the threshold of dropping. But a large $W$ may not track well the time varying characteristics of the video clips and it may not track the dynamics of user joining and leaving. In practice, $W$ needs to be chosen considering these tradeoffs.

B. More Complex Network Topology

We tested CAF approach in more complex networks and our experiments demonstrate effectiveness of CAF. Here we show a random example where the network has 7 links and 5 users, as shown in Fig. 9. Users 1, 3 send video Foreman, users 2, 4 send Akiyo, user 5 sends Carphone (vbr, 144 Kbps). The utility of each frame taking into account of frame dependency for "Carphone" is shown in Fig. 10.

There are 3 bottlenecks in the network, links 1, 5, and 6. We verified that after 3 iterations of our algorithm, as indicated in the Proposition. Our CAF approach yields the distortion after frame dropping: for user 1 and 2, they are the same as user 1 and user 3 in the previous Case b in Fig. 7(d) of the 2-link 3-user network; for user 3 and 5, the distortion is shown in Fig. 11; and for user 4, it only has one B frame dropped and the
The advantages of our CAF approach include: 1) It is based on the content aware frame dropping. It yields better end-to-end video quality when the network is congested. 2) It provides the possibility that video clips help each other to get a fair quality of service (in the sense of perceptual distortion) when they share the resources (link bandwidth). 3) The threshold of dropping frame can be computed on-the-fly, based on the time-varying video content. There is no need to estimate a utility function which is hard to track the time-varying characteristics of video. 4) It can naturally handle random events such as user joining, leaving, etc. For future work, we may utilize more advanced coding features like scalability and multiple description into the framework and develop more solutions.

ACKNOWLEDGEMENT
This work was in part supported by NSF CCF-0448012 and DARPA grant HR0011-06-1-0008. We gratefully acknowledge stimulating discussions with Bede Liu and Jennifer Rexford.

REFERENCES