Stability of Video Rate Control Algorithms Over Bandwidth-limited Network Paths

Kaliappa Ravindran & Arun Adiththan
City University of New York (CUNY-GSUC & City College)
New York, NY 10019.
Email: *ravi@cs.ccny.cuny.edu; arunadiththan@gmail.com*

Supratik Mukhopadhyay
Louisiana State University
Baton Rouge, LA 70803, USA
Email: *supratik@csc.lsu.edu*

***Abstract*—**The paper considers the stability of AIMD-based rate adaptation algorithm for video transmission over a bandwidth-limited network path. We examine the stability aspects from two angles: first, a quick relief from congestion by a faster reduction of send rate, and second, a larger limit-cycle time with a slow rate reduction in the steady-state (and hence lower rate jitter). The stability assessment also considers the loss of feedback signals from the video receiver that indicate the packet loss on data channels. We provide a management-oriented assessment of the convergence properties of video rate control system. This in turn allows an autonomic adjustment of the AIMD parameters and the system controller models for optimal performance.

I. INTRODUCTION

We consider a rate adaptive end-to-end video transport system (BAVT) realized over a bandwidth-limited network path as seen through a prism of control-theoretic management of network systems. The network setting may possibly be a UDP-IP transport connection or a MPLS-based path over IP, which does not allow direct bandwidth reservations[1]. Figure 1 shows the architecture of a BAVT, with a delineation of the core adaptation functions to adjust the send rate and the augmented system management functions that exercise an external control. This paper focuses on a control-theoretic management of the BAVT functionality from the perspective of end-user QoE (quality of experience).

The intrinsic complexity of BAVT system arises from two estimation problems: the available bandwidth along the path, and the bandwidth demand of a video data flow exercised on the path. Often, the packet flow from a video source is bursty, making its bandwidth demand highly unpredictable. If multiple video sources share the path, the statistical multiplexing effects also impact the bandwidth use. The lack of knowledge about bandwidth use is compounded by a presence of cross-traffic along the path, thereby making the available bandwidth B^*_{av} difficult to estimate [1], [2]. To get around this knowledge uncertainty, the receiver end-point signals the packet loss information (ELR) to the source end-point for the latter to carry out a suitable control action: rate increase or decrease, as the case may be.

[1]Even 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.

978-3-901882-66-1

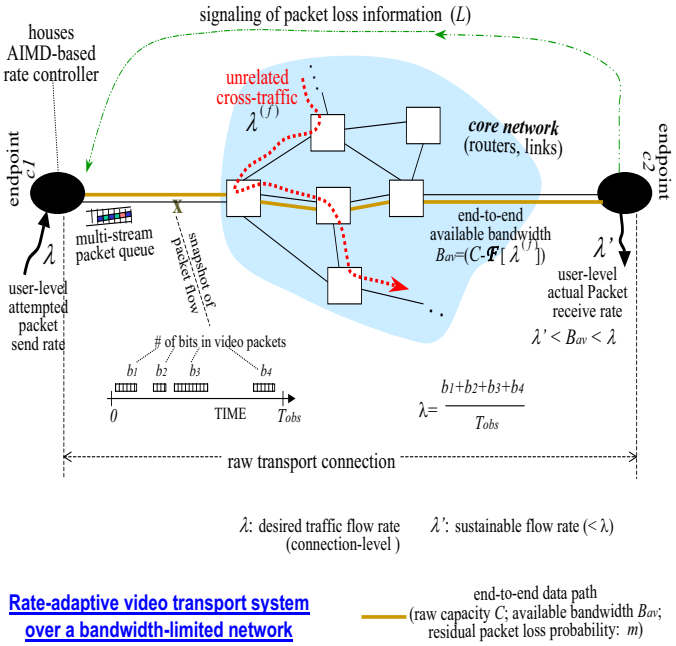


Fig. 1. Management view of bandwidth-adaptive video transport system

If λ is the rate at which the video source attempts to send packets, we denote the bandwidth demand of the flow as: $\mathcal{F}^*(\lambda, [m, \rho, \cdots])$ — where m is the residual packet loss rate on the link and ρ depicts the burstiness of packet flow. With flow aggregation over the data path, ρ depends on the number flows admitted N and the traffic smoothing exercised on flows[2] [3]. Since neither B_{av} nor $\mathcal{F}^*$ is known in an exact form, the adaptation condition: $\mathcal{F}^*(\lambda', [m, \rho]) < B_{av}$, is determined by model-based control — where λ' is the sustainable send rate such that $\lambda_{(\min)} < \lambda' < \lambda$.

