Online Control Techniques for Management of Shared Bandwidth in Multimedia Networks

K. Ravindran and M. Rabby

City College of CUNY and Graduate Center,
Department of Computer Science,
Convent Avenue at 138th Street,
New York, NY 10031, USA
ravi@cs.ccny.cuny.edu, mfrabby@yahoo.com

Abstract. In this paper, we provide an online monitor and control approach for adaptive bandwidth allocations in a QoS-aware multimedia network system. Accurate management of bandwidth allocations with a goal of maximizing the revenues is quite complex, due to the interactions among various data flows that dynamically share the network bandwidth. So, we adopt a heuristics-aided control that iteratively adjusts the bandwidth allocation based on the observed packet loss rate and delays. In terms of 'control theory', the bandwidth allocation and the packet loss rate constitute the system input and output respectively, with the heuristics-based bandwidth adjustment strategies incorporated in a controller along the feedback loop. A 'control-theoretic' treatment of the QoS adaptation problem allows studying the stability and convergence properties of the QoS delivered to the applications, while maximizing the connectivity service provider's revenues.

1 Introduction

The end-to-end data transfer in distributed multimedia applications may be viewed as a series of packet-level transport activities using a network service that offers QoS-enabled data connectivity between the end-points. The service provider (CSP) uses the shared resource pools maintained as part of its infrastructure, namely, the network capacity C available from a physical line provider (PLP), in offering the required logical connectivity between end-points. See Figure 1. The underlying network may, in one extreme, consist of the FIFO routers in a bandwidth-oblivious Internet and, in the other extreme, be the proxy nodes serving as packet forwarding points using UDP-based transport layer links.

The goal of CSP is to maximize its revenues by exercising just-enough bandwidth allocations that suffice to match the QoS needs of applications. This bandwidth allocation problem has two twists:

- Fuzziness in the bandwidth needs of an application-level data flow due to the inability to precisely characterize the traffic parameters;

– Statistical sharing of the network capacity among many logical connections that attempts to reduce the bandwidth allocation.

Despite the above sources of inaccuracies in the bandwidth management process, the CSP needs to accomplish its revenue-oriented goal.

The management functions of the CSP employ 'feedback control' principles to overcome the problems that arise due to the fuzziness in the bandwidth estimation process. We employ certain widely applicable monotonicity properties of the bandwidth allocation process to derive our heuristics-based control strategies. For instance, an increase in the send rate of a bursty data flow by δ incurs an additional bandwidth allocation of δ' such that $0 < \delta' < \delta$. Based on such (macroscopic) intuitive properties, the management functions of the CSP may exercise bandwidth allocation control at various degrees of granularity and accuracy.

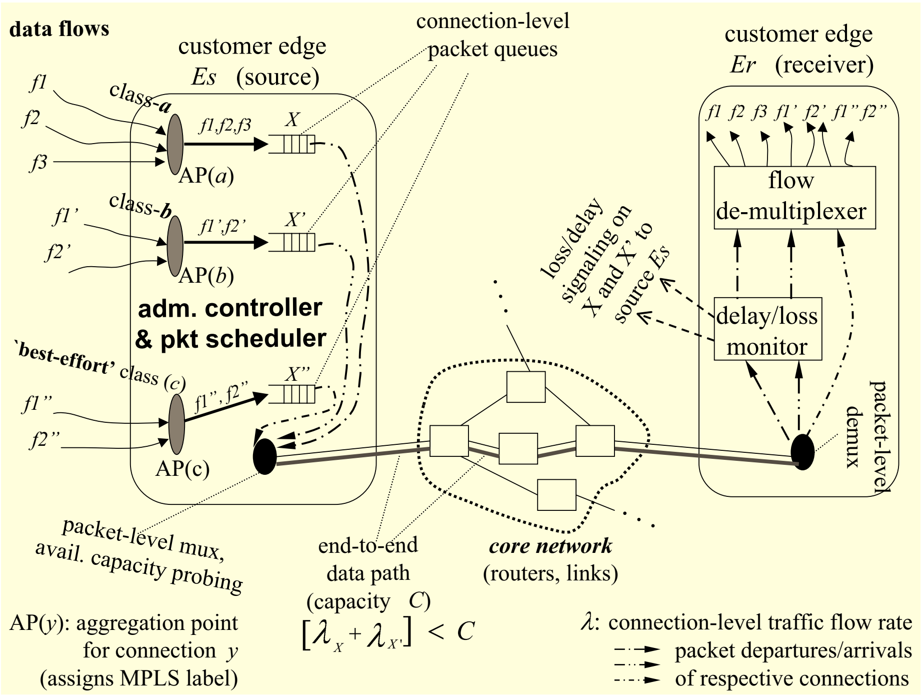

Fig. 1. System structure for edge-level control

The paper is organized as follows. Section 2 provides a management view of QoS-adaptive data transfers between multimedia end-points. Section 3 introduces the monitor-and-control based allocation of bandwidth. Section 4 discusses the system latency incurred during bandwidth allocation steps. Section 5 studies our monitor-and-control approach by simulation. Section 6 concludes the paper.

