A Feedback Control Model for Multiple-Link Adaptive Bandwidth Provisioning Systems

Hao Wang
Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada
nhwang@sce.carleton.ca

Changcheng Huang
Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada
huang@sce.carleton.ca

James Yan
Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada
jim.yan@sympatico.ca

Abstract— **Future IP networks are required to support services with different and distinct end-to-end Quality of Service (QoS) requirements. Adaptive bandwidth provisioning serves as an attractive solution for providing guaranteed distinctive QoS and maintaining high network efficiency at the same time. However, most previous research on adaptive bandwidth provisioning is limited to the single-link case and assumes dedicated bandwidth. This paper studies multiple-link adaptive bandwidth provisioning for end-to-end statistical QoS guarantee in bandwidth sharing networks with Generalized Processor Sharing (GPS) schedulers. A feedback control model for end-to-end multiple-link adaptive bandwidth provisioning systems is presented. This model is verified by simulations, and it is shown to match the actual dynamics of adaptive bandwidth provisioning systems well. Based on this feedback control model, different controllers are designed and analyzed using control theory, and their performances are compared. The analysis and simulations show that the proposed end-to-end multiple-link bandwidth provisioning scheme is able to provide guaranteed end-to-end statistical QoS, and that both the adaptive *P* controller and adaptive *PI* controller can achieve better performance than the simple non-adaptive *P* controller.**

Keywords – Quality of Service; adaptive bandwidth provisioning; end-to-end statistical QoS guarantee; network resource allocation; GPS scheduler; network management

I. INTRODUCTION

Data, voice, and video services that are currently carried on multiple service-specific networks, as well as other emerging new services, will be carried on one single flexible and ubiquitous converged IP network. However, current IP technology is still mainly best effort and cannot provide guaranteed distinct Quality of Service (QoS) required by QoS-sensitive services. Over-provisioning is the currently applied strategy for QoS in IP networks, and it serves as a reasonable solution for the core networks where technologies have made bandwidth abundant and relatively cheap. However, it is not practical to over-provision the access networks where bandwidth is limited.

The Integrated Service (IntServ) and Differentiated Service (DiffServ) frameworks both attempt to provide QoS in IP networks. IntServ works with Resource ReSerVation Protocol (RSVP) providing end-to-end per-flow deterministic QoS guarantee. However, the scalability issue of IntServ makes it not suitable for the core of large-scale networks, such as the current Internet. DiffServ, on the other hand, works on a per-node per-aggregated-class basis and avoids the scalability problem, but it can only provide qualitative or relative QoS differentiation, no quantitative QoS guarantee is provided. A framework of IntServ over DiffServ was proposed in [3], in which DiffServ works in the core to solve the scalability issue and IntServ runs in the access networks to provide guaranteed end-to-end QoS. However, IntServ has low efficiency problem, because it is based on the deterministic Generalized Processor Sharing (GPS) scheduler analysis [1] [2], which allocates bandwidth to the flows for the worst case and does not fully explore the statistical multiplexing gain of the traffic. Running IntServ in the access networks, where network resource is limited, makes its low efficiency issue more prominent.

To resolve the low efficiency problem of deterministic GPS analysis, much research work has been done to study the stochastic bound of GPS scheduler, mostly using large deviation approximations [4] [5] [6] [7]. In addition, since the observed actual network traffic often has long-range dependence (LRD), some research has also been conducted studying GPS systems fed by LRD traffic [8] [9]. However, the analytical results of statistical GPS scheduler have limitations that prevent their application in real networks to ensure end-to-end QoS.

Measurement-based adaptive bandwidth provisioning is an attractive solution for providing guaranteed and distinct statistical QoS to traffic flows and achieving high network efficiency at the same time. The term "adaptive" here means that the bandwidth assigned to a QoS-sensitive flow is dynamically adjusted in reaction to the near real-time measurement of the traffic dynamics and the received QoS of the flow, so that only the minimum necessary bandwidth needed for achieving the required QoS is assigned to the flow.