In the absence of precise knowledge about B^*_{av} and and $\mathcal{F}^*$, networking researchers have employed the AIMD algorithm that employs an *additive increase multiplicative decrease* strategy for video rate adaptation [4], [5]. We consider the AIMD running at the aggregation point of multiple video flows from end-users (instead of at the user devices). The AIMD algorithm

[2]A measure of the burstiness of a flow is expressible in terms of the peak and average rates: $\rho = [1 - \frac{\lambda_{(a)}}{\lambda_{(p)}}]$, with $\rho = 0$ depicting a CBR flow (e.g., music) and $\rho \to 1.0(-)$ depicting a highly bursty flow.

works reasonably well to get around the difficulties posed by the uncertainty in knowledge, *albeit*, with certain limitations that affect the QoE. Basically, the AIMD algorithm is unable to distinguish between the scenarios of a congestion in the network and a marginal over-rate allocation, as it searches for an optimal send rate of video: namely, lower the rate difference $(\lambda - \lambda')$. Consequently, AIMD does not have good handle on the convergence latency during a congestion relief, and the rate jitter incurred in the steady-state after congestion relief. This intrinsic characteristic of AIMD algorithm may manifest as instability in the rate control behavior of BAVT system: as depicted by large rate oscillations.

Furthermore, the loss of ELR messages on the signaling channel may also induce instability. For instance, the loss/delay of an ELR message may cause the source to believe that there is no congestion, and hence increase the frame rate instead of reducing the rate. If the ELR loss/delay persists over long durations, the ensuing incorrect source-level actions may exacerbate the instability issues. In this context, existing works on network stability analysis [15], [16] have assumed that a base system is intrinsically stable: which allows an incremental treatment of the stability problem by modeling the ELR message loss as an additive external noise. In contrast, the BAVT system we consider can become unstable under incorrectly set AIMD parameters: even in a scenario of no loss/delay of ELR messages.

In this paper, we provide extensions to AIMD in the form of dynamic plug-in of the rate adjustment parameters, namely, how fast or slow the rate should be reduced or increased. The extended AIMD, armed with a system-level support mechanism for dynamic parameter adjustments, improves the overall QoE of end-users when reacting to congestion. The QoE improvement is however contingent upon a stable behavior of the BAVT system. In this light, we provide a stability analysis of the BAVT system with the aid of control-theoretic techniques. Our analysis identifies the equivalent linear models of AIMD algorithm, whereupon the algorithm parameters can be determined to achieve a faster congestion relief and a lower post-relief limit-cycle frequency. Experimental results are given to demonstrate the viability of our extended AIMD which is provably stable (in a control-theoretic sense).

The paper is organized as follows. Sections II-III provide a model-based treatment of the BAVT system anchored on a black-box programming view of the underlying transport connection. Section IV describes the QoE aspects of video rate adaptation. Section V discusses the effects of "loss report" errors arising in the signaling channel. Section VI discusses related works from a network management perspective. Section VII concludes the paper.

II. A MODEL-BASED VIEW OF BAVT SYSTEM

A model-based description of the raw transport sub-system embodied in the BAVT system may be given as:

$$L^* = (1 - \frac{B^*_{av}}{\mathcal{F}^*(\lambda, [m, \rho, \cdots])}), \quad (1)$$

where L^* is the packet loss rate experienced due to congestion ($L^* > m$). In the model[3], B^*_{av} constitutes the internal state of end-to-end transport connection that forms the core of BAVT system and L^* depicts the system output result of sending video data over the connection.

In the absence of exact knowledge about $\mathcal{F}^*(\cdots)$ and/or B^*_{av}, an 'additive increase multiplicative decrease' (AIMD) algorithm [4], [5] is employed as a coarse modeling approximation to adjust the send rate to λ' matching the available bandwidth B^*_{av}. The AIMD steps are:

- Set the new send rate as: $\lambda^{(\text{new})} = \lambda^{(\text{cur})} \times exp(-\beta L)$ if $L > \delta_h$, where L is the observed end-to-end loss ratio and δ_h is a threshold to indicate congestion, and $\beta > 0$;
- Set the new send rate as: $\lambda^{(\text{new})} = \lambda^{(\text{cur})} + \alpha$ when $L < \delta_l$, where δ_l is a loss threshold below which congestion is deemed to be absent ($0 < \delta_l < \delta_h$), and $\alpha > 0$.

$L \in [\delta_l, \delta_h]$ depicts a stable operating region for the BAVT system, with the send rate λ' exhibiting a limit-cycle as determined by the hysteresis parameter $(\delta_h - \delta_l)$ — where δ_l and δ_h are small loss percentages: say, 0.75% and 1% respectively. Note that a video packet delayed by longer than a threshold D is deemed as lost. Figure 2 shows the algorithmic schema of AIMD-based rate adaptation procedure. The[4] AIMD parameters β/α determine how fast the send rate can be slowed down or ramped up without causing a jitter in the user-level perceptual QoS. The control iterations are separated by a time interval determined by τ_{conn}: the time-constant of raw transport connection, i.e., the time needed for a rate reduction/increase to have an observable impact on the bandwidth use (say, for the existing packets in router queues to trickle out and for reporting the change in packet loss) [6].

Whether a rate reduction relieves the congestion or not can be ascertained only by exercising the rate reduction on the transport connection and observing the end-to-end packet loss rate L therein. If $L > \delta_h$, the under-reduction in send rate is compensated in a subsequent rate reduction. The observe-and-adapt cycle continues over multiple iterations q until the congestion is relieved, i.e., $L < \delta_l$. The total time to adapt: $T_{c_{\text{AIMD}}} \approx q\tau_{conn}|_{q \geq 1}$, depicts the convergence latency of rate adaptation process (i.e., the duration of a *control round*). How the controller programmatically manages the system convergence process is described next.

III. BLACK-BOX VIEW OF TRANSPORT CONNECTION

The transport connection is abstracted as a computational function of the form: $L = \text{net}(\lambda)$, where $\lambda > 0$ and $L > 0$. The internal details of $\text{net}(\cdots)$ to determine L for an input λ are not known to the programmer of network control, i.e., $\text{net}(\lambda)$ appears as a black-box taking λ as the input and

[3]As a convention, we use '$*$' to denote the true value of a system parameter or function. Determining the true value is however subject to observational and/or modeling errors (due to dimensionality issues). A parameter or function without '$*$' denotes the actual value made available to the controller module with the use of finite resources by the underlying computation.

[4]The hysteresis parameter $(\delta_h - \delta_l)$ prevents ping-pong effects on the rate control when the external conditions change fast.

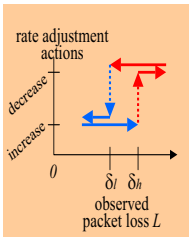
Iterative procedure for rate adjustment

Additive Increase Multiplicative Decrease (AIMD)

In each 'loss reporting' interval

$\lambda(new) = \lambda(cur) - \beta.L$ when $L > \delta_h$, where $\beta > 0$

$\lambda(new) = \lambda(cur) + \alpha$ when $L < \delta_l$, where $\alpha > 0$



L: observed packet loss rate

δ_l, δ_h: Acceptable loss thresholds

[$\delta_l < \delta_h$ to avoid ping-pong effect]

^^ Each execution of this procedure constitutes a "**control iteration**"

^^ A sequence iterations that lead to a steady-state in bandwidth allocation (when the flow specs changes or a set of new flows are admitted) constitutes a "**control round**"

Fig. 2. Schema of AIMD-based rate adaptation procedure in a BAVT system

returning L as the output. The programmer however knows how the function $\mathrm{net}(\cdots)$ behaves when λ changes, as given by the monotonicity relationship:

$$\mathrm{net}(\lambda + \Delta) > \mathrm{net}(\lambda) > \mathrm{net}(\lambda - \Delta) \tag{2}$$

for $\Delta > 0$. In other words, an increase in λ leads to a higher L, and a decrease in λ leads to a lower L.

Suppose λ_0 is an initial input for which $\mathrm{net}(\lambda_0)$ returns a value $L_0 > \delta_h$. We develop an algorithmic procedure that reduces λ in multiple steps of $(\beta \times L)$ decrements to a value λ_f such that $\mathrm{net}(\lambda_f) < \delta_l$. When $L < \delta_l$, the procedure increases λ in multiple steps of a constant increment α until reaching a rate $\lambda_{f'}$ such that $\mathrm{net}(\lambda_{f'}) > \delta_h$. Thereupon, the procedure switches back to the rate reduction cycle. The sustainable send rate is given by: $\lambda' = \frac{\lambda_f + \lambda_{f'}}{2}$, where $0 < \lambda' < \lambda$. The repetitive computational steps that realize the AIMD procedure (i.e., control iterations) look like the following:

```
L := net(λ)      /* Initial output L > δh */
λ(g) := λ;
state := CONG;
while (true)
    if (state = CONG)   /* path is congested */
        lth := δl;
    else
        lth := δh;   /* Recall that 0 < δl < δh */
    if (L ≥ lth)   /* loop 1: multiplicative decrease */
        λ(g) := λ(g) − βL;
        state := CONG;
    else   /* loop 2: additive increase */
        λ(g) := λ(g) + α;
        state := ¬ CONG;
    L := net(λ(g));
    if (steady_state is reached)
        break;
λ' := λ(g).
```

Each iteration of "**while**" loop 1 realizes the multiplicative decrease phase of AIMD rule and that of "**while**" loop 2 realizes the additive increase phase, with a time-passage of τ_{conn} per iteration. Given the monotonicity property of $\mathrm{net}(\cdots)$ function, the AIMD procedure is guaranteed to terminate, i.e., find λ' in a finite number of iterations.

The AIMD procedure gets invoked upon the rate controller module receiving signal from the observer module to notify congestion: i.e., when the condition $L > \delta_h$ is sensed. This may occur due to a depletion in the available bandwidth of network B^*_{av} or an increase in the send rate emanating from video sources λ. The AIMD procedure however cannot distinguish between the loss condition induced by an asynchronously occurring fluctuation in B^*_{av} versus an over-increase in $\lambda_{(g)}$ in its attempt to maximmize the use of B^*_{av}. This imprecision of knowledge when executing the rate reduction ("**while**" loop 1), combined with the propensity to search for a higher send rate when $L < \delta_l$ ("**while**" loop 2), thus leads to a continuous execution of the AIMD procedure even in the steady-state.

IV. QoE ASPECTS OF VIDEO RATE ADAPTATION

The QoE of BAVT systems depends on multiple parameters that capture the perceptual QoS experienced by a video flow by virtue of the system's adaptation functionality:

$$[(\lambda, \lambda'), T_{c_{\mathrm{AIMD}}}, \frac{d\lambda'}{dt}, (\Delta\lambda', T_l)],$$

where λ is the desired send rate, λ' is the sustainable send rate, $\Delta\lambda'$ is the rate swing in the steady-state and T_l is the corresponding limit-cycle time, $\frac{d\lambda'}{dt}$ depicts how fast the rate fluctuates in the limit-cycle region, and $T_{c_{\mathrm{AIMD}}}$ is the rate convergence time to effect a congestion relief (i.e., the time to move from a congested state to the steady-state) using the AIMD rule. Rate jitter is a key QoE parameter associated with the steady-state behavior, and may be captured in terms of the range and frequency of rate fluctuations $[\frac{d\lambda'}{dt}, (\Delta\lambda', T_l)]$. The convergence latency $T_{c_{\mathrm{AIMD}}}$ depicts how responsive is the system to a congestion, and hence is an indication of QoE.

A. QoE-aware setting of adaptation triggers

Given a video transport session, the AIMD-based procedures housed in the controller compute an adjustment of video send rate λ'. The video delivery point at receiver site is an observable interface to the BAVT system, which feeds the input for an algorithmic computation of control actions by the controller agent. The adaptation goal is to keep the video rate perception quality experienced by a human viewer, i.e., QoE, within acceptable levels. The safety constraint of C_P, as pertaining to the QoS monitored at the transport service layer, is prescribed as:

$$|\frac{d\lambda'}{dt}| < R_0 \wedge \lambda' > \lambda^{(min_1)}, \tag{3}$$

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

$$|\frac{d\lambda'}{dt}| \leq r_0 \wedge \lambda' \geq \lambda^{(min_2)}, \quad (4)$$

where r_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 QoE aspects, we program the AIMD algorithm to generate a control trajectory that spreads out the rate reduction over multiple rounds q, where $1 \leq q \leq q_m$. 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 $q_m \times T_m$ depicts the maximum allowed system inertia with human-in-the-loop. Our QoE characterization is different from the network-oriented measures described in [18].

The constraint (4) depicts the domain-knowledge made available to the video transport 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 (3) purports to define the safe execution paths of controller, and guide the algorithm-internal operations therein. In this light, a setting of R_0 and $\lambda^{(min_1)}$ such that $R_0 > r_0$ 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 the controller detects the safety violation $|\frac{d\lambda'}{dt}| \geq R_0$ or $\lambda' \leq \lambda^{(min_1)}$, the cases: $|\frac{d\lambda'}{dt}| \in [R_0, r_0]$ and $\lambda' \in [\lambda^{(min_2)}, \lambda^{(min_1)}]$ depict the onset of symptoms indicative of the possibility of rate jitter or rate depletion becoming noticeable to the human user. The safety margins $(r_0 - R_0)$ and $\lambda^{(min_1)} - \lambda^{(min_2)}$ indicate how soon the controller triggers a reduction of λ', to avoid QoE degradations. See Figure 3 for an illustration. The points Y and Z depict safety violations at both levels, i.e., the constraints (3) and (4) 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 in the controller but does not cause a safety violation at user-level. A controller capability 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.

B. System stability as an instance of QoE

Our main idea in QoE considerations 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. We evaluate how good the rate-control system is in reducing the video send rate of source(s) when congestion occurs along the network path. Here, the rate-control functions to effect congestion relief, as realized by the source and receiver agents, primarily consist of the reporting of ELRs, inferencing of path properties (such as available bandwidth

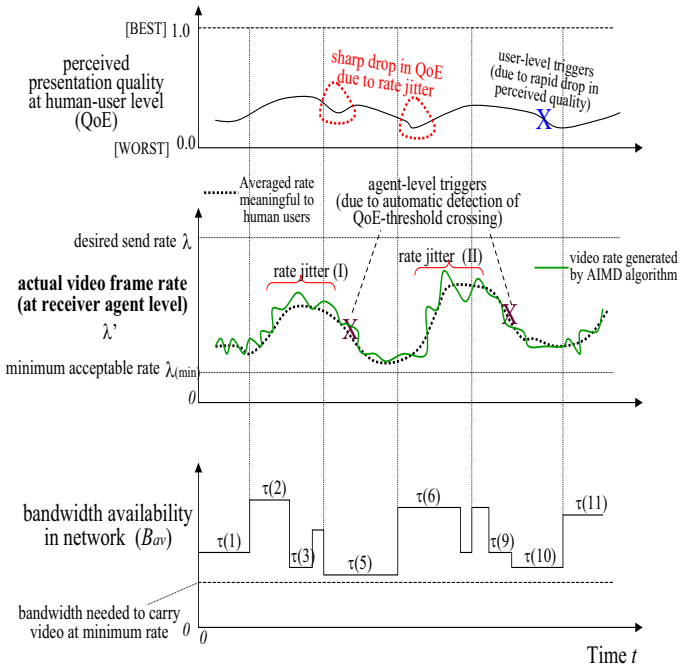


Fig. 3. QoE-aware adaptation triggers based on time-scale separation

B^*_{av}), 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 rate of video source(s). QoS is quantified in terms of parameters meaningful to video source & receiver: such as how fast an adaptation to packet loss conditions occurs, stability & convergence of send rate, and how smooth is the changes in send rate. 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 BAVT system in various dimensions of interaction with the human ergonomic aspects. 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 application session — c.f. Figure 3. The end-user quality perception is however a qualitative factor[5]. 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 a high system-level QoS at the end-point housed in the network edge during rate adaptation.

[5]The 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.

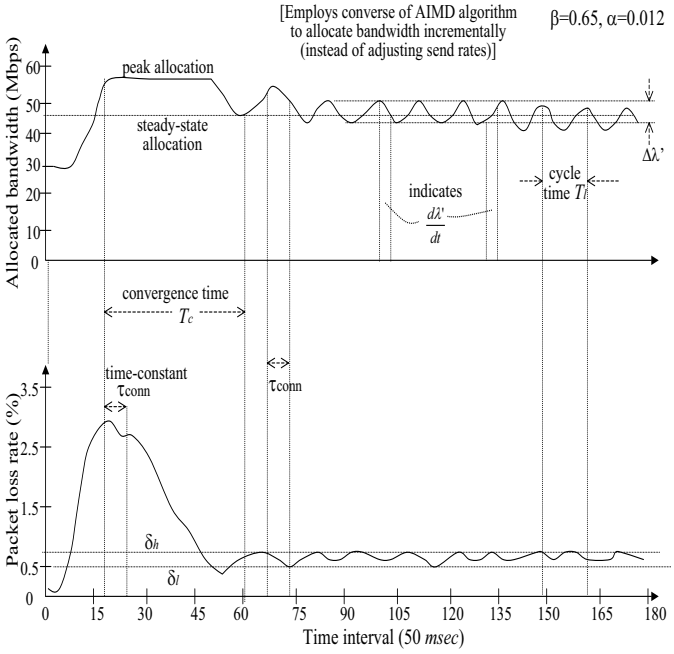


Fig. 4. Adaptation behavior of BAVT system: $\beta = 0.65, \alpha = 0.012$

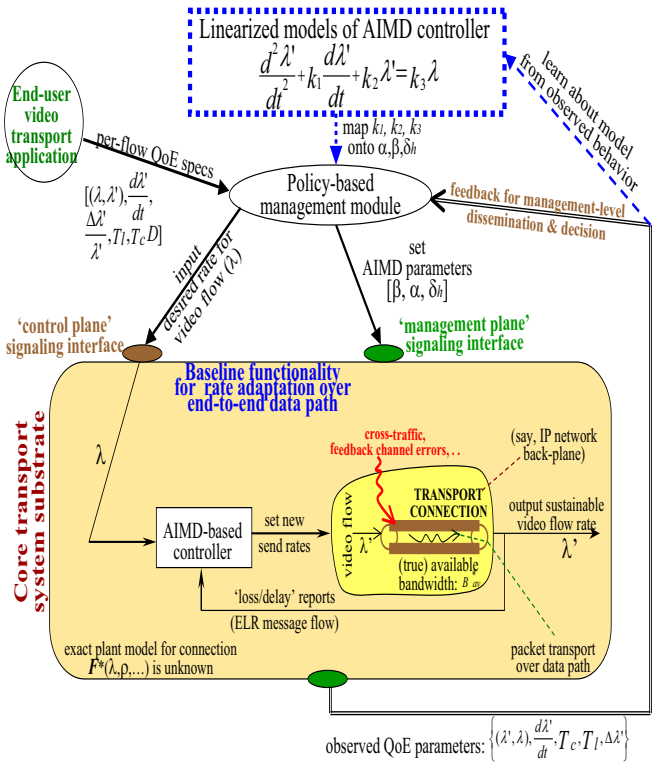


Fig. 5. Model-based supervision of rate adaptation in BAVT system

C. Adaptive setting of AIMD coefficients (empirical study)

We consider the control steps in AIMD algorithm invoke the $\mathrm{net}(\lambda)$ function. Suppose λ_0 is an initial input for which $\mathrm{net}(\lambda_0)$ returns a value $L_0 > \delta_h$, where $\delta_h > 0$. In that case, the algorithm reduces λ in multiple steps of $(\beta \times L)$ decrements to a value λ_f such that $\mathrm{net}(\lambda_f) < \delta_l$, where $\beta > 0$ and $0 < \delta_l < \delta_h$. Thereupon, the algorithm increases λ in multiple steps of α increments to a value λ_t such that $\mathrm{net}(\lambda_t) > \delta_h$, where $\alpha > 0$. Thereafter, the decrease procedure kicks in again. Thus, there is a repetitive cycle of decrease and increases of λ.

A study of how (L, λ) varies with respect to i for each of the cases: i) $\beta > \beta', \alpha = \alpha'$; ii) $\beta = \beta', \alpha > \alpha'$; iii) $\beta > \beta', \alpha > \alpha'$; and iv) $\beta < \beta', \alpha < \alpha'$ gives an insight on the QoE impact of AIMD parameters.

Figure 4 shows the experimental results for a converse version of AIMD algorithm that allocates/de-allocates bandwidth in incremental steps to sustain a given flow rate (see [6]). In this converse version, the roles of β and α respectively are to determine the additional bandwidth allocation needed to counter an observed packet loss and the bandwidth de-allocation margin in a case of no packet loss. An initial traffic load of 10 video flows (from *Jurassic Park* clips) for an allocation of 30 $mpbs$ was increased to 15 flows, with the loss-tolerance set as: $\delta_l = 0.05\%$ and $\delta_h = 0.08\%$. The final steady-state allocation determined by the algorithm is in the range 45-46 $mbps$. Results on the adaptation performance in reaching the steady-state illuminate two points of interest. First, the rate convergence time T_c varies with the β-parameter: $T_c = 3.25\ sec$ for $(\beta = 0.55, \alpha = 0.012)$ and $T_c = 2.1\ sec$ for $(\beta = 0.65, \alpha = 0.012)$. Second, the limit-cycle time in the steady-state are about $0.9\ sec$ and $0.75\ sec$ respectively. As can be seen, a formulation of QoE parameters involves a complex mapping of QoS-to-QoE spaces.

A stable operation of AIMD requires the ability to detect system convergence to a steady-state: say, based on the occurrence of a certain minimum number of limit-cycles. A predicate-based detection of steady-state may be employed [7]: which involves specifying one or more boolean conditions on the BAVT system output that are deemed to hold in a steady-state. The controller evaluates the predicate in order to detect if the system has reached a steady-state.

D. Supervised control of QoE in BAVT system

The BAVT system exhibits a non-linear relationship between B^*_{av} and the QoE parameters: rate convergence and rate jitter. The ability to dynamically adjust the AIMD parameters as the BAVT operating region changes is thus desirable to improve the human-oriented QoS aspects (i.e., QoE). The level of rate sustainability $(\lambda - \lambda')$ for a flow is a QoE metric: namely, how much the system can offer relative to the flow's expectation.

Figure 5 shows the parametric interactions between the BAVT system management module and the rate adaptation control module vis-a-vis QoE control, and the underlying adaptation elements in the core parts of BAVT system. The degree of control exercised on the rate adaptation module may vary from one system to another. But the core adaptation functions are fixed, with a suitable management and signaling interface to make them *programmable* (in a functional sense) by having an external module supply the QoE-oriented user-level parameters and algorithm-oriented AIMD parameters.

A *learning module* approximates the BAVT system with linear models to enable a control-theoretic analysis of system stability, whereupon the model parameters are mapped onto the AIMD coefficients $[\alpha, \beta, \delta_h]$. The learning is driven by an analysis of the desired rate adjustment profiles.

V. Impact of 'loss feedback' signaling errors

The adaptation performance of BAVT system may also be affected by the quality of end-to-end 'loss reports' signaled by the receiver. If a 'loss report' (ELR) is lost in the signaling network, it causes a missed/delayed rate adjustment by one or more sources. Such a missed action is compensated for in some way by the other sources acting upon the 'loss report' (if they receive it). The lack of common knowledge about missed ELR messages and the source-level action inconsistency arising therefrom may lead to an uneven split of rate reduction among the sources participating in a congestion relief.

Given that the rate adjustment by each of the N sources is orchestrated in a distributed manner without any explicit coordination among them, the missed actions at one or more sources $\{y\}$ manifests as a less-than-planned rate change. The error however is undetectable for the controller due to the lack of knowledge about B^*_{av} and/or $F^*(\lambda_y, \cdots)$. From a modeling standpoint, the error may simply be deemed as a noise-induced annulment of the rate changes that the missed ELR would have otherwise caused at $\{y\}$. We conducted experiments to demonstrate that the rate adaptation process is however resilient against the loss/delay of the signaled ELR messages. Given a lossy behavior of the signaling channel, we consider two cases: i) the lost ELR messages are recovered by retransmissions; and ii) no ELR recovery is carried out. The rate convergence behavior for the cases (i) and (ii) are then compared to reason about the BAVT system resilience.

Given an initial send rate of 25 fps each by V_a, V_b, V_c, we injected packet loss in the signaling network at 5%, 7%, and 10%. For each loss probability, we compared the case of ELR loss recovery with a case of no recovery. Figure 6 shows the ensuing changes in convergence behavior during a congestion relief (the limit-cycles are not shown, to improve clarity). The difference between the two cases arises only in the rate convergence phase, but the final send rate is the same at 6.5 fps regardless. The reason is that signaling channel errors are benign, affecting the various flows in an indiscriminate manner over a large time-scale. So, in a statistical sense, the cumulative average of the impact of signaling errors on various flows is zeroed out over a large time-scale (though short-term impacts on QoE may be present).