2 Management view of QoS-adaptive data transfers

The end-to-end connectivity service provider (CSP) allocates a portion B of the available capacity C for use in supporting data transfers that yield QoS-based revenues, where $B < C$. The left-over bandwidth $(C - B)$ is then taken by the non-revenue yielding 'best-effort connection' data. The bandwidth allocation B is itself realized by 'weighted fair queue' (WFQ) packet scheduling mechanisms, based on the ratio $\frac{B}{C}$.

Our paper focuses on the 'management plane' functions to determine the optimal value of B that can be suitably spread out across various data flows to meet the CSP's revenue objectives.

2.1 Flow-based classification of connections

We group the application-level data flows based on their traffic characteristics: such as the rate of packet arrivals, the burstiness in bandwidth demands, and the nature of service guarantees expected (say, deterministic or probabilistic). Here, a flow is a sequence of packet arrivals from an application that belong to a particular traffic group (or class). A data flow exercises the shared capacity maintained by the CSP's infrastructure in order to meet the QoS needs prescribed in a service-level agreement (SLA) — such as bounding the packet delivery latency. Widely dissimilar flows, such as video versus voice, may exercise different amount of demands on the underlying network bandwidth — and hence may fall under different traffic classes for bandwidth management purposes. All the data flows of a particular traffic group that emanate at an end-point (potentially from multiple applications) are bundled into a single *logical connection* for bandwidth management purposes. Refer to Figure 1. The grouping of closely-similar data flows is a 'control plane' function in our system architecture.

A characterization of packet arrivals based on traffic loads is useful for an effective management of the network bandwidth. It allows establishing a mapping between the QoS-oriented guarantees expected by the applications (as incorporated into SLAs) and their usage of network bandwidth (internal to the service infrastructure) [4–6]. Based on the estimated bandwidth needs, a packet scheduler at the sending end moves the packets of various flows waiting in the connection queues over the data path maintained by the network infrastructure.

2.2 Exploiting statistical bandwidth sharing gains

Potentially, multiple classes of packet traffic may share a common network infrastructure maintained by the CSP. Given the bursty and random nature of most packet transfers (e.g., video), the CSP incorporates the gains accrued from statistical bandwidth sharing in its QoS management decisions.

Typically, a bandwidth allocated for the flows belonging to a class i can be used by the flows belonging to a class j when i does not have any demands in a certain time interval but j has demands. A statistical resource sharing method interweaves with the QoS delivery to the applications by making the guarantees probabilistic (instead of a deterministic). An example is to keep the 90-percentile response time of an application flow within a given tolerance limit. Since a probabilistic quality suffices for most real-time multimedia applications, the CSP's revenue goals can be met by a careful control of the degree of statistical sharing of network bandwidth. Our grouping of data flows at end-points towards this goal aligns architecturally (along the 'control plane') with the hose model of VPNs studied elsewhere [3].

Though statistical bandwidth sharing by itself is not a new concept, how the gains accrued from such a sharing can be quantified for incorporation into the CSP's QoS management decisions has not been studied so far. To realize this revenue-driven goal, we describe an online monitor-and-control approach next.

3 Online control procedures for bandwidth allocation

The problem of optimal bandwidth allocation is quite complex for two reasons: i) the CSP's desire to allocate just enough bandwidth in the presence of statistical multiplexing gains among bursty data flows — such as video, and ii) the absence of precise QoS-to-bandwidth mapping relations due to incomplete traffic specs. We deal with the complexity issue by incremental bandwidth allocations over multiple heuristics-aided control iterations.

3.1 Mapping of flow specs to bandwidth needs

We may view the flow specs of a multimedia data, specified as a tuple [peak_rate,average_rate,loss_tolerance], as being mapped onto the required amount of bandwidth (measured in bits/sec *bps*) to transport the data over a connection. The bandwidth needs of a QoS-controlled data connection may be denoted as a transfer function:

$$\text{bandwidth needed} = \mathcal{F}([n,f],e),$$

where f is the flow specs (such as data send rate and loss tolerance) and n is the number of flows multiplexed over the connection. The parameter e pertains to the external environment that is not under the control of the end-point users — such as the burstiness of data and the residual loss rate on a transport link. If the allocated bandwidth $B_v > \mathcal{F}([n,f],e)$, there is no noticeable packet loss; otherwise, the (per-flow) packet loss rate is proportional to the amount of under-allocation of bandwidth, namely, $\mathcal{F}([n,f],e) - B_v$.

The function $\mathcal{F}$, which is bound to a logical connection, satisfies the property of weak additivity across multiple flows carried over this connection, i.e., $\mathcal{F}([n+\delta n,f],e) - \mathcal{F}([n,f],e) < \mathcal{F}([\delta n,f],e)$. This property depicts the statistical multiplexing gains accrued by having the flows share the connection bandwidth.

The mapping function $\mathcal{F}$ is only coarse and approximate, due to the random and bursty arrival of data and the inability of a traffic specs method to capture comprehensive data flow characteristics. So, the mapping function $\mathcal{F}$ may not be known to the end-point management functions. The absence of knowledge about the mapping relation $\mathcal{F}$ in a closed-form forces the management station to determine the bandwidth needs using end-to-end monitoring techniques. Any analytical formulation of $\mathcal{F}$ (or a tabular representation of $\mathcal{F}$) that may coarsely be known is used to enhance the accuracy of bandwidth estimator $\mathcal{F}$.

Another source of inaccuracy is the potentially large dimensionality of the traffic meta-data. This is compounded by the inherent limitations in a flow specs language chosen to describe the data traffic. In a way, most of the traffic specs are only closed-form approximations to the real traffic descriptions. Consequently, a static estimate of the bandwidth allocation needs for a data flow may deviate significantly from the actual needs. Such inaccuracies can best be countered by resorting to an 'allocate-and-see-what-happens' approach. In this context, the hose resource provisioning model studied elsewhere [8] estimates the bandwidth needs from the traffic parameters using linear programming techniques. Such estimates can be iteratively refined further to a reasonable accuracy with our online monitor-and-control approach.

3.2 Procedural realization of monitor-and-control

Figure 2 illustrates how a monitor-and-control paradigm can be realized for QoS adaptation. The agents S_C_Agent and R_C_Agent employ a heuristics-

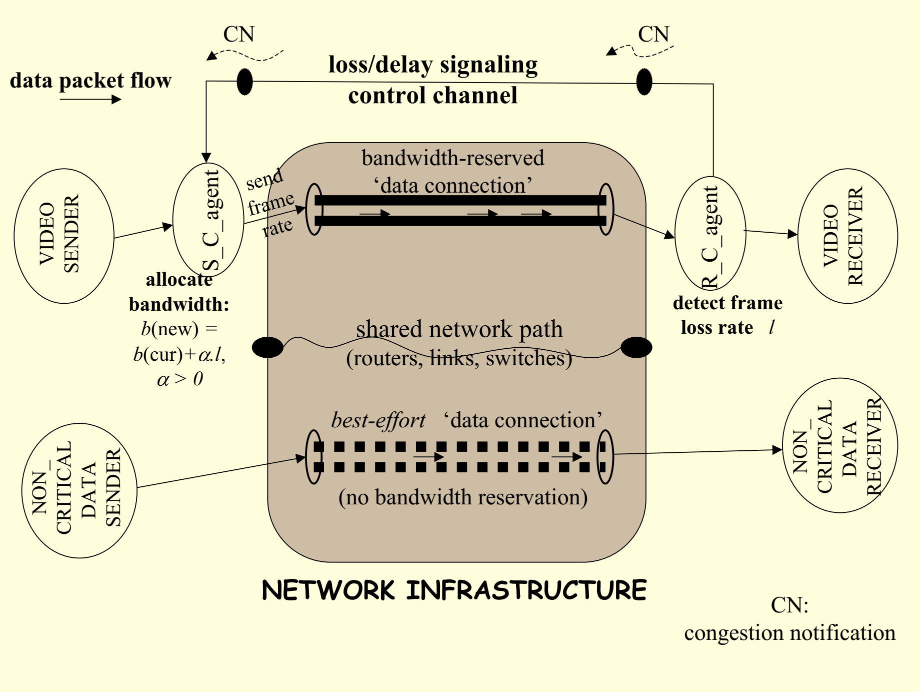

Fig. 2. Agent-based control of bandwidth allocation for video transport

based control algorithm that adjusts the allocation B_v based on the monitored data loss rate, in order to determine the bandwidth needs. For this purpose, we characterize the multimedia transport system with the following properties:

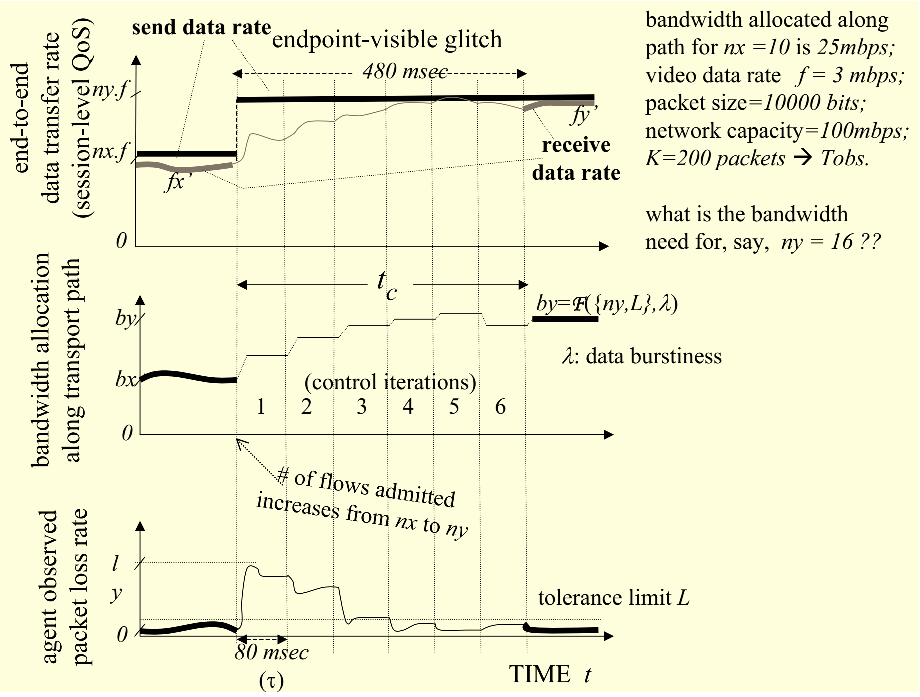

Fig. 3. An empirical time-line of bandwidth allocations based on packet loss monitoring

(i) When the bandwidth is increased from B_v to $B_v + \delta b$, it takes τ time units for the increase in bandwidth to have an observable effect on the data loss rate (due to the coupling between the BE and reserved connections that share a single network infrastructure);

(ii) The monitor observes K back-to-back packets in order to accurately determine the packet loss rate after a bandwidth allocation B_v occurs;

(iii) The control algorithm employs:

- A multiplicative law that increases B_v in proportion to the observed loss rate, as given by: $B_v(\text{new}) = B_v(\text{cur}) + \alpha l$, where l is the packet loss rate and $\alpha > 0$ is a constant;
- An additive law that decreases bandwidth by a fixed amount when no packet loss is observed, as given by: $B_v(\text{new}) = B_v(\text{cur}) - \beta$;

the increase/decrease steps are taken in each control iteration;

(iv) The data receivers exhibit a small amount of tolerance to packet loss (L).

Suppose the send rate over the connection is increased due to the admission of new data flows at time = 0, thereby increasing the number of flows from n_x to n_y — say, 6 more data flows are added to the current level of 10 data flows. For the parameter values $\tau = 80$ *msec*, $K = 200$, $\beta = 2{\times}10^5$ and $L = 2\%$, we can experimentally determine a timeline of how the controller may adjust the bandwidth reservation B_v — see Figure 3 for an illustration.

A reasonable value of α needs to be chosen though. The timeline for different values of α can also be experimentally determined, in order to study the convergence time and the stability of the control mechanism. The convergence property is given by a time constant: $t_c \approx R\tau$, where R is the number of control iterations to reach the steady state allocation. Referring to Figure 3, $\tau = 80$ *msec* and $R = 6$ — which gives $t_c = 480$ *msec*.

4 Latency in monitoring bandwidth effects

The control action, namely, a bandwidth allocation/de-allocation, occurs at the source agent (S_C_Agent). Whereas, the sensing action, namely, an observation of packet loss/delay, occurs at the receiver agent (R_C_Agent). Refer to Figure 2. The latency in observing the changes in packet loss caused by a change in bandwidth allocation arises due to factors such as the scheduler-level (short-term) cross-effects between logical connections and the loss notification delay over signaling paths.

4.1 Sources of control delays

Suppose X, X', X'' are the logical connections set up between customer edges E_s (source) and E_r (receiver) to carry distinct groups of data flows. E_r periodically sends a report to E_s indicating the packet loss rates observed on the bandwidth-reserved connections X and X' — note[1] that X'' is a 'best-effort' connection. See Figure 4. A network element in the data path may be a router in the core/access network or a overlay proxy node, forwarding the packets of connections X, X', X'' and any other unrelated cross-traffic through FIFO queues.

When a change in bandwidth allocation occurs, say, for connection X, how[2] long E_s should wait to assess the impact on packet loss depends on the following.

Forward propagation of scheduling effects:
A changed schedule of packet arrivals from E_s at a network element R is reflected in the packet departures from R only after all the packets currently queued up at R have departed. Given the FIFO packet queuing at R, the packet schedules generated at E_s transparently pass through R to the next element in the path. If $\overline{q}$ is the average queue length at R and $\overline{S}$ is the average packet size, a measure of the time taken for the packet departure schedule at R to change is $\frac{\overline{q}\overline{S}}{C_R}$, where

[1] RTCP-like signaling protocols can be employed for the loss notification.

[2] A bandwidth increase of δb on connection X manifests as increasing the scheduling weight assigned to the packet queue for X by $\frac{\delta b}{C}$.

C_R is the output capacity of R (note that $C_R \geq C$). Thus, the change in packet arrival schedule from E_s 'tunnels' through various network elements in the path until it reaches E_r, suffering a non-zero delay at each element — we denote the combined delay at all the elements as $t_{(s,r)}$.

Steady observation of packet loss behavior:
To assess the impact of a bandwidth change (that occurred at E_s) on packet loss rate, E_r should receive K packets of connection X under the changed arrival schedule. This manifests as a delay of $\frac{K.S}{C.w_X}$ after E_r receives the first packet of X under the changed schedule, where w_X is the weight assigned to X (on a normalized scale of 1.0) relative to the available capacity of data path from E_s to E_r. If T_l is the loss reporting interval, the number of back-to-back intervals that should elapse at E_r before the impact of a bandwidth change is fully captured in a loss report is: $\lceil \frac{t_{(s,r)} + \frac{K.S}{C.w_X}}{T_l} \rceil$.

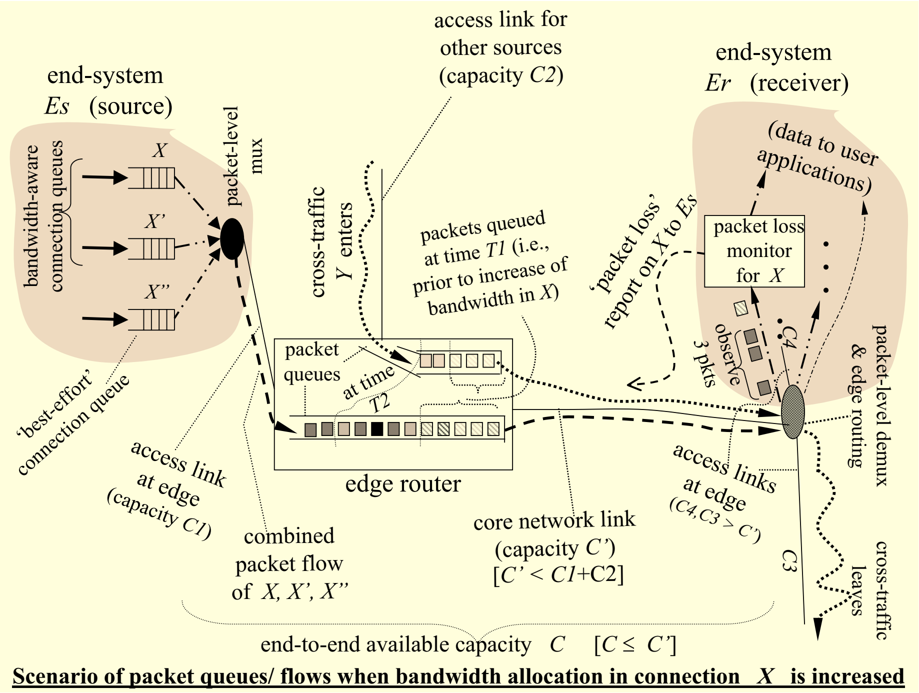
end-system
Es (source)
bandwidth-aware connection queues
X
X'
X"
packet-level mux
'best-effort' connection queue
access link at edge (capacity C1)
combined packet flow of X, X', X"
access link for other sources (capacity C2)
cross-traffic Y enters
packet queues
at time T2
packets queued at time T1 (i.e., prior to increase of bandwidth in X)
edge router
core network link (capacity C') [C' < C1+C2]
'packet loss' report on X to Es
end-system
Er (receiver)
(data to user applications)
packet loss monitor for X
observe 3 pkts
C4
access links at edge (C4,C3 > C')
C3
packet-level demux & edge routing
cross-traffic leaves
end-to-end available capacity C [C ≤ C']
Scenario of packet queues/ flows when bandwidth allocation in connection X is increased

Fig. 4. Latency factors during a change in bandwidth allocation

As can be seen, the system dynamics, as captured by the time-constant parameter τ, depends on the propagation of scheduling effects through network elements and the observation of resulting changes in packet arrivals at E_r.

4.2 Determination of system time-constant

Basically, E_s sends a time-stamped probe message over the data path, and E_r bounces this message back along the reverse path. The time elapsed between the send and return of the probe message at E_s gives the propagation delay $[t_{(s,r)} + t_{(r,s)}]$. The time-constant of the system, which is the time taken for the control point in E_s to observe the visible effects of a bandwidth change, is then given by: $\tau = t_{(s,r)} + t_{(r,s)} + T_l.[\lceil \frac{K.\overline{S}}{C.w_X.T_l} \rceil + 0.5]$.