There has been extensive research on measurement-based adaptive bandwidth provisioning, and the solution approaches can be classified as QoS-unaware or QoS-aware, depending on whether the algorithm tries to guarantee a specific QoS target. They can also be classified as feedforward or feedback schemes, based on the type of control mechanism utilized.

In QoS-unaware adaptive bandwidth provisioning [10]-[14], the algorithms adaptively adjust the bandwidth associated with the traffic flow without knowing the relationship between the adjusted bandwidth and the QoS experienced by the flow. They are not able to guarantee that the traffic meet a specific QoS target. In contrast, QoS-aware adaptive bandwidth provisioning [15]-[21] tries to achieve a desired QoS target, in terms of packet loss or delay, by adaptively adjusting the bandwidth assigned to the flow.

This work was supported in part by NSERC Grant RGP 261769 - 2003 and by NSERC CRDPJ 282925 - 03.

1-4244-0355-3/06/$20.00 (c) 2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

In feedforward adaptive bandwidth provisioning [10]-[16], information about the input traffic, e.g. the mean rate, peak rate and variance, is gathered and used by the control algorithm to adjust the bandwidth serving the input traffic. However, because there is no input about the flow's actual experienced QoS, feedforward adaptive bandwidth provisioning can not guarantee to achieve the targeted QoS accurately, even if when it is QoS aware.

In feedback adaptive bandwidth provisioning schemes [17]-[21], the algorithms measure the current QoS experienced by the flow and adjust the bandwidth provisioned to the flow accordingly. Therefore, they are able to reach the desired QoS of the flow. Feedback adaptive bandwidth provisioning schemes can be further classified as direct feedback and indirect feedback algorithms. In direct feedback [17] [19], the interested QoS metric is directly measured, and this measurement is used as the input to the control algorithm. In contrast, indirect feedback schemes [18] [20] [21] use measurements other than the interested QoS metric and rely on a certain mapping relationship between the measurements and the interested QoS metric.

One limitation of previous research on adaptive bandwidth provisioning is that most of them assume dedicated bandwidth, i.e. there is no bandwidth sharing among queues or flows. However, this assumption is not true in IP networks with GPS schedulers. Disregarding the bandwidth sharing among flows will result in bandwidth over-provisioning and reduced network efficiency.

In addition, most of the earlier research on adaptive bandwidth provisioning is restricted to the single link case. However, in real networks, a traffic flow has to traverse multiple links from its source to destination, and all these multiple links contribute to the end-to-end QoS received by the flow. It is suggested in some single-link based research that the end-to-end QoS requirement of the flow be divided into local QoS requirements on individual links, and each link performs its own local adaptive bandwidth provisioning. However, efficiently budgeting the end-to-end QoS requirement into local QoS requirements is a very difficult task, because it should consider the links' local information, such as utilization or congestion level. This problem is further complicated by the fact that the real network traffic is very dynamic, and the condition of a link can change dramatically and frequently.

Therefore, in this paper we propose end-to-end multiple-link adaptive bandwidth provisioning, which involves multiple links on the end-to-end path of the flow simultaneously. In this proposed end-to-end multiple-link bandwidth provisioning scheme, the actual end-to-end QoS experienced by a flow is measured at its receiving end of its path. The receiving end compares the actual measurement with the end-to-end QoS requirement, calculates the amount of bandwidths adjustment needed for this flow, and then sends this information to the multiple links along the flow's end-to-end path, so that each link adjusts its bandwidth assigned to this flow accordingly.

One implication of having multiple links involved in the adaptive bandwidth provisioning is that the indirect feedback schemes mentioned earlier, which map packet loss rate or packet delay into queue performance in the single queue single link case, is no longer applicable, because the explicit mapping relations are no longer valid in the multiple link situation. Thus, direct feedback schemes will be employed in the end-to-end adaptive bandwidth provisioning we propose.

The rest of the paper is organized as follows. In Section II, we present a feedback control model for the end-to-end multiple-link adaptive bandwidth provisioning system, which is verified by simulations using a simple non-adaptive *P* controller. Based on the feedback control model, adaptive *P* and adaptive *PI* controllers are designed in Section III and Section IV respectively, and their performances are studied as well. Finally, Section V summarizes the paper and indicates future research directions.

II. System Model

Most QoS sensitive traffic flows carry data for real-time applications, such as voice and video, and their QoS requirements can be well represented in a statistical form by the end-to-end packet delay violation ratio (DVR), which is the ratio of packets experiencing delay greater than the required delay bound [23]. In this paper, DVR is the QoS requirement we are interested in, and the end-to-end QoS requirements of the QoS flows are given in the statistical DVR form,

$$P(D_i > d_i) < \varepsilon_i , \tag{1}$$

in which D_i is the end-to-end packet delay of flow i, d_i is the required end-to-end delay bound of flow i, and ε_i is the maximum acceptable end-to-end DVR of flow i. It should be pointed out that (1) not only puts requirement on packet delay but also on packet loss, because lost packets are equivalent to having infinite large delays. In addition, for many real-time applications, packets having excessive delay are discarded and treated the same as packet loss.

One of the benefits of using a statistical QoS requirement as shown in (1) is that the efficiency of the access networks can be greatly improved, because the bandwidth needed for provisioning a flow is much less than the bandwidth needed when using deterministic guarantee [17].

A. Feedback Control System Model

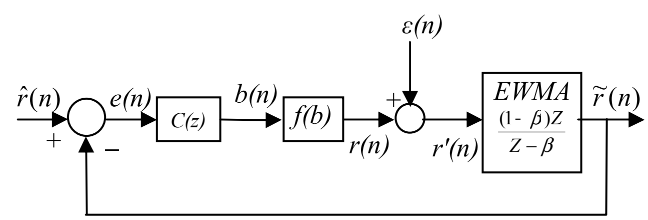

Figure 1. System overview.

By treating end-to-end adaptive bandwidth provisioning for a traffic flow in a GPS network as a feedback control system, we can draw the block diagram of this system as shown in Fig. 1.

The input of the system is $\hat{r}(n)$, which is the desired end-to-end DVR of the flow. The output of the system is $\tilde{r}(n)$, which is the result of passing *r'(n)* through an Exponential Weighted Moving Average (EWMA) filter with parameter β. *r'(n)* is the actual end-to-end DVR of this flow measured at the receiving end of the flow during the time interval [*(n-1)T, nT*], in which *T* is the control update interval. Since this is a direct feedback system, $\tilde{r}(n)$ is used as the feedback signal directly.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

The difference between $\hat{r}(n)$ and $\tilde{r}(n)$, denoted as *e(n)*, is used as the input to the controller *C(z),* which in turn calculates how much bandwidth, denoted as *b(n)*, should be assigned to this flow on the multiple links on its end-to-end path. The controller is located at the receiving end of the flow, and the amount of bandwidth assigned to this flow is calculated by this controller and then sent to all links along the flow's path by signaling.

The block *f(b)* is the DVR function of the flow, which is a mapping between the assigned bandwidth *b(n)* to this flow and the resulting DVR experienced by this flow *r(n)*. It should be pointed out here that *r(n)* is the long term steady state DVR of the flow assuming that *b(n)* and the characteristics of this flow as well as all other competing flows in the network are kept constant. Therefore the mapping between *b(n)* and *r(n)* is deterministic. For every specific value of *b(n)*, there is a corresponding deterministic specific value of *r(n)*. However, because of the stochastic nature of GPS networks, as well as the limited finite duration of the control interval, the actual measured DVR of the flow at the end of each control interval, denoted as *r'(n)*, is not deterministic but rather a random process, which can be modeled as the result of combining the deterministic *r(n)* with a random noise signal *ε(n)*, as show in Fig. 1.

The function of the EWMA block after *r'(n)* is to act as a low pass filter filtering out the random noise *ε(n)* within *r'(n)* and to generate $\tilde{r}(n)$, which is an estimation of *r(n)*. Please note that it is possible to replace the EWMA filter in the system block diagram with other types of low pass filters. We choose to use EWMA in this paper because of its simplicity, and it is also proven to work very well in our study. We leave the topic of using other types of low pass filters to future studies.

B. Linearized System Model

In Fig. 1, the DVR function of the flow, *f(b)*, is one of the key parts in the system model. However accurate description about *f(b)* is very difficult to get. Only results on bounds of the asymptotic tail queue behavior in the steady state are available. It is not difficult to realize that *f(b)* is a decreasing function of *b*, and from the results in [5] we know that for a flow with (ρ, Λ, α) exponentially bounded burstiness (EBB) arrival process, as defined in [24], the DVR function *f(b)* for this flow can be upper bounded by an exponential decay function[1],

$$r = \Pr\{D \geq d\} = f(b) \leq \ \Lambda^{*} \cdot e^{-\alpha d \cdot b}, \qquad (2)$$

in which, *r* is the DVR of this flow; *b* is the bandwidth assigned to this flow on its end-to-end path; *D* is the end-to-end delay of packets belonging to this flow; *d* is the required end-to-end delay bound of this flow; and Λ^{*} is a constant number, the calculation details of which can be found in [5].

It is well known that an exponential decay function is rather flat when the value of the function is approaching 0, therefore it is reasonable to assume that *f(b)*, upper bounded by an exponential decay function, should be relatively smooth and flat too when the value of *f(b)* approaches 0, which implies that for small values of DVR, the DVR mapping function, *f(b)*, near its operation point could be well approximated by a linear function:

$$f(b) \approx -Kb + B, \qquad (3)$$

in which $K > 0$, and *B* is a constant number.

The transfer function of a linear function is a constant gain, which is the derivative of this linear function. Therefore, after the linearization of *f(b)*, as expressed in (3), the system model shown in Fig. 1 is transformed as shown in Fig. 2. Please note that the specific value of *K* is still unknown here.

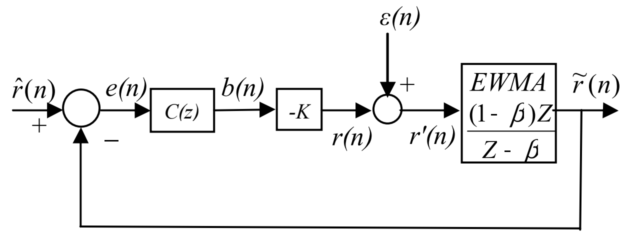

Figure 2. Linearized system block diagram.

C. Model Verification

To verify this linearized feedback control model, simulations using a simple non-adaptive proportional controller (*P* controller) are conducted.

The simulations are implemented on an *ns-2* platform. The topology is shown in Fig. 3. A QoS sensitive flow, named f_4, travels from node 0 to node 4, and there are four links on the end-to-end path of f_4. On each of the four links, there is also crossing traffic, named f_0, f_1, f_2 and f_3 respectively. The bandwidths of the four links are set to be 10M bps. In the simulation, the QoS flow is an aggregation of 200 voice sessions. The crossing traffic flows are imported using the BellCore trace [22] with the mean rate multiplied by a random factor, so that the four links have different congestion levels.

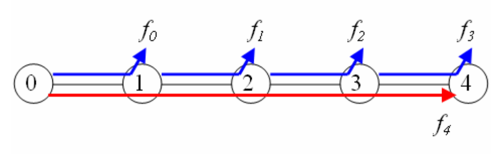

Figure 3. Simulation topology.

The control algorithm that implements the simple non-adaptive *P* controller for f_4 is given below.

$$b(n+1) = \max\{M\ ,\ M(1+\alpha\frac{\tilde{r}(n)-\hat{r}}{\hat{r}})\}, \qquad (4)$$

$$\tilde{r}(n) = (1-\beta)\cdot r'(n) - \beta \cdot r'(n-1), \qquad (5)$$

The reason of having the max{ } function in (4) is because the bandwidth assigned to a flow should not be less than its mean rate. The EWMA of measured DVR is an estimation of the actual DVR. The difference between the DVR estimation and DVR requirement is divided by the DVR requirement for normalization purpose. Then it is multiplied by the flow's mean rate to take the size of the flow into consideration, because larger flows will need more additional bandwidth to improve its QoS.

In the simulation, the required end-to-end delay bound and the acceptable end-to-end DVR of f_4 is set to be 100 ms and 10^{-3} respectively. The time interval for bandwidth adjustment is

1. Please note the notation differences between here and [5]. We use *b* to indicate the bandwidth assigned to the flow, while in [5] it is denoted by g_i^{net}.

set to be 20 seconds, and the values of α and β used are 2.0 and 0.9995 respectively.

The transfer function of the simple non-adaptive P controller as described in (4) and (5) is:

$$C(z) = -K_P = -\frac{M}{\hat{r}}\alpha . \quad (6)$$

Therefore, the system block diagram of the linearized system model using this simple P controller is as shown in Fig. 4.

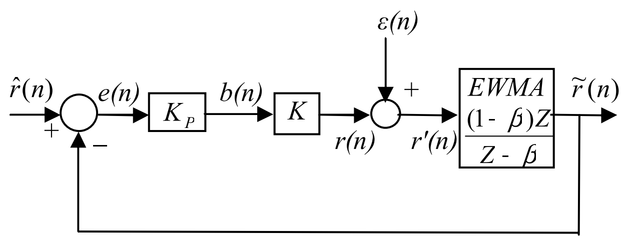

Figure 4. Linearized system block diagram with simple P controller.

It is easy to see that this is a first order system, with one pole at 0.9995 and one zero at the origin. Since the pole is inside the unit circle, from control theory we know that this first order system is stable and the unit step response of this system should have a steady state error:

$$e(n)|_{n\to\infty} = \frac{1}{1+G_{ol}} , \quad (7)$$

in which G_{ol} is the open-loop gain of the system,

$$G_{ol} = K_P \cdot K = \frac{M}{\hat{r}}\alpha K . \quad (8)$$

For the QoS flow f_4, the values of M and $\hat{r}$ are known. Since α is the preset parameter of the controller, it is also known. Although the exact value of K is unknown, it is a constant value in our linearized model. Therefore the value of G_{ol} can be altered by changing the value of α, and as indicated by (7) the steady state error of the system should also change accordingly. In addition, from (7) and (8) we know that if the linearized model matches the dynamics of the actual adaptive bandwidth provisioning system, the open-loop gain G_{ol}, derived from the steady state error using (7), should have a linear relationship with α, as expressed in (8).

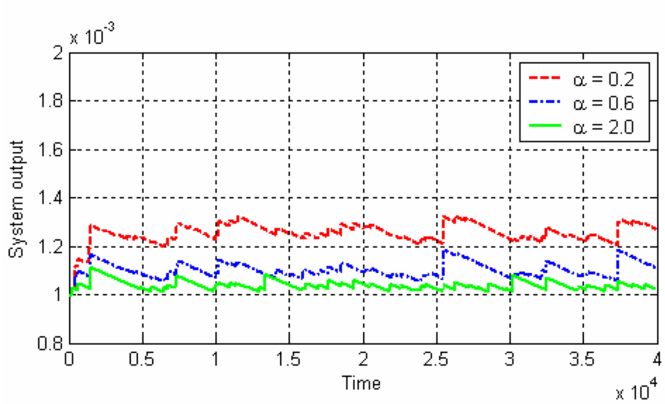

Figure 5. The system output using different α.

Fig. 5 shows the output of the adaptive bandwidth provisioning system for f_4 using several different values of α. It can be seen that the system output does have a steady state error, and the steady state error does change with different values of α. This agrees with the prediction made earlier using the linearized system model.

Simulations with more α values are also conducted. The steady state errors using different values of α are measured, and the G_{ol} corresponding to these steady state errors are calculated using (7). Fig. 6 plots the calculated open-loop gain against the value of α. It shows that, as predicted by (8) in the linearized system model, α and G_{ol} do have a nearly linear relationship, which further verifies our model.

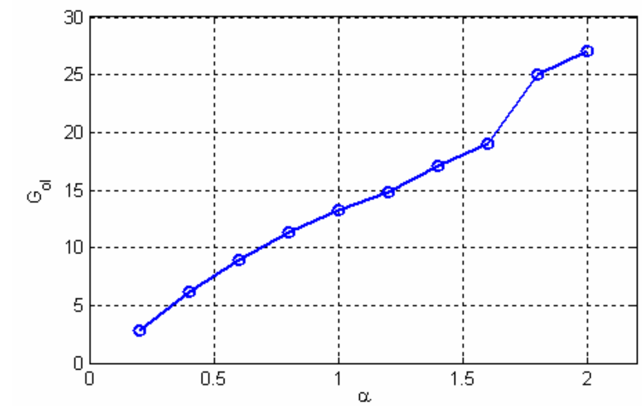

Figure 6. Open-loop gain VS. the value of α.

The above simulation results show that the linearized system model presented in this section is appropriate, and it matches the dynamics of real adaptive bandwidth provisioning system very well.

III. Adaptive P Controller

As discussed in previous section, when using a simple non-adaptive P controller, the steady state error of the system output has a relation with the system open-loop gain as described in (7). If a certain amount of steady state error, e.g. 10%, can be tolerated, and if the open-loop gain of the system is designed properly, a simple non-adaptive P controller can perform satisfactorily in guaranteeing the end-to-end statistical QoS requirement. However, because the value of K in the linearized system model, as shown in Fig. 2, is unknown, it is very difficult to design the proper value of α in order to get the desired open-loop gain, using (8).

From previous discussion, it is known that K is the derivative of the DVR function of the flow at its operation point, which is determined by the QoS requirement of the flow. Thus, for flows with different traffic characteristics or QoS requirements, the value of K can be very different. Even for the same traffic flow with the same QoS requirement, the value of K can change over time, because the network traffic condition is very dynamic. Therefore, it is very difficult, if not impossible, to design a particular value of α that works well with all flows under all conditions.

Fig. 7 provides a demonstration of this problem, which shows the performance of the QoS flow f_4, under two different end-to-end QoS requirements, given in the form of a

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

pair of values representing the required end-to-end delay bound and the acceptable DVR of the flow respectively. Assuming that 10% steady state error in the system output is acceptable, for the QoS flow f_4 with end-to-end QoS requirement (100 *ms*, 1.0×10^{-3}), the non-adaptive *P* controller with $\alpha = 0.6$ can achieve the desired system output. However, using the same controller but changing the end-to-end QoS requirement of flow f_4 to (50 *ms*, 1.0×10^{-3}), the steady state error of the system output becomes 17%.

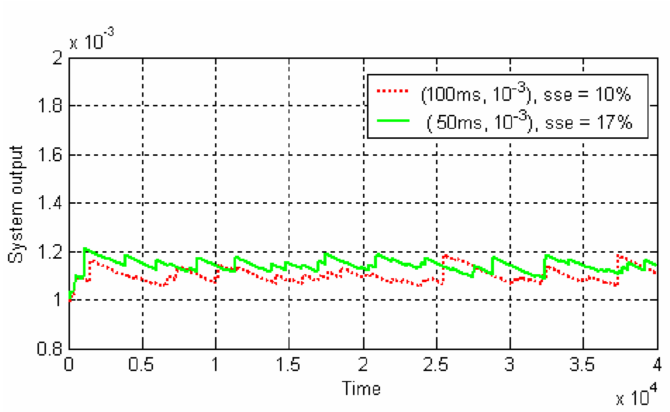

Figure 7. System output with different QoS requirements using the same simple non-adaptive *P* controller.

Another example is given in Fig. 8. The dotted line shows the system output of flow f_4 having end-to-end QoS requirement (100 *ms*, 1.0×10^{-3}) using the simple non-adaptive *P* controller with $\alpha = 0.6$. From earlier simulations, we know that the steady state error in this case is 10%. The solid line is the system output of f_4 having the same QoS requirement and using the same controller, but the traffic rates of the crossing traffic flows are doubled. The steady state error in this case changes to 15%.

Figure 8. System output with different network conditions using the same simple non-adaptive *P* controller.

The above discussions and examples show that because of the value of *K* is uncertain, it is very difficult to design a particular value of α that can achieve the same performance for different flows and different network conditions. Therefore, it is desirable that the value of *K* for a flow can be measured online, so that the appropriate value of α for this flow can be calculated to just meet the steady state requirement.

A. Estimation of K

In theory, the value of *K* can be estimated by observing the input and the output of the DVR function *f(b)*, which are *b(n)* and *r(n)* respectively. Unfortunately, the information about *r(n)* is not accessible. Only the measured DVR *r'(n)*, which is the combination of *r(n)* and the random noise signal *ε(n)*, is available. However, if we assume that the noise signal, *ε(n)*, can be completely filtered out by the EWMA low pass filter, then part of the system block diagram can be equivalently transformed as shown in Fig. 9.

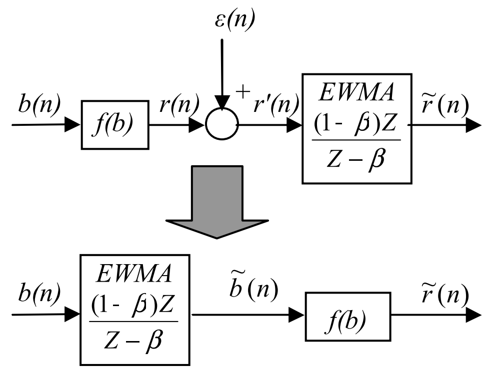

Figure 9. Equivalent transform of part of the system.

Fig. 9 suggests that the value of *K* can be estimated by observing $\tilde{b}(n)$ and $\tilde{r}(n)$ instead of $b(n)$ and $r(n)$. $\tilde{b}(n)$ is the result of passing *b(n)*, which is available information, through an EWMA filter. Therefore, from (3), we can have

$$\tilde{r}(n) = f(\tilde{b}(n)) \approx -K \cdot \tilde{b}(n) + B\,, \qquad (9)$$

$$\tilde{b}(n) = (1-\beta)\cdot b(n) + \beta \cdot b(n-1)\,. \qquad (10)$$

From (9), we can further derive that

$$K \approx -\frac{\tilde{r}(n) - \tilde{r}(n-1)}{\tilde{b}(n) - \tilde{b}(n-1)} = K'(n)\,, \qquad (11)$$

In this paper, *K'(n)* is treated as the value of *K* interfered by some noise signal. Therefore, we pass *K'(n)* through an EWMA filter, and use the resulting output, $\tilde{K}(n)$, as the online estimation of *K* for the linearized DVR function of this flow.

$$\tilde{K}(n) = (1-\beta)\cdot K'(n) + \beta \cdot K'(n-1)\,. \qquad (12)$$

B. Adaptive P Controller

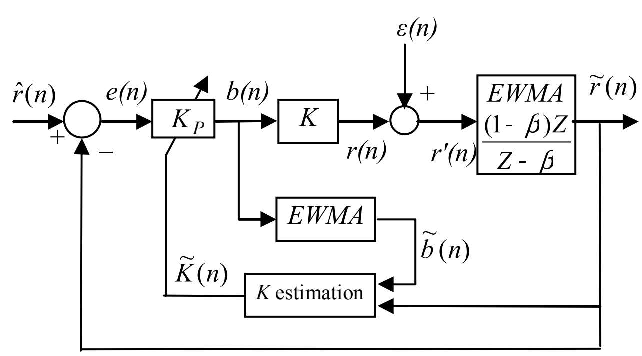

Figure 10. Linearized system block diagram with adaptive *P* controller.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

Having $\widetilde{K}(n)$ as the online estimation of *K*, it is possible to adjust the parameters of the *P* controller according to this estimation. