Layered Internet Video Engineering (LIVE): Network-Assisted Bandwidth Sharing and Transient Loss Protection for Scalable Video Streaming

Xiaoqing Zhu, Rong Pan, Nandita Dukkipati, Vijay Subramanian and Flavio Bonomi
Advanced Architecture & Research Group, Cisco Systems Inc., San Jose, CA 95134, U.S.A.
{xiaozhu, ropan, ndukkipa, vijaynsu, flavio}@cisco.com

Abstract—This paper presents a novel scheme, Layered Internet Video Engineering (LIVE), in which network nodes feed back virtual congestion levels to video senders to assist both media-aware bandwidth sharing and transient loss protection. The video senders respond to such feedback by adapting the rates of encoded H.264/SVC streams based on their respective video rate-distortion (R-D) characteristics. The same feedback is employed to calculate the amount of forward error correction (FEC) protection for combating transient losses. Simulation studies show that LIVE can minimize the total distortion of all participating video streams and hence maximize their overall quality. At steady state, video streams experience no queuing delays or packet losses. In face of transient congestion, the network-assisted adaptive FEC scheme effectively protects video packets from losses while minimizing overhead. Our theoretical analysis further guarantees system stability for an arbitrary number of streams with heterogeneous round trip delays below a prescribed limit. Finally, we show that LIVE streams can coexist with TCP flows within the existing explicit congestion notification (ECN) framework.

Index Terms—scalable video streaming, media-aware bandwidth sharing, forward error correction (FEC), explicit congestion notification (ECN)

I. INTRODUCTION

Recent years have seen a rapid growth of video traffic over the Internet. According to [1], Internet video is now approximately one-third of all consumer Internet traffic, and will account for over 60 percent by year 2013. In addition to the sheer volume increase in traffic, video streaming applications also require persistently high bandwidth and timely packet delivery to ensure continuous media playout. The compressed video streams are also sensitive to packet losses. Most research that address these challenges have adopted end-to-end schemes in which the video senders infer network condition from estimated packet delay and loss statistics [2]. Such schemes can only react to ongoing network congestion or packet losses, and suffer from long queuing delays and persistent packet losses even at steady state. Moreover, they usually cannot adapt agilely enough to abrupt changes in traffic or network conditions, e.g., sudden arrival of new streams in a fully utilized network.

The benefit of active network participation for streaming video has long been recognized. Most existing designs are based on the bandwidth reservation model [3]. Such an approach requires network nodes to maintain per-flow states and incurs high implementation complexity, especially at high link speeds. On the other hand, simpler forms of explicit network feedback have been widely explored, for instance by means of explicit congestion notification (ECN) [4]. Most earlier works along this direction are designed for general Internet traffic, not for streaming video.

In this paper, we study how simple network participation can benefit video streaming in a novel framework. Our system design follows a few basic criteria. First, an ideal bandwidth sharing scheme should result in no standing queues at network nodes, so as to avoid queuing delays and persistent packet losses at steady state. Second, when multiple video streams compete for shared network bandwidth, it is desirable that their allocated rates reflect their respective rate-distortion (R-D) characteristics, while maintaining comparable rate share for background non-video traffic. Furthermore, the loss protection mechanism should provide sufficient protection against transient losses, yet avoid overhead when losses are unlikely. Finally, the system should be stable for arbitrary network topologies, scale with arbitrary number of streams, and react fast to changes.

We present the design, analysis, and performance evaluation of such a scheme, named Layered Internet Video Engineering (LIVE). Under LIVE, network nodes feed back locally calculated congestion information to video senders. Such information guides the video sender in calculating both the optimal allocated rates and the percentage of forward error correction (FEC) protection. The streams are pre-encoded using the scalable video coding (SVC) extension of the H.264/AVC standard, allowing on-the-fly rate adaptation according to the allocated rates and recommended FEC percentage.\footnote{The basic LIVE scheme may also accommodate other forms of video rate adaptation such as bitstream switching, transcoding, or packet pruning from a non-scalable stream. We leave such explorations for future work.}

Our simulation evaluation shows that LIVE’s simple network feedback allows the video streams to adjust to network congestion in a proactive manner, without incurring large queuing delays or packet losses. The feedback also facilitates fast convergence of the video rates. LIVE’s media-aware bandwidth sharing scheme leads to more balanced qualities among video streams than conventional fair-rate allocation. LIVE’s network-assisted FEC scheme effectively protects video packets from transient congestion without compromising the video quality at steady state. Our theoretical analysis guarantees that the LIVE system is stable for arbitrary number of video streams with heterogeneous round trip times below a prescribed limit. Our proposed ECN-based implementation of LIVE further assures coexistence of video streams with competing TCP flows.

In what follows, Section II reviews prior research related to LIVE. Section III explains the scheme in detail. Section IV presents simulation study of the proposed scheme. In Section V, we present an ECN-based solution for coexistence of LIVE and TCP streams. Section VI concludes the paper and discusses future work.

II. RELATED WORK

A. Multi-Stream Bandwidth Sharing

For bandwidth sharing among multiple streams, TCP-friendly rate control (TFRC) [5] is commonly used for guiding the video rate adaptation per stream. Most such schemes rely on end-to-end packet statistics, and achieve the same TCP-friendly rate allocation irrespective of the streams’ R-D characteristics [6]. Previous work such as ECN [4] and MaxNet [7] has explored the benefit of explicit network feedback, without specifically catering to streaming video.
The importance of R-D optimized rate allocation has been recognized for multiplexing multiple video streams over a common bottleneck link [8]. Most such schemes are centralized in nature, and require knowledge of video R-D information of all participating streams at a common entity. A mathematical framework for distributed rate allocation has been presented by Kelly et al. [9]. This framework is combined with video R-D information for reducing quality fluctuation in an end-to-end scheme [10]. It is also extended to rate allocation for wireless video streaming [11]. LIVE incorporates explicit network feedback within Kelly’s framework, and augments its design for streams with heterogeneous round trip times.

B. FEC Protection for Layered Video

A vast body of literature has explored how to protect video streams against packet losses using FEC. Most work studies the application of FEC in settings with no feedback between the senders and receivers, as in multicast and broadcast video [12]. In LIVE, on the other hand, the FEC scheme is designed for systems with feedback. The second category of work describes how to adapt the FEC protection level in a unicast setting based on end-to-end packet loss rates measured at the sender [13]. The losses are usually due to transmission errors, e.g., over wireless links, whereas LIVE applies network-assisted FEC to protect against congestion-induced losses.

Previous work has also investigated how to optimally allocate FEC protection among different layers of the video stream bearing different importance [12]. These studies are complementary to LIVE’s approach, and can be applied to enhance the current scheme with media-aware FEC protection.

III. THE LIVE SCHEME

We now introduce the LIVE system as shown in Fig. 1. Each network node periodically calculates its virtual congestion level (VCL). The sender calculates the optimal rate based on its own video R-D parameters, as well as the maximum VCL value reported from the receiver. The VCL information is also used to adjust the FEC protection percentage, which determines the final SVC streaming rate.

Next, we describe each system component in greater detail. We keep our descriptions of the scheme at a generic, conceptual level, and defer discussion on deployment-related issues till Section V.

A. Media-Aware Bandwidth Sharing

Conventional bandwidth sharing schemes typically aim at allocating equal rate among competing flows, assuming the same rate utility function for each flow. In LIVE, we propose to share the bandwidth among competing video streams in a media-aware fashion, i.e., to

minimize the total distortion of all streams while achieving target utilization at the bottleneck link. This can be formulated as:

$$\min_{\{r_i\}} \sum_i d_i(r_i)$$

s.t. $$y_i = r_i + \sum_{\ell \leq i} r_{\ell} \leq \gamma c_l$$

where $$d_i(r_i)$$ denotes the R-D function of each video stream. The capacity of the bottleneck is denoted as $$c_l$$; the total rate over the link is $$y_i$$, comprising of the rates of all video streams $$r_i$$'s and rate of all non-video streams $$r_l$$ traversing that link; the target utilization $$\gamma$$ is chosen to be slightly less than unity. We adopt the parametric model from [14] for characterizing video R-D tradeoff curves: $$d_i(r_i) = d_i^0 + \theta_i (r_i - r_i^0)$$. The parameters $$d_i^0$$, $$r_i^0$$ and $$\theta_i$$ are fitted from empirical R-D points of a pre-encoded video stream for each group of pictures (GOP).

The optimization problem in (1)-(2) can be solved in a distributed manner, following Kelly’s framework [9]. At the network node, the virtual congestion level $$q_l(t)$$ for Link $$l$$ is updated as $$\hat{q}_l(t) = \kappa(y_l(t)/c_l - \gamma)$$. The initial value of the virtual congestion level is chosen as $$q_l(0) = 0$$.