Consider the data connections X, X', X'' set up from E_s to E_r — refer to Figure 4. The packets of these connections are sent through an edge router at E_s before entering the core network, and are then sent through an edge router at E_r before reaching the user-level applications. The packet loss monitor at E_r needs to observe K back-to-back packets of a connection before its loss report to E_s can accurately indicate the packet loss rate. Suppose E_s increases the bandwidth allocation on X at time T_1. For the packet queues shown in the Figure at time T_2 —- where $T_2 > T_1$, the latency at E_s in determining the impact of bandwidth increase on packet loss rate can be estimated in terms of the two packets queues (assume, for simplicity, that there are no cross-traffic packet arrivals after T_2).

Suppose, in Figure 4, 100 packets are already in the input queues of the router in access network when E_s increases the bandwidth of connection X (at time T_1). For $C = 25$ *mbps* and $\overline{S} = 10000$ *bits*, $t_{(s,r)}$ is the time for these packets to be drained out, which is 40 *msec*. For $K = 200$, $w_X = 0.6$, and $T_l = 250$ *msec*, the time-constant $\tau = 455$ *msec* — assuming that $t_{(r,s)} = t_{(s,r)}$.

A global control function is embodied in E_s and E_r to determine the τ parameter. Basically, E_r determines the delay in forward propagation of packet scheduling effects based on time-stamped probe messages sent by E_s over the data path. And, E_s determines the loss notification delay by having E_r time-stamp its loss report messages. This delay information, combined with the knowledge of K and C, allows determining τ.

In a way, τ captures the intrinsic 'feedback control' effects occurring within the WFQ packet scheduler when a step increase/decrease in bandwidth allocation is exercised in one of the connections sharing the transport link. Whereas, t_c depicts the number of such allocation steps needed to accurately determine the bandwidth. We note here that the work in [7] employs modeling techniques primarily to determine τ. In contrast, our goal is to determine t_c, with the knowledge of τ needed therein obtained by the methods we discussed earlier.

5 Simulation results

The simulation study is based on generating packet flows that closely approximate the video traffic traces of a *starwars* movie segment. These packet flows are then subjected to our bandwidth allocation scheme. The peak rate used in our study is 3.5 *mbps*. We do not use any apriori traffic analysis of the traces (such as estimating the burstiness and average rate parameters from the packet size distributions). Instead, we simply send the data traffic through the simulated model of our end-system.

Figure 5 gives our preliminary results on the adjustment of bandwidth allocation based on the monitored packet loss rate. The results are for the case of multiplicative increase in bandwidth. The transport network assumed is a physical link with a capacity of 100 *mbps*.

There are 10 video data flows initially, multiplexed over a single data connection. The initial combined bandwidth allocation is 30 *mbps* (i.e., 3.0 *mbps* per flow). Besides this candidate connection under study, two other connections carrying 8 and 6 flows respectively are also set up over the shared physical link (with approximately 25 *mbps* and 20 *mbps* allocated respectively). The latter is to incorporate the cross-coupling effects between multiple connections in our study. We then increase the number of flows from 10 to 16 on the connection being tested (i.e., apply a step increase in the input), and let the iterative control algorithm determine the new combined bandwidth needs. The result shows that it takes a little less than about 1.2 *sec* for a bandwidth adaptation procedure to complete successfully i.e., $t_c = 1.2$ *sec* — which is the time taken to determine the bandwidth allocation needs when the input traffic load is changed. There are about 6 control iterations, with each iteration taking about 200 *msec*. We have set 50 *msec* as the loss reporting interval.

We now include the additive decrease in bandwidth as well (when there is negligible packet loss) in the control algorithm, and then repeat the experiments.

