

Multi-source Cooperative Adaptation for QoE-aware Video Multicast Rate-control

Kaliappa Ravindran

Department of Computer Science, Graduate School and University Center
The City University of New York, New York, NY 10019, USA.
Email Address: ravi@cs.cuny.edu;

Abstract—We consider a wide-area video conferencing application where the video sources adapt their send rates according to the available bandwidth in the network paths. We advocate a *QoE-aware cooperative rate control* of the sources to relieve the congestion, instead of running multiple (independent) instances of a single-source adaptation algorithm in a QoE-oblivious manner and additively superposing their results. Our paper focuses on the architecture of such a QoE-aware video multicast system. Dove-tailed to the core functionality of rate adaptation is the session-layer configuration control mechanisms to deliver video to various end-user devices.

I. INTRODUCTION

We consider a wide-area video conferencing application where the sources adapt their video send rates to the available bandwidth in the multicast paths set up over an underlying data transport network to reach various receivers (wired or wireless devices). In our system architecture, a multicast receiver is basically a proxy node for a group of mobile client devices serviced in a certain geographic region to stream conference-quality video over wired or wireless access networks. We infuse QoE-awareness in the receiver proxy nodes servicing groups of client devices with diverse capabilities. See Figure 1. In this mobile application setting, there are two issues in employing multicast congestion control:

- 1) Congestion topology inferencing that allows the video sources to *cooperatively* adapt their send rates matching the available bandwidth in network paths leading to the receiver agents (using AIMD-like adaptation algorithm);
- 2) Managing the groups of mobile devices serviced by receiver proxies in a way to enforce the desirable QoE properties globally across the multicast session (e.g., fairness, isolation) while taking cognizance of the local context of device capabilities (e.g., video encoding).

In this paper, we outline the algorithmic issues in a realization of the receiver proxy nodes: namely, participate in a global congestion control via the end-to-end *packet-loss reporting* mechanism, and a management of mobile devices via the session-layer *configuration management* mechanism.

Consider the video multicast scenario shown in Figure 1. Say, the sources V_a and V_b send data at the rates λ_a and λ_b respectively, which flow over a bottleneck link $\text{lnk}(x, z)$ on their way to the receiver agents R_1 - R_7 . All receivers except R_2 are serviced via low-to-moderate congested links. The end-

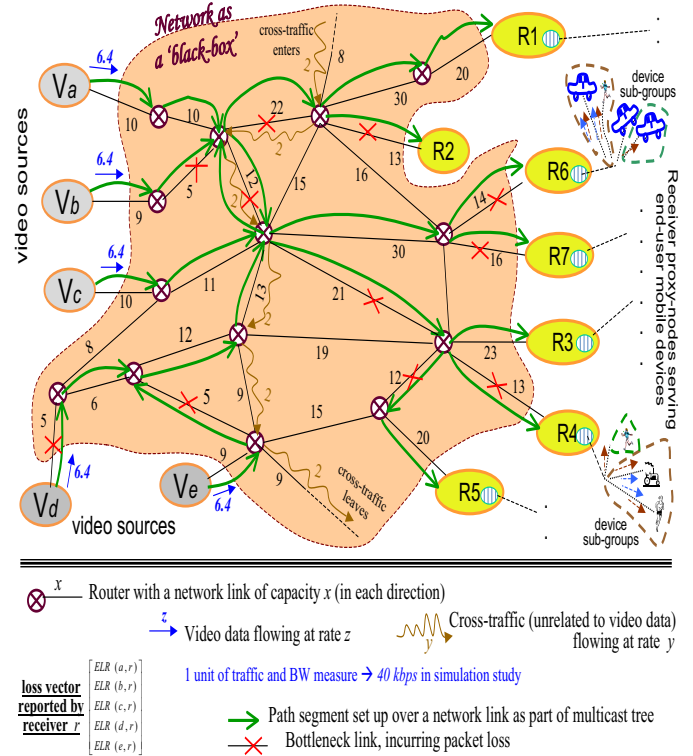


Fig. 1. Multicast network system structure in our approach

to-end packet loss seen by a receiver for these flows is:

$$L_{(a/b,r)} = \left[1 - \frac{av(x, z)}{\lambda_a + \lambda_b} \right] \text{ for } r = R_1, R_3 - R_7$$

where the $av(x, z)$ is the available bandwidth on $\text{lnk}(x, z)$ and $\lambda_a + \lambda_b > av(x, z)$. The goal of an adaptation algorithm is to reduce λ_a and λ_b to λ'_a and λ'_b respectively such that $\lambda'_a + \lambda'_b < av(x, z)$. The question is about splitting the rate reduction $[\lambda_a + \lambda_b - \lambda'_a - \lambda'_b]$ between the sources V_a and V_b in a way that is determined by application objectives (instead of randomly apportioning the reduction between V_a and V_b). If V_a has a higher priority than V_b for instance, it is reasonable that V_b be required to reduce its rate by a larger amount than V_a . This however requires a coordination among V_a and V_b , at an algorithmic level, in orchestrating their rate reductions.

A receiver proxy R resides in the leaf node of a multicast tree that distributes the video data from various sources. R acts as a 'protocol gateway', mapping the QoE parameters of various client devices onto a composite session-level rate control QoS. The session-level configuration base (SCB) maintained by R describes the capabilities of various client devices attached to R : such as the encoding, bit rate, and display size supported. The SCB also maintains a list of functions to transcode the source data onto the formats expected by various devices, and a list of mappers that provide a coarse estimate of the frame rates to be generated matching the supported bit rates. The device admission related functions inter-work with the rate control functions hosted in R , using the information maintained in the SCB. The protocol for local feed from R to the attached devices can embody customized functions: such as a cooperative local relay of video among the devices and a transcoding of video to suit specific device capabilities.