In a case of loss recovery of ELR messages, it took (8.5,6.5,8.75) observe-adapt cycles for (V_a, V_b, V_c) to reach the steady-state final rate when $v = 5\%$; (9.5,7.5,10) cycles when $v = 7\%$, and (11.5,9.5,12) cycles when $v = 10\%$. In a case of no ELR recovery, it took (17,15,19) cycles, (18.5,15.25,20) cycles, and (21,15.75,22) cycles respectively[6]. As can be seen, the case of no ELR recovery leads to a high T_c: almost double that of the case of ELR recovery. The reason is that, given an asynchronous execution of the ELR delivery and the rate adjustment processes, a lost ELR message causes the missing of a rate adjustment cycle; whereas, the protocol-level recovery-induced delay in ELR message delivery (and hence the increased τ_{conn}) only causes a small lateness in the rate adjustment action[7]. That a missed action is more harmful than a delayed action is reflected in the higher T_c for the case of no ELR recovery.

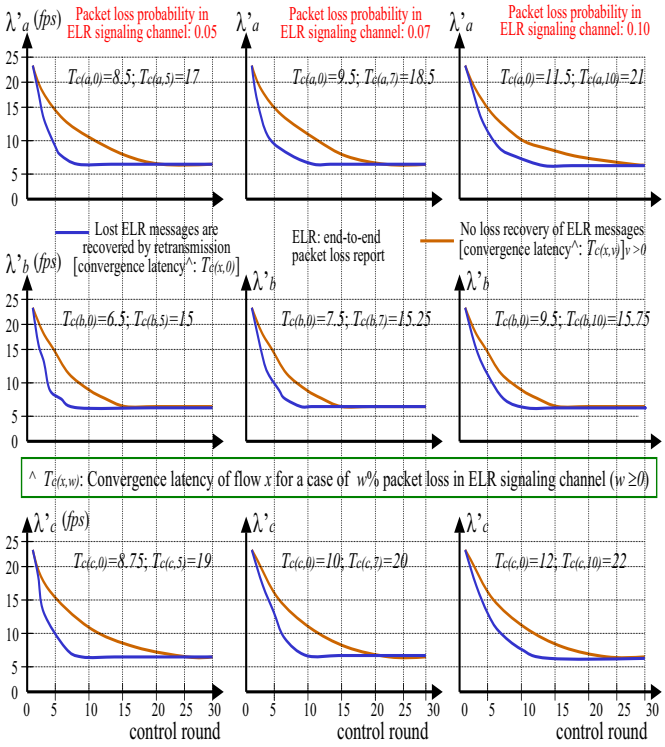


Fig. 6. Impact of ELR signaling errors on convergence behavior of sources

Furthermore, the increase in T_c depicts a monotonic convex behavior with respect to v in both the cases:

$$\frac{dT_c}{dv}|_{v=v_1} < \frac{dT_c}{dv}|_{v=v_2} \quad \text{for} \quad v_2 > v_1.$$

With loss recovery of ELR, (V_a, V_b, V_c) experience an increase in T_c by $(1.0, 1.0, 1.25)$ cycles When v increases from 5%

[6]Differences in the convergence behavior of sources V_a, V_b, V_c arise due to scenarios of dissimilar bootleneck segments in the access network before their flows reach the transport connection maintained in the core network. So, the difference in their T_c is not an indication of any session-level differentiated rate scheduling.

[7]ELR recovery protocol often involves a coordination between the video receiver and the video source (e.g., 3-way hand-shake) to effect an ELR delivery — and even among the multiple sources in cases where mutual action consistency is needed [8]. The coordination manifests as a protocol-level delay in delivering an ELR message to the video source (even when the native signaling channel does not lose ELR messages). This is in contrast from the stateless nature of best-effort delivery mechanisms in a case of no support for ELR recovery in the signaling channel: here, an ELR may be lost but the delivery of a received ELR is never delayed.

to 7%; and, the increase in T_c is $(2.0, 2.0, 2.0)$ cycles When v increases from 7% to 10%. This depicts a non-linear increase of T_c, with the marginal change $\frac{dT_c}{dv}$ observed as $(0.5, 0.5, 0.63)$ at $v = 5\%$ increasing to $(0.67, 0.67, 0.67)$ at $v = 7\%$. With no ELR recovery on the other hand, the corresponding increases in T_c are $(1.5, 0.25, 1)$ cycles and $(2.5, 0.5, 2.0)$ cycles respectively, with $\frac{dT_c}{dv}$ increasing from $(0.75, 0.13, 0.5)$ at $v = 5\%$ to $(0.82, 0.17, 0.82)$ at $v = 7\%$. This shows that the behavior of a network system degrades exponentially as the network loss condition increases.

The marginal change relative to T_c, i.e., $\frac{\frac{dT_c}{dv}}{T_c}$, is smaller for the case of no ELR recovery. This depicts a lower sensitivity to fluctuations in the loss behavior of ELR signaling channel, *albeit*, at a degraded convergence behavior (when compared to a case of ELR recovery support).

VI. Existing works on complex network systems

Network systems embody both adaptation behaviors and functional properties. The former, as focused in our paper, pertains to adjusting the system operations according to the environment conditions. With a separation of adaptation behaviors from functional correctness aspects, we focused on AIMD algorithms that reduce the video send rate to deal with bandwidth congestion in a network path, in the presence of uncertainty in knowledge about the path characteristics.

Some existing works deal with systems engineering for the control-theoretic aspects of adaptation: such as stability and convergence [9], [10]. In an earlier work [11], we explored sub-system level separation as a software engineering concept, with specific application to AIMD-based video rate adaptation. System-level tools exist to aid such studies (e.g., probabilistic monitoring & analysis [12]). The need for re-usable control loops across different domains is identified in [13]. The work [14] provides QoS-aware network service composition and adaptation to infuse a core self-management intelligence for autonomic operations of heterogeneous networks.

In the present work, we have employed the autonomic system management concepts to infuse a dynamic QoE-aware setting of the AIMD parameters. Our separation of the core rate adaptation functionality and its externalized management control can be employed in other application domains as well (such as autonomous vehicle navigation).

VII. Conclusions

The paper employed a model-based approach to video rate adaptation on a bandwidth-limited network path: such as the Internet. The intrinsic complexity of video rate adaptive system arises from two estimation problems: first, the available bandwidth along the path, and second, the bandwidth demand of a video data flow exercised on the path. The well-known AIMD (additive increase multiplicative decrease) algorithm works reasonably well to get around these difficulties, albeit, with certain limitations that affect the QoE (quality of end-user user experience).

In this paper, we provided extensions to AIMD in the form of dynamic plug-in of the rate adjustment parameters, namely, how fast or slow the rate should be reduced or increased. The extended AIMD, armed with a system-level support mechanism for dynamic parameter adjustments, improves the overall QoE of end-users when reacting to congestion. The paper described experimental results to demonstrate the viability of our extended AIMD.

Employing a software-engineering view for complex network systems, the paper covered the modeling and software-level issues in orchestrating harmonious interworking of the core functions and the management functions in a BAVT system.

References

[1] S. Keshav. A Control-Theoretic Approach to Flow Control. In proc. *ACM SIGCOMM*, 1991.

[2] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. In IEEE Network, 2003.

[3] C. Patridge. Traffic Shaping. In Chap. 6, *Gigabit Networking*, Addison-Wesley Publ., 1994.

[4] 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, Anchorage (AK), 2001.

[5] J. C. Bolot. Characterizing End-to-End Packet Delay and Loss in the Internet. In proc. *ACM SIGCOMM*, San Francisco (CA), 1993.

[6] M. Rabby and K. Ravindran. Online Control Techniques for Management of Shared Bandwidth in Multimedia Networks. In proc. IEEE/IFIP *MMNS'07*, San Jose (CA), Oct. 2007.

[7] J. P. Katoen and I. S. Zapreeva. Safe On-The-Fly Steady-State Detection for Time-Bounded Reachability. In proc. *QEST06*, 2006.

[8] M. Rabby, K. Ravindran, S. Mukhopadhyay, R. Bharadwaj, and G. Mangukiya. Control Plane Properties for Signaling in Loss-feedback based Video Rate Adaptation over Shared Multicast Paths. In proc. 2nd Intl. Conf on *Comm. Systems and Networks* (COMSNETS'10), IEEE-ACM, Bangalore (India), Jan. 2011.

[9] B. Li, K. Nahrstedt. A Control-based Middleware Framework for Quality of Service Adaptations. In *IEEE JSAC*, 17(9), Sept.1999.

[10] C. Lu, Y. Lu, T. F. Abdelzaher, J. A. Stankovic, S. H.. Son. Feedback Control Architecture and Design Methodology for Service Delay Guarantees in Web Servers. In *IEEE TPDS*, 17(7), Sept. 2006.

[11] K. Ravindran and M. Rabby. Cyber-Physical Systems Based Modeling of Adaptation Intelligence in Network Systems In proc. IEEE Intl. Conf on *Systems, Man, and Cybernetics* (SMC), Anchorage (AK), Oct. 2011.

[12] M. Brunner, D. Dudkowski, C. Mingardi, G. Nunzi. Probabilistic Decentralized Network Management. In proc. IEEE/IFIP Symp. on *Integrated Network Management* (IM-2009), June 2009.

[13] A. Goel, D. Steere, C. Pu, and J. Walpole. Adaptive Resource Management Via Modular Feedback Control. Tech. Report, Oregon Graduate Institute School of Science & Engineering, 1998.

[14] J. Xiao and R. Boutaba. QoS-Aware Service Composition and Adaptation in Autonomic Communication. In IEEE Journal on Selected Areas in Communications, 23(12), Dec. 2005.

[15] W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of Networked Control Systems. In IEEE Control Systems Magazine, pp.84-99, Febr. 2001.

[16] B. A. Sadjadi. Stability of Networked Control Systems in the Presence of Packet Losses. In proc. IEEE Conf. on Decision and Control, 2003.

[17] Y. Diao, J. L. Hellerstein, G. Kaiser, S. Parekh, and D. Phung. Self-Managing Systems: A Control Theory Foundation. In *IBM Research Report*, RC23374 (W0410-080), Oct.2004.

[18] M. Venkataraman and M. Chatterjee. Quantifying video-QoE degradations of internet links. In IEEE/ACM Transactions on Networking, 20(2), pp.396-407, April 2012.