At the video sender, the optimal rate is calculated based on the maximum congestion level along its path: $$\hat{q}_l(t) = \max_{\ell \leq i} q_{\ell}(t)$$. The optimal rate depends on the video R-D function $$d_i(r_i)$$, as: $$r_i^0(t) = \arg\min_{r_i} [d_i(r_i) + \hat{q}_l(t)r_i] = r_i^0 + \sqrt{\theta_i/\hat{q}_l(t)}$$. As illustrated in Fig. 2, the optimal allocation balances between the competing needs of increasing rate to reduce video distortion, and decreasing rate to avoid network congestion. A higher virtual congestion level leads to a lower allocated rate whereas a lower virtual congestion level encourages a higher video rate. In addition, it can be noted that the same value of $$\hat{q}_l$$ leads to different optimal rates for video streams with different R-D parameters.

We now build upon this basic solution a practical design to guarantee system stability with streams of arbitrary round trip times (RTTs), while remaining simple and scalable. At the network node, calculation of the congestion level is updated once every time interval $$\tau$$, so as to limit the extra processing burden imposed on the network nodes. The procedures at the network nodes are summarized as below.

**Network Nodes:**

*Every update interval*

Calculate the virtual congestion level as:

$$q_l(t) = q_l(t - \tau) + \kappa(y_l(t)/c_l - \gamma)\tau.$$  \hspace{1cm} (3)

*Upon packet arrival*

For LIVE video packet, update the virtual congestion level packet header field as $$\hat{q}_l(t) = \max(q_l(t), \hat{q}_l)$$.

At the video sender, the update of video rate is modified to compensate for the effect of heterogeneous RTTs experienced by each stream. First, each video sender predicts the current bottleneck congestion level both from the past sample $$\hat{q}_l(t - \tau)$$, and the
freshly received sample \( \hat{q}_i(t) \). This predicted congestion information, \( \hat{q}_i(t) \), is used for optimal rate calculation. Second, we make gradual rate updates to approach the target optimal rate \( r^*_i(t) \). The update step sizes are inversely proportional to the RTTs, to avoid big rate swings for streams with long RTTs. The steps at video senders are summarized as below.

\section*{Video Senders:}

Upon receiving an ACK packet

1. Predict current congestion level

\[ \hat{q}_i(t) = \hat{q}_i(t) + \alpha \left( \frac{\hat{q}_i(t) - \hat{q}_i(t - \tau_i)}{\tau_i} \right). \tag{4} \]

In (4), \( \tau_i \) denotes the rate update interval; \( \hat{r}_i \) is the estimated round trip time; \( \alpha \) designates the reference time for congestion level prediction.

2. Calculate the optimal target rate as:

\[ r^*_i(t) = r^0_i + \sqrt{\frac{\theta_i}{\hat{q}_i(t)}} \tag{5} \]

3. Gradually approach the target rate with step sizes tuned by RTTs:

\[ r_i(t) = r_i(t - \tau_i) + \frac{r^*_i(t) - r_i(t - \tau_i)}{\eta \tau_i}. \tag{6} \]

In (6), \( \eta \) is the rate update scaling factor.

The following theorem formally establishes stability of the proposed practical design:

**Theorem:** Let \( r \tau t \) be the maximum round trip time in the system, \( d_{\text{min}} \) be the minimum distortion and \( r_{\text{max}} \) be the corresponding maximum video rate in the system. Further assume that \( \alpha \gg r \tau t \) and \( \eta \gg 1 \). Then, given

\[ \kappa \leq \frac{\pi \eta (d_{\text{min}} - d^0)}{\gamma \alpha (r_{\text{max}} - r^0)}, \tag{7} \]

the overall system is stable for any number of streams with round trip times less than \( r \tau t \).

**Proof:** see [15].

\section*{B. Network-Assisted Adaptive FEC}

The simplest protection against transient network congestion is to always add a fixed amount of FEC within the optimal rate budget as calculated in the above section. However, fixed FEC protection unnecessarily introduces overhead during steady state, when congestion is unlikely. Instead, we propose to leverage the virtual congestion level feedback and to adapt the FEC percentage in a proactive and efficient manner.

We apply \((n, k)\) Reed-Solomon (RS) erasure codes across \( k \) video packets within each frame to generate \( n - k \) parity packets.\(^2\) The parameters \( n \) and \( k \) are adjusted on the fly for each video frame based on past and current congestion levels. This protects against any \( n - k \) lost packets within the same frame, introduces an overhead percentage of \( (n - k)/n \), and incurs an additional delay comparable to frame interval.

The protection percentage \( f_a \) is calculated based on the virtual congestion feedback \( q \), observed for each stream, as:

\[ f_a = \begin{cases} 0 & , \Delta \hat{q}_i < 0 \\ \frac{\Delta \hat{q}_i}{\Delta \hat{q}_{\text{max}} f_{\text{max}}} & , \Delta \hat{q}_i > \Delta \hat{q}_{\text{max}} \end{cases} \tag{8} \]

\(^2\) We choose RS code due to its optimality for erasure protection, while the adaptive algorithm is general enough to accommodate other channel codes.

In (8), \( \Delta \hat{q}_i \) denotes the difference between two samples of the congestion level observed at the sender. We choose the time interval between the two observations to be 200ms, so as to avoid overreacting to temporary local fluctuations in the observation. Full FEC protection at \( f_{\text{max}} \) is invoked when the difference \( \Delta \hat{q}_i \) exceeds \( \Delta \hat{q}_{\text{max}} \). The value of \( \Delta \hat{q}_{\text{max}} \) can be chosen empirically by learning from RTT statistics.\(^3\)

If the recommended FEC amount suddenly falls to zero, the scheme maintains the last positive value for at least three RTTs before following the recommendation. The adaptive scheme also dictates full FEC protection at the start of a stream, before sufficient information is collected from the network. In addition, no FEC packets are injected unless the recommended amount is greater than 5\%, so as to reduce false alarms.

\section*{C. SVC Rate Adaptation}

The final step in the LIVE system is to determine the rate of the SVC stream. We consider video streaming with pre-encoded contents in this work. In H.264/SVC, each video frame is encoded into multiple video packets corresponding to multiple quality layers [16]. The video packets are classified as base layer (BL) and enhancement layer (EL) packets. The video frames are organized into multiple temporal layers, in that a frame from the \((m + 1)^{th}\) temporal layer is encoded via bi-directional prediction from adjacent reconstructed frames in the \(m^{th}\) temporal layer. On-the-fly rate adaptation can therefore be achieved by sequentially omitting EL packets according to their temporal layers. A stream with \( M \) temporal layers hence has \( M + 1 \) available rate points. The video R-D parameters are fitted from a discrete set of available rates and qualities, and are stored as meta data along with the streams.

Given a target rate \( r \) calculated from media-aware bandwidth sharing and a recommended FEC percentage \( f_a \), the SVC stream rate \( r_{\text{svc}} \) is determined as:

\[ r_{\text{svc}} = r_m, \quad r_m \leq (1 - f_a)r < r_{m+1}, \quad 0 \leq m \leq M, \tag{9} \]

where the set of rates \( r_m \)'s denote available rate points for the stream. The rest of the optimal rate is padded with FEC packets, as \( r_{\text{fec}} = r - r_{\text{svc}} \). In practice, this rate can only be approximated when transmitting each video frame comprising \( n \) network packets, by adding \( k = \lfloor nr_{\text{fec}}/r \rfloor \) FEC packets.

\section*{IV. PERFORMANCE EVALUATION}

\subsection*{A. Simulation Setup}

We evaluate performance of the LIVE scheme in 2 simulations. At the network node, the target utilization is \( \gamma = 95\% \); the virtual congestion level update interval is \( \tau = 10\text{ms} \); the update scaling factor is \( \kappa = 0.1\text{MSE/Kbits} \). At the video sender, the reference time for virtual congestion level prediction is \( \alpha = 250\text{ms} \) and the video rate update scaling factor is \( \eta = 4.0 \).

Two video sequences are used in the simulations: Harbor and City. They have a spatial resolution of \( 704 \times 576 \) pixels per frame, and a temporal resolution of 30 frames per second. Each stream is encoded using the H.264/SVC reference codec [17] with two quality layers and a GOP length of 32 frames, corresponding to 6 temporal layers and 7 rate points. The video packets are further segmented into network packets with a size of 1500 bytes for transmission. Upon receipt of every network packet, the receiver feeds back an acknowledgement (ACK) packet containing the maximum VCL value along the forwarding path.

\(^3\) In our current implementation, it corresponds to an oversubscription limit by 50\% of link capacity for the entire duration of 200ms.
B. Study of Transient Behavior

We first illustrate the basic dynamics of the media-aware bandwidth sharing scheme in a simple scenario. The Harbor and City streams share a bottleneck link with a capacity of 4Mbps. Figure 3 shows traces of total traffic rate, virtual congestion level (VCL) values, and allocated video rates. When only Harbor is active, the link can easily accommodate the maximum rate of the stream, and the congestion level remains at zero. When City starts streaming at \( t = 20s \), the instantaneous traffic rate over the link exceeds its capacity, leading to a sharp increase in the congestion level and drives both streams to reduce their rates. Note that the more complex Harbor streams a higher rate than City. When Harbor stops streaming at \( t = 40s \), the congestion level quickly drops to zero, thereby allowing the remaining City stream maximum rate and quality.

As a reference, Fig. 4 compares the LIVE scheme against TCP-friendly rate control (TFRC) [5]. Given the virtual congestion level information from network nodes, rate allocation in LIVE converges much faster than in TFRC. The TFRC scheme further suffers from standing queues and occasional packet losses at steady state, resulting in drastic drops in ongoing video quality and higher packet delivery delay. The LIVE scheme, in contrast, manages to avoid packet losses completely in this scenario. It yields an empty queue at steady state, thereby reducing the packet delivery delay. When both streams are active in the network, the allocated video rates from LIVE reflect differences in the video R-D characteristics of the two streams. Allocation from TFRC, on the other hand, result in equal rate allocation among the two streams and a larger gap in their video qualities.

C. Effect of Media-Aware Bandwidth Sharing

Next, we examine the optimality of LIVE’s media-aware bandwidth sharing scheme. For the simple case of two streams sharing a single link, we plot the average video quality of both streams as achieved by LIVE, by fair-rate allocation, and by all other admissible allocations. The latter refers to various rate combinations of both streams that do not exceed the target link utilization. Figure 5 shows such comparisons for the sequence pair Harbor vs. City over a bottleneck links of capacity 4Mbps. It can be noted that LIVE achieves the highest average video quality of both streams among all admissible rate allocations. It outperforms fair-rate allocation by 0.57 dB in PSNR of the average video quality.

D. Effect of Network-Assisted FEC

We now study the network-assisted FEC scheme in a demanding scenario. Three additional video streams simultaneously arrive at a single link previously occupied by a single stream. The link capacity is 10Mbps; all 4 video streams experience an RTT of 240ms and have the same content of Harbor. Figure 6 compares the network-assisted FEC scheme against two other heuristics operating at extreme measures: the No-FEC scheme never injects FEC packets; the Max-FEC scheme always streams the video at base-layer quality only, and the proposed network-assisted adaptive scheme are successful in recovering most packet losses, therefore both achieve a higher video quality than the No-FEC scheme. At steady state, video quality from the network-assisted adaptive scheme is only slightly lower than the No-FEC scheme. It outperforms the Max-FEC scheme, as the latter compromises video quality at steady state with constant FEC overhead.
Fig. 7. Coexistence of LIVE streams and TCP streams within the ECN framework.

V. COEXISTENCE WITH TCP FLOWS

So far we have described and evaluated the LIVE scheme from a clean-slate perspective. In reality, however, success of this scheme also hinges on how well it can coexist with legacy systems. We address this crucial question in this section, and present a solution for LIVE’s coexistence with TCP flows while retaining most of its benefits.

We achieve this goal by leveraging the existing explicit congestion notification (ECN) mechanism [4]. A random marking probability $p$ is calculated as $p = p_{\text{max}} q / q_{\text{max}}$, based on the current congestion level $q$, the maximum congestion level $q_{\text{max}}$, and the maximum marking probability $p_{\text{max}}$. With a probability of $p$, the network node then marks the Congestion-Experienced (CE) bit within the IP header of traversing packets, as specified by [4].

The parameters $q_{\text{max}}$ and $p_{\text{max}}$ are globally defined, so that the receiver can recover the congestion level based on estimated packet marking ratio $\tilde{q}$, as $q = q_{\text{estm}} \approx \frac{q_{\text{max}} p_{\text{max}}}{\tilde{q}}$. The value of $q_{\text{estm}}$ is reported to the video sender in an application-layer packet header field. Subsequently, the sender calculates the optimal rate according to (4) - (6), substituting $q_{\text{estm}}$ for $\tilde{q}$. Since $q_{\text{estm}}$ is essentially a low-pass filtered observation of $\tilde{q}$, the scheme is expected to converge more slowly, without affecting the final allocation result.

No modification is required for the TCP flows, as long as the TCP sender and receiver react to the ECN markings according to [4]. A random marking probability for LIVE's coexistence with TCP flows while retaining most of its benefit from LIVE.

VI. CONCLUSIONS

In this paper we have described LIVE, a new architecture for streaming video in the Internet. Our simulation studies and theoretical analysis show that video streams benefit immensely from simple explicit network feedback. The media-aware bandwidth sharing scheme in LIVE allows the bottleneck link to be shared in an efficient manner, minimizing total distortion of all competing video streams. Rate allocation adapts to transient events with fast convergence, while avoiding steady-state queuing delays or persistent packet losses. With the network-assisted FEC scheme, a video stream can proactively shield itself from transient network congestion before its quality is impacted. Going forward, we will explore how real networks can benefit from LIVE.

REFERENCES