From a QoE standpoint, both the core functionality of rate adaptation for congestion relief and the non-functional goals of staging the video to end-user devices need to be dealt with. An example is the YouTube feed from one device to another during a popular event such as a Soccer game watched by a larger number of users in a geographic area. Here, the video feed may originate from multiple cameras capturing: say, the actual game played at one place and the commentator box located at another place, with the former assigned a higher priority. A congestion in the core network path causes a priority-based reduction in the video feed rates to a receiver proxy R , which then gets distributed to various end-user devices via the local access network signaling. If the rate reduction is oblivious of the session-layer priority of multiple video feeds, the resulting rate assignments may not be QoE-cognizant. On the other hand, even if the rate adaptation process in the core network is priority-based, the local feeds via the access network may not be QoE-aware. Thus, both these aspects should be considered in the design of a rate adaptive video multicast system.

Our paper focuses on the architecture of a QoE-aware video multicast system that embodies rate adaptation based congestion relief as a part of its core functionality. We consider two algorithmic elements:

- 1) Inferencing of congestion topology based on the 'packet loss' reports from video receivers $\{R\}$;
- 2) Cooperative rate adjustment of video sources based on the inferred congestion topology.

Dove-tailing this core functionality is the session-layer configuration mechanism to deliver video to the end-user devices. The SCB housed in a receiver node R stipulates the access network signaling needed for device-level video delivery.

II. QOE-AWARE RATE-CONTROLLED VIDEO MULTICAST

A rate-controlled video multicast system, studied by researchers in video conference settings [3], [2], exercises the network bandwidths to meet application-level rate specs. It offers the means to accommodate session-wide QoE during the

rate adaptation process. The rate adjustment protocol executes in a distributed way at the source and receiver agents.

A. Characteristics of system environment

The infrastructure network consists of nodes that provide session-level path between application end-points over interconnected transport links. In one case, the transport links may be UDP based connections set up between different nodes — as in overlay-based virtual networks. In another case, the links may be native IP-based network layer path segments between different nodes. Regardless of the network type, the inter-node links may be a hybrid of low/high bandwidth transport paths, and the infrastructure nodes carry out data forwarding functions to realize session-level connectivity between the end-points. The access network over which the end-user devices connect to their receiver proxy nodes is usually made up of 802.11 wireless links and/or cellular data connections.

The core network sets up a tree-structured multicast path, with one-to-many forwarding of data packets by the on-tree nodes. In addition, many-to-one merging of packet flows from different sources in a multicast session (e.g., video/images exchanged between conference participants) occurs at the on-tree nodes. The overlay routing architectures studied elsewhere such as Scattercast [4] and End-system Multicast [5] support such multicast delivery functionalities. Data-oriented session-layer control is often relegated to the end-point hosts: such as video encoding, content filtering, and rate control.

Protocol-level *end-to-end control* of rate adaptations, as studied by researchers, allows faster recovery from congestion (in comparison to user-triggered recovery)¹. Congestion detection by packet loss observation at receivers and the recovery actions therein can possibly be orchestrated by an external management station that runs the rate adaptation protocols.

B. System-level view of multicast rate control

A set of sources $\{V_y\}_{y=a,b,c,\dots}$ multicast video packets to a set of receiver proxies $\{R_i\}_{i=1,2,3,4,\dots}$ over a tree-structured network path that implements 'best-effort' packet transport — as in video conferencing over Internet. Refer to Figure 1. The protocol agents $\{AM(y), AM(i)\}_{y=a,\dots,e;i=1,2,\dots,7}$ execute in the 'control plane', through a signaling API, to adapt the send rates of video sources. The functions to generate/disseminate video frames and to send/receive them over the multicast path reside in the 'data plane' of system architecture.

A source V_y sends packets at the desired rate λ_y , where $0 < \lambda^{(min)} \leq \lambda_y$ (e.g., $\lambda_y = 20 \text{ fps}$, $\lambda^{(min)} = 5 \text{ fps}$). The goal is to determine the sustainable send rates of sources $\{\lambda'_y\}_{\forall y}$ when one or more path segments are congested:

$$\lambda^{(min)} \leq \lambda'_y \leq \min(\{\lambda_y, \lambda^{(max)}\}),$$

where $\lambda^{(max)}$ is the maximum allowed send rate (e.g., $\lambda^{(max)} = 25 \text{ fps}$). Note, λ_y is the intended send rate of

¹Even on a software-defined network (SDN) with the ability to dynamically switch the path to different routes, a congestion scenario may arise due to uncontrolled cross-traffic flows or the connectivity provider's inability to find a non-congested route.

V_y ; whereas, the sustainable rate λ'_y is the actual send rate that keeps the loss rate suffered by V_y in the path to a receiver r within a small tolerance limit δ , i.e., $L(y,r) \leq \delta$ where $0.0 < \delta \ll 1.0$ (e.g., $\delta \approx 0.04$).

Application-level QoS specs are the video frame rates and loss tolerances: $\{\lambda_y, \delta_y\}_{y=a,b,\dots}$ — which also depict the session-wide QoE. A rate adaptation protocol adjusts the send rates of sources so that packet flows can be sustained by the available bandwidth in various network paths.

C. QoE-aware setting of adaptation triggers

Given a video multicast session, we view the video sources and receivers as part of a physical system P involving the human dissemination of video streams. Whereas, the AIMD-based procedures to compute an adjustment of video send rate λ'_y form the controller of P , denoted as C_P . The video delivery point at receiver site is the observable part of P , which feeds the input for an algorithmic computation of control actions by C_P . The adaptation goal is to keep the video rate perception quality experienced by a human viewer within acceptable levels. The safety constraint of C_P , as pertaining to the QoS monitored at the service layer, is prescribed as:

$$\left| \frac{d\lambda'_y}{dt} \right| < R_0 \wedge \lambda'_y > \lambda^{(min_1)}, \quad (1)$$

depicting that the frame rate is above a threshold $\lambda^{(min_1)}$ and the rate jitter is within a limit R_0 . The safety point of P , as pertaining to the QoE at user-level, is prescribed as:

$$\left| \frac{d\lambda'_y}{dt} \right| \leq q_0 \wedge \lambda'_y \geq \lambda^{(min_2)}, \quad (2)$$

where q_0 and $\lambda^{(min_2)}$ depict the human experiential jitter threshold and minimum send rate respectively deemed as acceptable. To factor in these human-oriented QoS aspects (i.e., QoE), we program the AIMD algorithm to generate a control trajectory that spreads out the rate reduction over multiple rounds k , where $1 \leq k \leq k_m$. The time interval of a control round T is set as: $\tau < T \leq T_m$, where τ is the time-constant of video transport system (i.e., the time taken for a rate change to impact the congestion-induced packet loss along a path) and $k_m \times T_m$ depicts the maximum allowed system inertia with human-in-the-loop. Our QoE characterization is different from the network-oriented measures described in [6].

The constraint (2) depicts the domain-knowledge made available to the video multicast service-layer agents: namely, the information about user tolerance to a jitter in frame arrival rate and a lower-than-desired frame rate. Whereas, the constraint (1) purports to define the safe execution paths of C_P , and guide the algorithm-internal operations therein. In this light, a setting of R_0 and $\lambda^{(min_1)}$ such that $R_0 > q_0$ and $\lambda^{(min_1)} < \lambda^{(min_2)}$ depicts a cross-layer flow of domain-knowledge to the service-level agents — in contrast with a random setting of R_0 and $\lambda^{(min_1)}$.

When C_P detects the safety violation $\left| \frac{d\lambda'_y}{dt} \right| \geq R_0$ or $\lambda'_y \leq \lambda^{(min_1)}$, the cases: $\left| \frac{d\lambda'_y}{dt} \right| \in [R_0, q_0]$ and $\lambda'_y \in [\lambda^{(min_2)}, \lambda^{(min_1)}]$ depict the onset of symptoms indicative

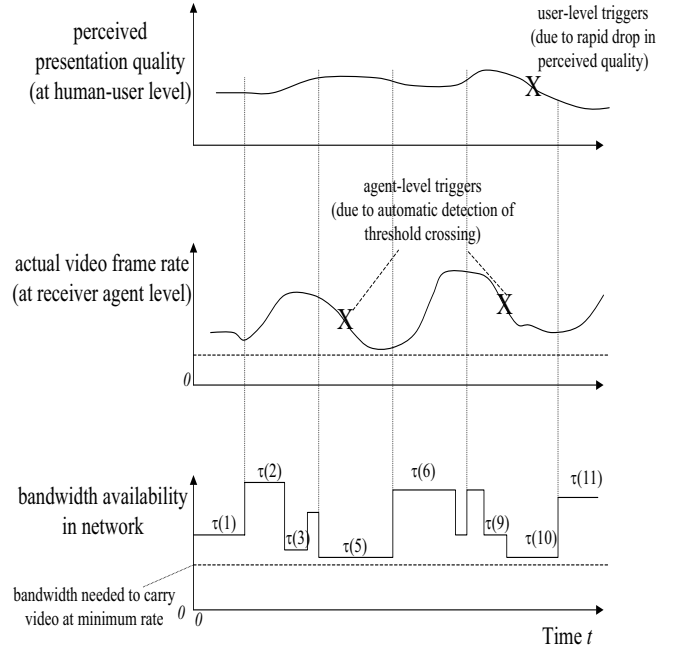


Fig. 2. QoE-aware adaptation triggers based on time-scale separation

of the possibility of rate jitter or rate depletion becoming noticeable to the human user. The safety margins $(q_0 - R_0)$ and $\lambda^{(min_1)} - \lambda^{(min_2)}$ indicate how soon the controller C_P triggers a reduction of λ'_y , to avoid QoE degradations. See Figure 2 for an illustration. The points Y and Z depict safety violations at both levels, i.e., the constraints (1) and (2) are not met — thereby requiring user-level recovery. Here, the human user notices a degradation in QoS, and then adjusts the perceptual quality expectation. Whereas, the point X depicts a safety violation of C_P but does not cause a safety violation of P . A capability in C_P to detect the scenario X and trigger agent-level recovery by, say, a bandwidth allocation or a source-level rate reduction, allows the application to continue without QoE degradation in many cases.

The main idea in the above QoE consideration is that a fast time-scale detection of an impending rate reduction by agents, and then trigger a suitable recovery before the human users see the rate reduction.

III. QOS vs QoE IN VIDEO MULTICAST RATE CONTROL

Enforcing a QoE-aware video multicast service is a challenge because the QoE specifications are hard to quantify — which is unlike the QoS parameters themselves (see [15], [14] for a general discussion on QoE versus QoS). The problem is further exacerbated due to the inherent complexity of congestion detection and relief in a distributed manner, and the consequent fuzziness in what one might consider as an acceptable QoS specs template. Thus, the mapping of a QoE specs (if it can be quantified) to the service-oriented features and parameters of a video rate-control system is a major challenge.

Based on the inferred topology, one or more sources reduce their send rates in an effort to relieve congestion, as indicated by the condition: $L_{(y,R)} < \delta$, where $L_{(y,R)}$ is the end-to-end loss ratio (ELR) experienced by a source V_y in reaching a receiver R — as² indicated in the loss report from R .

We evaluate how good the rate-control system is in reducing the send rate of loss-experiencing video sources when congestion occurs along the network paths. Here, the rate-control functions to effect congestion relief, as realized by the source and receiver agents, primarily consist of the reporting of ELRs, congestion topology inferencing, and rate adaptation scheduling. Macroscopically, the QoS of rate-control system depicts the ability of system-level support mechanisms to orchestrate the adaptation of send rates of loss-experiencing video sources. QoS is quantified in terms of parameters meaningful to video multicast receivers: such as how fast an adaptation to packet loss conditions occurs, stability & convergence of send rates, and fairness in apportionment of send rates. The application-level QoS parameters reflect the overall quality of underlying system-level support mechanisms for rate-control in effecting a relief from congestion.

QoE, on the other hand, relates to the totality of QoS support functions, as determined by what value an end-user sees from the rate-adaptive video delivery system in various dimensions of interaction with the human ergonomic aspects. In a multicast setting involving receivers $\{R\}$, a QoE parameter directly tied to the video rate-control functions is how often a receiver $r \in \{R\}$ is forced to drop out of multicast session. A drop-out of r may occur due to a severe bottleneck link close to r in the topology whereupon the system forces the other receivers $\{R\} - r$ into fielding video at a rate substantially lower than that possible in a session r is not a part of. The forced drop-out of r improves the overall fairness among receivers in their influence on video send rates³ — and hence improve the QoE.

Furthermore, every aspect of system-level QoS such as the frame rate, video encoding & resolution, and display size impacts the QoE. For instance, a video with low rate jitter increases the pleasure of end-user participating in the conference session — c.f. Figure 2. The end-user quality perception is however a qualitative factor⁴. That the client devices may possibly be serviced over a low bandwidth and fragile wireless access network in a geographic region exacerbate the QoS-to-QoE mapping issues. For instance, even if the QoS enforced by the rate-control system is good, connectivity issues in wireless access networks (such as noise and channel fading) may degrade the QoE. A good QoE however requires enforcing

²Due to the burstiness of video data traffic, compounded by the randomness in bandwidth sharing with other traffic in the network, $L_{(v,R)}$ is often a small positive value γ (where $0 < \gamma \ll \delta$) even when there is no congestion.

³A case where r drops out of the multicast session due to the inability of transport system to sustain the minimum needed data rate $\lambda^{(\min)}$ at r is different. This case, where the end-user sees no value for having r continue in the session, may arise even when the other receivers $\{R\} - r$ are sending at close to (but above) $\lambda^{(\min)}$ — and hence fairness is not an issue.

⁴The system-level QoS parameters can be appropriately set by user-level actions initiated from a device: say, to turn On/OFF the video capability and change the display size/resolution.

a high system-level QoS during rate adaptation. As can be seen, a formulation of QoE parameters involves a complex mapping of QoS-to-QoE spaces.

We now describe the baseline rate adaptation mechanisms coordinated across multiple sources.

IV. MULTI-SOURCE COORDINATION OF ADAPTATION

A key element of QoE-aware rate adaptation is the ability to detect the sharing of paths by multiple sources, and allocate therein send rates to these sources.

A. Detecting path sharing from ELR reports

Suppose $\text{lnk}(v, x)$ and $\text{lnk}(x, R)$ are the network links in the path connecting a source v and a receiver R , and λ_v is the only flow traversing these links. If both these links are congested — i.e., $av(x, R) < av(v, x) < \lambda_v$, we have:

$$L_{(v,R)} = \left[1 - \frac{av(x, R)}{\lambda_v}\right].$$

The end-point algorithm senses a congestion in the path connecting v to R when $L_{(v,R)} > \delta$. If another flow λ_u merges at node x to share $\text{lnk}(x, R)$ with λ_v , we have:

$$L_{(v,R)} = \left[1 - \frac{av(x, R)}{\lambda_v + \min(\{av(z, x), \lambda_u\})}\right],$$

where $\text{lnk}(z, x)$ depicts an upstream link traversed by λ_u before merging with λ_v . Here, the per-hop loss rate (PLR) on the congested link $\text{lnk}(x, R)$ is spread out among the elastic data flows $\{\lambda_u, \lambda_v\}$, with the loss apportionment being proportional to their send rates.

Consider the ELR experienced by V_v when sharing the congested link $\text{lnk}(x, R)$ with one or more flows $\{\lambda_u\}$ merging at node x . A case of non-congestion on $\text{lnk}(v, x)$ depicts that $av(v, x) > \lambda_v$ and $av(x, R) < \lambda_v + \sum_{\forall u} \lambda_u$. A *rate perturbation* from λ_v to $(1 - \epsilon) \times \lambda_v$ exercised on V_v in this case, where $0 < \epsilon < \delta$, splits the resulting decrease in per-hop loss among various flows in proportion to $[\lambda_v, \{\lambda_u\}]$ (the same rule of proportional split applies for rate increases as well).

In a more general case, a flow may traverse a sequence of path segments whose available bandwidths are less than the flow's bandwidth demand. These path segments can be equivalently replaced by a single path segment with the smallest of available bandwidths (i.e., the bottleneck segment). That the composite ELR suffered by the flow is the same in both the cases is a key property used in topology inferences by an end-point algorithm.

B. Topology inference for rate adaptation

There have been many works on inferring the network path topology, based on end-to-end measurements of packet loss at receivers in a multicast tree [1], [7], [8]. The inferencing mechanism for single-source trees studied in these works is used as a building-block to extend to the case of a multi-source tree topology in our paper.

Receiver agents report the ELRs to source agents, as a means of congestion notification, say, using a RTCP-like

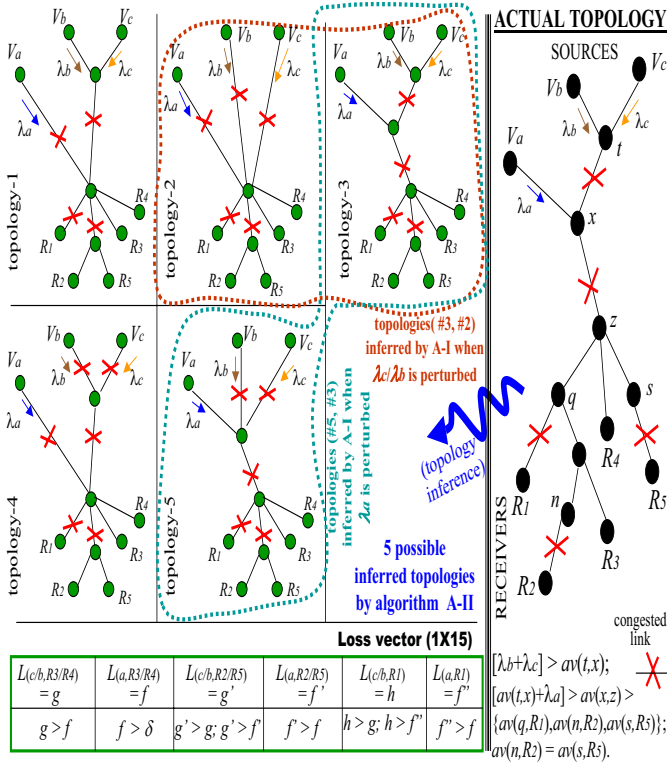


Fig. 3. Illustration of sample topology inferences

signaling protocol (the agents execute within the algorithm layer). The report from a receiver agent of $R_i |_{i=1,2,\dots}$ indicates the source-itemized ELRs experienced at R_i , based on checks of the sequence numbers assigned by a source agent of $V_y |_{y=a,b,\dots}$ for the packets from V_y . Each source agent carries out a correlation analysis of the ELRs, to infer the congestion status in one or more path segments. Here, an inferred topology is a set of path segments where the available bandwidth is insufficient to support the video packet flows of one or more sources, resulting in packet loss exceeding δ .

Consider the congestion scenario shown in Figure 3. The multicast tree interconnects 3 sources V_a, V_b, V_c and 5 receivers R_1-R_5 . The flows λ_a and λ_b merge at a node t and then merge with the flow λ_a at a downstream node x , and then fan out from the next adjacent node z towards R_1-R_5 . The links $lnk(t, x)$ and $lnk(x, z)$ are congested, and in addition, a downstream link each in the paths close to R_1, R_2, R_5 are also congested. These congested links incur per-hop loss on the flows $\lambda_a, \lambda_b, \lambda_c$ to various degrees. These losses, which occur deep in the network, manifest as increased ELRs observed by R_1-R_5 .

Based on ELR reports received and the topology inferred therein, the source agents coordinate with one another to decide which of the sources V_a, V_b, \dots should reduce their send rates and by how much. The application then reconfigures to work with these send rates (which may include the removal of a source and/or a receiver from the session, if needed).

C. AIMD-based rate adaptation rule

The sources implement a rate adjustment rule: exponential reduction in send rate when the path is congested, and additive increase in send rate when the path is not congested. This AIMD ('additive increase multiplicative decrease') rule⁵ uses the ELR reports of receivers to determine the amount of rate reduction/increase, as follows:

$$\begin{aligned} \lambda'_y(j+1) &= \lambda'_y(j) \cdot (1 - \beta_y \overline{L_y(j)}) & \text{for } \overline{L_y(j)} > \delta_{h(y)} \\ \lambda'_y(j+1) &= \lambda'_y(j) + \alpha_y & \text{for } \overline{L_y(j)} \leq \delta_{l(y)}, \end{aligned} \quad (3)$$

where $j = 0, 1, 2, \dots$ is the iteration number in a run-time execution of the rate adjustment algorithm (with $\lambda'_y(0) = \lambda_y$), $\overline{L_y(j)}$ is, say, the average ELR for source y observed across all receivers in j^{th} iteration ($y = a, b, \dots$), β_y and α_y are positive constants, and $\delta_{l(y)}$ is a loss threshold such that $\delta_{l(y)} < \delta_{h(y)}$. A computed rate $\lambda'_y(j)$ in j^{th} iteration is subject to a max-min limit, namely, it cannot exceed a maximum rate $\lambda_y^{(max)}$ and cannot go below a minimum rate $\lambda_y^{(min)}$. The constants $(\beta, \alpha, L^{(min)})$ impact the adaptation performance such as stability, steady-state error, and convergence. A larger β increases the rate of decay to the lower send rates; likewise, a larger α increases the rate of ascent to the higher send rates. A faster change in send rates may however result in a QoS jitter at the user-level.

The adapt-observe procedure is executed by the source and receiver agents over faster time-scales: say, every 2 sec. The observe-adapt time-interval T in a control round, combined with the AIMD coefficients β and α , determine the dynamics of rate adaptation. We consider two cases for reaching the final send rates just-enough to relieve congestion along a video multicast path:

- 1) Relief in the first round by a drastic rate reduction and subsequent rate increases over many rounds (i.e., $k = 1$);
- 2) Relief by phased rate reductions over multiple rounds followed by rate increases over fewer rounds (i.e., $k > 1$).

In both cases, (α, τ) determines how fast λ can be ramped up when the multicast path is not congested. The QoE aspects in 1 versus 2 impact the choice of (β, α, T) — and hence the initial trajectory planning of the controller. When the receivers are disparate (i.e., experience different loss rates and have different priorities), the quickness of rate reduction at a loss-experiencing receiver and the rate fluctuations seen by unaffected receivers impact the session-wide QoE. Our multi-source rate control algorithm strives to improve the QoE by accommodating such session characteristics in a rate schedule.

The algorithms for topology inference are based on correlation analysis of the ELR reports. The goal is to not only detect the congested path segments but also identify the flows that share a congested segment — such as disambiguating the

⁵AIMD strategies have been widely studied by the Internet research community in various contexts of congestion control and flow control, such as the dynamic adjustment of TCP window sizes based on RTT and packet loss [11], [12] and the available bandwidth based video streaming [13].

topologies 3 and 5 in Figure 3. For the algorithm details, see our earlier paper [9].

V. ALGORITHM FOR COORDINATED RATE ADAPTATION

We employ a coordinated adjustment of the video send rates of various sources to realize a common application-wide QoS objective. The coordination involves inference about the sharing of path segments to receivers among various sources, and then using this information for the apportionment and scheduling of rate reduction/increase among multiple sources. The adaptation procedures incorporate a *joint optimization* of the source send rates by factoring in the statistical multiplexing effects and the cross-dependencies in bandwidth usage that arise from path sharing. Consider, for example, the streams from sources V_a and V_b merging at a node x on their way to the receivers R_1 and R_2 . If R_2 sees more loss than R_1 for the packets from V_a (say), it implies that the packet loss of V_b as seen by R_2 is at least the same as that seen by R_1 . This inference arises from the monotonicity properties of packet loss behavior as the merged flows descend down the tree. Thus, a knowledge about path sharing allows a better realization of the QoS objectives.

A. Utility-based benefit assignments

Unlike the self-centered strategy in rate adaptation procedures adopted in existing works, we infuse a cooperative strategy among the sources to achieve an application-wide objective: namely, maximizing the QoE benefits accrued to the end-users. Incorporating a cooperative strategy requires:

- 1) Quantifying the impact of dynamically changing video send rates of various sources on application-level QoS (i.e., QoE);
- 2) Mapping the QoE impacts onto a send rate schedule that collectively maximizes the application objective.

We employ a utility-based characterization of end-user QoE vis-a-vis the video send rates, using which we develop the schema to determine the optimal schedule of send rates (say, enforcing session-wide fairness [10]).

Given the desired send rates $\{\lambda_y\}_{\forall y}$, the rate control system strives to keep the actual sustainable rates $\lambda'_y \approx \lambda_y|_{\forall y}$. Suppose $U_y(\{\lambda'_y, \lambda_y\}_{\forall y})$ denotes the usefulness of a video multicast from V_y at the send rate λ'_y sustained by the network in response to a desired rate λ_y , where $0 < \lambda'_y \leq \lambda_y$. Treating the video multicast as a network service, the user-oriented QoS utility curve may be as shown in Figure 4.

As can be seen, a QoS degradation $\overline{\lambda_y - \lambda'_y}$ is associated with a reduced utility. The QoS utility $U_y(\lambda'_y, \lambda_y)$ exhibits a *concave* behavior over long-term changes in the send rates — which depicts an increasing level of user-forgiveness to degradation in perceptual quality as λ'_y becomes closer to λ_y .

The adaptation algorithm may consider the removal of heavy loss-experiencing receivers from the multicast session, if such a removal would substantially relax the need to lower the video rate at other receivers — thereby improving the session-wide utility. The tradeoff is about serving all receivers in the group at a lower video rate *versus* removing some outliers from

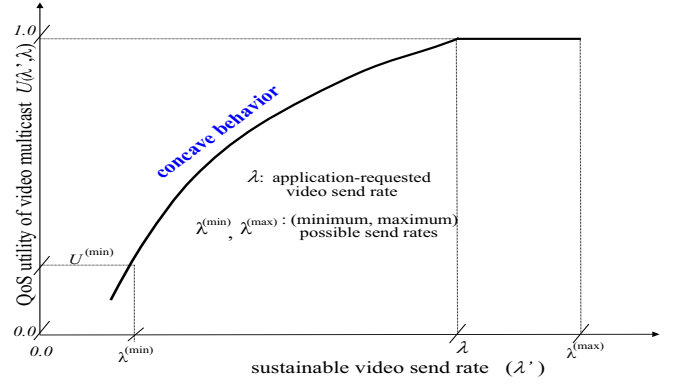


Fig. 4. Penalty assignment based on QoS utility of send rates

the group to yield a higher video rate among the remaining receivers — e.g., removing R_4 in the scenario shown in Figure 1.

A coordinated rate reduction, guided by an optimization strategy, allows a faster/smooth relief from congestion and convergence to an optimal send rate schedule.

B. Multi-source rate reduction procedures

The information on path sharing by a source V_t with a set of other sources $\{V_{t'}\}$ is maintained by our algorithm as: $slst = [V_t, \{V_{t'}\}]$, for use during rate adaptations. These sources are cross-listed in the various ELRs assembled during the loss reporting phase. The information is used in the AIMD formula to compute a rate reduction:

$$\Delta\lambda = \eta \times \beta \times L_{(q,i)} \times \sum_{\forall q \in slst} \lambda_q, \quad (4)$$

where η is a constant such that $1.0 < \eta$. We set $\eta < \eta_m$ to bound the rate reduction $\Delta\lambda$, where η_m depicts the maximum consideration for multiplexing gain. η is chosen in a way that the rate reduction exercised on the flows $slst$ is high enough to reverse the multiplexing gains already accrued by sharing a congested path segment (and hence quickly relieve congestion). The AIMD-computed rate reduction/increase $\Delta\lambda$ is split between the various sources known to be sharing a congested path segment. The rate apportionment is based on the relative QoS utility associated with these sources.

Referring to the topology inference in Figure 3, we illustrate a sequential approach to rate reduction. V_c is selected to reduce its rate first, because $L_{(c,r)} > L_{(a/b,r)} > \delta$. After the ELR of V_c reduces to acceptable level (i.e., $L_{(c,r)}' < \delta$), V_a and V_b are selected to reduce their send rates. The AIMD-computed rate reduction:

$$\Delta\lambda_{a/b} = \eta \times \overline{\lambda_a + \lambda_b} \times \beta \times L_{(a/b,r)}, \quad (5)$$

is split between V_a and V_b according to their priorities, as determined by their relative QoS utility for the application (a high priority source reduces its rate by a less amount). While V_a/V_b begin to exercise their rate reductions in parallel as above, V_c also exercises its rate reduction as:

$$\Delta\lambda_c = \lambda_c \times \beta \times L_{(c,r)}. \quad (6)$$

This concurrent rate reductions proceeds until $L_{(c,r)'} = L_{(a/b,r)'}$ — which indicates that the congestion in the upstream path of V_c has been relieved while the loss ratios of V_a/V_b are reduced, i.e., $L_{(a/b,r)'} < L_{(a/b,r)}$. Thereupon, a new rate reduction is computed as:

$$\Delta\lambda_{a/b/c} = \eta \times \beta \times L_{(c,r)'} \times \overline{\lambda'_a + \lambda'_b + \lambda'_c},$$

which is split between V_c , V_a , and V_b as per their relative utility values. It is also possible that the congestion relief on $\lnk(x, z)$, i.e., $L_{(a/b,r)'} < \delta$, occurs before that on $\lnk(V_c, x)$ — in which case V_c completes the relief on $\lnk(V_c, x)$.

The exact split rate reductions among a set of parallel sources can be guided by an optimization strategy that attempts to minimize the application-wide penalty. In this sense, our approach using the knowledge of path sharing offers a better means to search for an optimal send rate (instead of one-at-a-time control of send rates or a random parallel search).

VI. ARCHITECTURE OF RECEIVER PROXY NODES

A. 'data-plane' functions in R

We envisage three types of main functions in R that pertain to the video data transfer from R : device hand-off, video transcoding, and packet multicast. The $SCB(R)$ lists the multicast sub-groups serviced by R . A multicast sub-group entry for a device d is made up of the following:

- A logical group address L that binds to the wide-area video multicast session to which d has subscribed;
- The encoding capability $e(d)$ implemented by d to field the video data;
- The native multicast address of access network $M(d, e)$ which d listens at to receive the encoded video data;
- The network-specific attributes and parameters: such as the maximum packet size, header template, and bandwidth capacity.

The entry for d also links to the packet buffers to store the video and the local channel transmission handling methods.

Devices serviced by R over the same access network and with the same encoding requirements form a distinct sub-group. A video packet is transmitted to the sub-group with a single multicast action from R , exercised using the native multicast feature of network. Devices requiring different loss recovery capabilities (e.g., use of FEC) can either be placed in different sub-groups or be serviced via point-to-point channels, because the packets generated therein may be different.

As a device d moves from one geographic region to another, d should be removed from its erstwhile serving node R and assigned to a new serving node R' . This entails in a deletion of the device information d from $SCB(R)$ and including it in $SCB(R')$. An execution of the mobility handling protocol in the native access network is a part of this hand-off process.

B. 'control-plane' functions in R

We envisage three types of functions in R that enable controlling the network-level QoS for video data transport over the access network: channel quality assessment, channel bandwidth estimation, and group membership management.

The channel quality in an access network is determined by the environment parameters such as SNR, fading effects, and bit-error rate. The quality assessment allows determining the bit-level transmission rate C sustainable on the channel. The estimated capacity C constitutes the available bandwidth to be shared across various device-level transmission scheduled by R . A bandwidth share apportioned to the sub-group containing device d is based on an estimate of the bandwidth demand imposed by the video flow targeted to d . It is possible to take into account the statistical multiplexing effects of video flow in conjunction with other data flows emanating from R (including the differently encoded video packets of the same session). For instance, the packet loss arising from an aggressive setting of video data transmission schedules impacts the admission of devices by R . The (coarse) estimation of C provides a handle on the available bandwidth for R to decide on the admission of devices seeking subscription to a multicast session.

A mobility-triggered device handoff to/from R may manifest in updating the group membership information in $SCB(R)$. It may involve simply adding/removing a device to/from an existing sub-group, creation of a new sub-group to accommodate the needs of an incoming device, or, deletion of an existing sub-group after removing a departed device. Furthermore, a sudden drop in the channel quality may force R to suspend video delivery service to the affected devices — and hence cause their sub-groups to be deleted from $SCB(R)$. Such configuration updates involve signaling between the devices and their serving proxy nodes.

The QoE aspects of a functionality in R for the automatic suspend-and-resume of video delivery versus the user-triggered restart of a prior-stopped delivery is itself distinct from the multicast rate-control functionality.

C. OAM functions in R

R implements functions that enable the management of video streaming as a service delivery to the devices attached to R . The OAM functions, as seen from the standpoint of a device d , include the following:

- Cost assignment to d for network bandwidth usage;
- Billing of d based on revenue policy of service provider;
- Mapping of QoS-utility from the QoE specs of d .

R implements the signaling for QoE prescriptions from the various devices $\{d\}$ serviced by R . The exercising of OAM functions in a context of the devices $\{d\}$ enables R to generate a composite profile about the configuration supported by R .

The support for dynamic device configurations in R impacts the cost assignment and billing to the devices being serviced. In the presence of configuration changes, the OAM station evaluates the bandwidth share apportioned to a device d — and hence the cost assigned to d . What granularity of configuration changes is appropriate to trigger a change in cost assignment to the end-user is based on the pricing policy of service provider. Furthermore, the actual cost assignment to end-users for bandwidth usage also factors in the characteristics of access networks attached to R . In general, the QoS utility cast by

R across the multicast session can change due to dynamic variations in the device profiles serviced by R .

VII. QOE SUPPORT FUNCTIONS IN RECEIVER NODES

Given the need to service end-user devices over wireless channels of unpredictable quality, the OAM station factors in the possibility of spontaneous exit of a receiver R from the multicast session — even if the upstream wire-area link close to R is not congested under the original desired send rates. Referring to Figure 1, the receiver proxy node n may leave the configuration if the automobile serviced by n moves to a different geographic region. Basically, the algorithm recomputes the send rates of sources in the new context of a session without R . First, the constraint on rate search presented hitherto by the bottleneck links in upstream path segments close to R is now eliminated. Based on how dominant the eliminated constraint was, a recomputed send rate is at least as high as the erstwhile rate. Furthermore, the departure of R may change the global utility due to a high priority hitherto enjoyed by R . This may in turn cause the rate-control algorithm to search for a different optimal point in the rate space.

An adjustment of video send rates for QoE improvement may be triggered by dynamic configuration changes in the session that occur when one or more receivers volunteer out of the session (say, because of a low QoS experienced). The tradeoff is about sending at higher rates but with widely disparate QoS *versus* sending at lower rates but with a lower disparity in QoS.

We have conducted simulations to study the impact of spontaneously-triggered leave of receivers from a multicast session. We consider two reasons for the departure of a receiver R from a configuration: i) inability to service R through a heavily bottlenecked link, and ii) a spontaneous depletion of the devices making use of R for video delivery. Our simulation study corroborates the algorithmic elements and signaling support for a re-computation of the send rates when session-level configuration changes occur at the receivers.

Referring to Figure 1, the multicast topology highlights the need for dynamic configuration changes. The multicast session studied has five video sources V_a-V_e and seven receiver-proxy nodes R_1-R_7 , with the initially attempted send rate of sources being 25 *fps* each. For the bottleneck links shown, a final send rate of V_a-V_e that relieves congestion is: 12.16, 9.26, 9.83, 9.97, 9.03 *fps* respectively. Assuming that the sources prescribe an identical QoS utility each, a relevant performance index is the maximum combined send rate of sources — which is 50.3 *fps* in the above case.

Each of the receivers R_2 and R_5 , which are heavily bottlenecked in that order, is removed separately in distinct runs of the experiment. The removal of a receiver occurs well before the congestion is relieved: during the 2nd cycle of a 15-cycle convergence process. Here, the 'exponential decrease' part of AIMD algorithm accommodates the session reconfiguration by ignoring the ELRs from R_2/R_5 , as the case may be, in the loss-averaging process. The final send rate of V_a-V_e now increases to 51.63 *fps* and 53.39 *fps* respectively.

Likewise, we had R_2 and R_5 voluntarily depart from the multicast session in distinct runs of the experiment. A receiver departure occurs after the congestion has been relieved at the end of 14-15 control cycles. Here, the 'additive increase' part of AIMD algorithm starts the session reconfiguration after pruning the path segment of multicast tree connecting to R_2/R_5 , as the case may be. The final send rate of V_a-V_e now increases to 51.45 *fps* and 53.67 *fps* respectively.

As can be seen, the final send rate does not change between a voluntary departure of R_2/R_5 and a congestion-induced departure of R_2/R_5 . The departure of R_5 however causes a higher increase in the final send rate than that of R_2 , since the link connecting to R_5 has a more severe bottleneck than that to R_2 .

The experimental study demonstrates the ability of our rate-control algorithm to reconfigure the multicast session in the presence of dynamic changes in the population of devices serviced by various receiver-proxy nodes — as is the case with mobile wireless end-users.

VIII. CONCLUSIONS

We study the multi-source joint rate control methodology, as a constrained utility optimization problem. Our idea is to employ heuristics-based search of the rate control space to determine a reasonably optimal solution with low convergence latency, while taking cognizance of session-level QoE aspects. The QoE consideration stems from the fairness in rate allocation among sources and the isolation among receivers needed in the face of disparity in their capabilities. The paper described a receiver-proxy based end-system architecture for QoE enforcement.

REFERENCES

- [1] A. Adams, and et al. **The Use of End-to-End Multicast Measurements for Characterizing Network Internal behavior.** In proc. *MINI-COMM'00*, Jan.2000.
- [2] J. Vieron, T. Turlitti, K. Salamatian, and C. Guillemot. **Source and Channel Adaptive Rate Control for Multicast Layered Video Transmission based on a Clustering Algorithm.** in *Eurasip Journal on Applied Signal Processing*, pp.158-175, Jan. 2004.
- [3] P. Thapliyal, Siddhartha, Jiang Li, and S. Kalyanaraman. **LE-SBBC: Loss-event Oriented Source-based Multicast Congestion Control.** *Journal of Multimedia Tools and Applications*, Springer publ. Oct.2004.
- [4] Yatin Dilip Chawathe. **Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service.** In *Ph.D. Thesis*, University of California (Berkeley), Dec. 2000.
- [5] Yang-hua Chu and Sanjay G. Rao and Srinivasan Seshan and Hui Zhang. **Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture.** In proc. *Communication Architectures, Protocols, and Applications*, ACM SIGCOMM01, Aug. 2001.
- [6] M. Venkataraman, Member and M. Chatterjee. **Quantifying Video-QoE Degradations of Internet Links.** In *IEEE-ACM Transactions on Networking*, (20)2, April 2012.
- [7] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang. **Network Routing Topology Inference from End-to-End Measurements.** in proc. *IEEE INFOCOM'08*, 2008.
- [8] N. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. **Multicast Topology Inference from Measured End-to-End Loss.** In *IEEE Transactions on Information Theory*, vol.48, no.1, pp.26-45, Jan. 2002.
- [9] M. Rabby, K. Ravindran, and Jun Wu. **Distributed Adaptation Algorithms for Rate-controlled Video Multicast over Shared Infrastructure Networks.** In proc. Intl. Conf. on *Communication Systems and Networks (COMSNETS)*, IEEE-COMSOC, ACM-SIGMOB, Bangalore (India), Jan.2010.

- [10] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. **Towards Network-wide QoE Fairness Using OpenFlow-assisted Adaptive Video Streaming**. In proc. FhMN'13 workshop, ACM-SIGCOMM, Hong Kong (China), 2013.
- [11] J. F. Kurose and K. W. Ross. **Principles of Reliable Data Transfer**. Chap. 3.4, *Computer Networking: a Top-Down Approach*, A-W Publ., 2008.
- [12] K. Lee, T. Kim, and V. Bharghavan. **A Comparison of End-to-End Congestion Control Algorithms: The case of AIMD and AIPD**. in proc. *IEEE INFOCOM'01*, 2001.
- [13] P. Papadimitriou and V. Tsaoussiddis. **A Quality Adaptation Scheme for Internet Video Streams**. in proc. 5th Intl. conf. on *Wired/Wireless Internet Communications*, LNCS 4517 (Springer-Verlag), 2007.
- [14] M. Siller and J. Woods. **Improving Quality of Experience for Multimedia Services by QoS Arbitration on a QoE Framework**. In Proc. of the 13th Packet Video Workshop 2003.
- [15] K. R. Laghari, N. Crespi, B. Molina, and C.E. Palau. **QoE aware Service Delivery in Distributed Environment**. In proc. workshop on *Advanced Information Networking and Applications (WAINA2011)*, pp.837-842, March 2011.