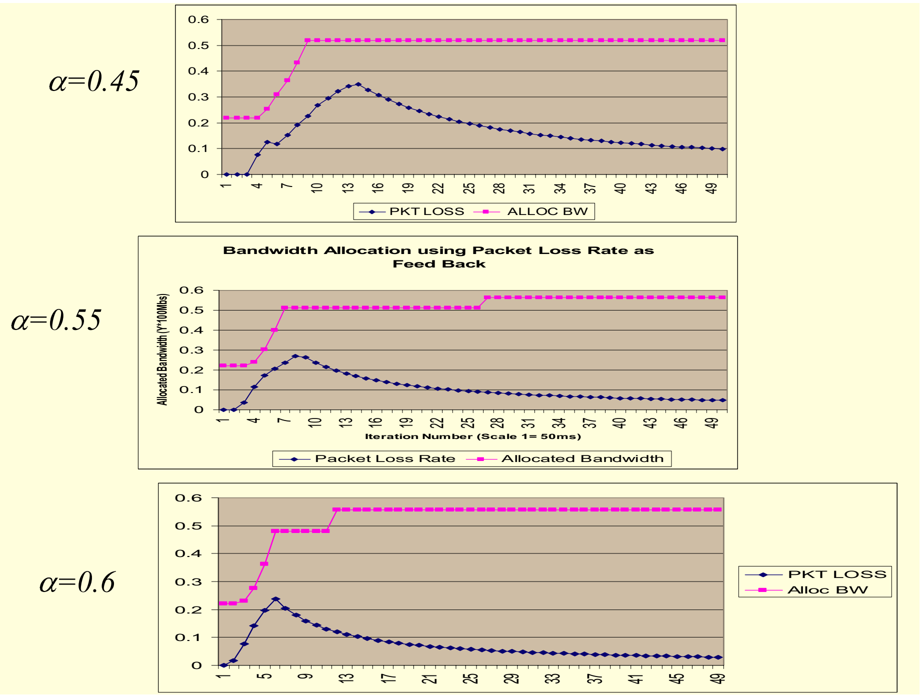

Fig. 5. Case of 'multiplicative increase' in bandwidth (with multiplexed video traffic)

The results are shown in Figures 6 and 7, for different values of α and β. The dynamics of the packet scheduler is itself measured by the parameter $\tau \approx 225\ msec$, which is the time taken for the loss rate to start decreasing (or increasing) after an additional bandwidth allocation (or a bandwidth de-allocation) has been made. A parameter value of $\beta = .012$ used in the experiments (with a scaling factor) corresponds to a bandwidth reduction of 1.2 $mbps$, whereas $\alpha = 0.55$ and $l = 0.1$ (depicting a 1% packet loss) corresponds to a bandwidth increase of 5.5 $mbps$ — with a bandwidth adjustment occurring potentially at every control iteration (which is of 50 $msec$ duration).

The time-constant of the overall system is: $t_c \approx 5.75\ sec$ for $[\alpha = 0.55, \beta = 0.012]$ (case 1) and $t_c \approx 5.25\ sec$ for $[\alpha = 0.65, \beta = 0.012]$ (case 2). The total time to determine the optimal bandwidth allocation is however considerably longer, taking many cycles of increase and decrease in bandwidths. The settling time is dependent on α and β. As can be seen, the cycle time in the steady state is about 750 $msec$ for case 1 and about 700 $msec$ for case 2. The bandwidth allocation settles down to about 45 $mbps$, which is about 2.8 $mbps$ per flow.

Figures 6 and 7 give some preliminary results on the dynamics of the packet scheduler, and at a macro-level, the dynamics of the overall QoS-adaptive bandwidth allocation.

For brevity, we have not attempted to determine the best allocation for various cases of bandwidth sharing and flow specs. Nevertheless, the results demonstrate the benefits of statistical bandwidth sharing among multiple video streams and the usefulness of heuristics-based control strategies in achieving a reasonably optimal steady-state allocation.

6 Conclusions

We described a model of MIAD-based bandwidth allocation for end-to-end QoS-controlled data connectivity. The model addresses the complex traffic management problem that involves bandwidth sharing across many applications over a data transport network: such as LAN/MANs, proxy-based overlay networks, and leased-line networks. Due to the complexity of interactions among the network components, a closed-form relationship does not exist to map the end-to-end QoS and the network resources. So, we resort to an on-line monitor-and-control approach that employs heuristics-based linear predictions to estimate the bandwidth needs. Using the observed packet loss/delays as indicative of the accuracy of bandwidth estimation, the controller adjusts the future bandwidth allocations.

We used video transport as a case study to illustrate the MIAD technique. We find that our technique scales quite well, is reasonably accurate, and does not require extensive management-level computations.

References

1. S. Erfani, V. B. Lawrence, and M. Malek. **The Management Paradigm Shift: Challenges from Element Management to Service Management**. In *Appli-*

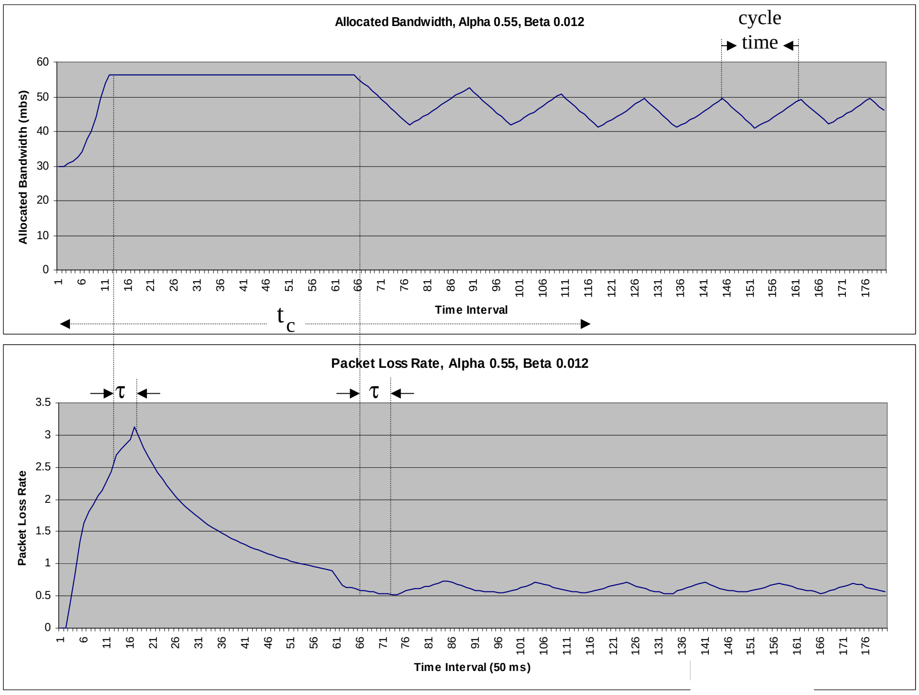

Fig. 6. Case 1 of 'multiplicative increase additive decrease' in bandwidth

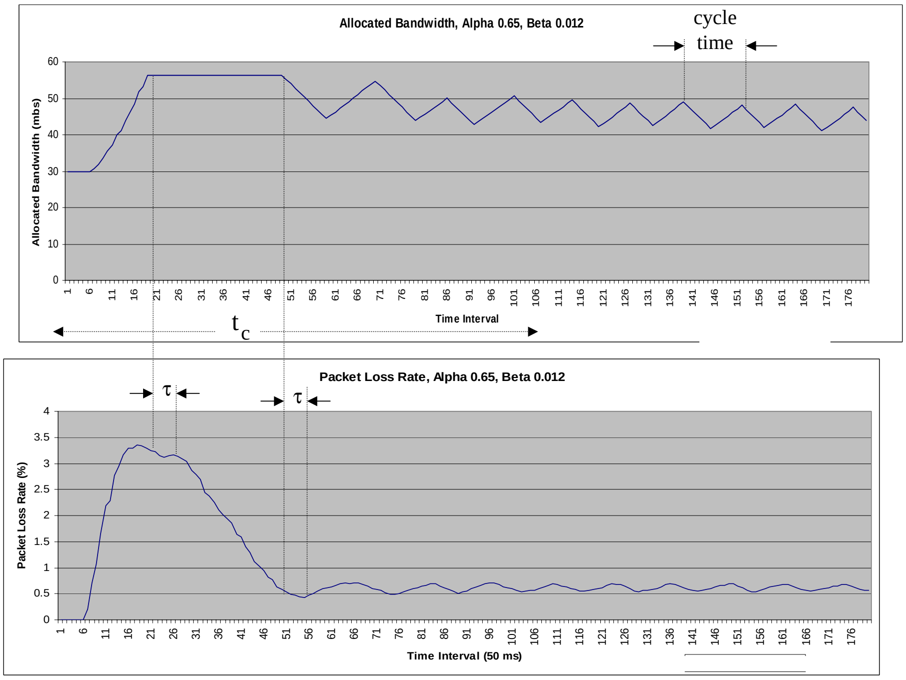

Fig. 7. Case 2 of 'multiplicative increase additive decrease' in bandwidth

cations, Platforms, and Services, Bell Labs Technical Journal, vol.5, no.3, pp.3-20, July-Sept. 2000.

2. S. Keshav. **Real-time Scheduling**. In *An Engineering Approach to Computer Networking*, Addison-Wesley Publ. Co., 1996
3. N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, and J.E.V. Merwe. **Resource Management with Hoses: Point-to-Cloud Services with Virtual Private Networks**. In *IEEE/ACM Transactions on Networking*, pp.679-692, Oct.2002.
4. Z. Wang, X. Zhu, and S. Singhal. **Utilization and SLO-based Control for Dynamic Sizing of Resource Partitions**. in DSOM 2005, Springer-Verlag LNCS 3775, pp.133-144, 2005.
5. K. Lai, L. Rasmusson, E. Adar, S. Sorkin, L. Zhang, and B. A. Huberman. **Tycoon: an Implementation of a Distributed Market-based Resource Allocation Systems**. In Technical Report, HP Labs, Palo Alto, Nov. 2004.
6. A, Sahai, S. Graupner, V. Machiraju, and A. V. Moorsel. **Specifying and Monitoring Guarantees in Commercial Grids through SLA**. In Technical Report (HPL-2002-324), HP Labs, Palo Alto, Nov. 2002.
7. B. Li and K. Nahrstedt. **A Control-based Middleware Framework for Quality of Service Adaptations**. In *IEEE Journal on Selected Areas in Communications*, vol.17, no.9, Sept.1999.
8. A. Juttner, I. Szabo, and A. Szentesi. **On Bandwidth Efficiency of the Hose Resource Management Model in Virtual Private Networks**. In proc. of INFOCOM 2003, pp.386-395, March 2003.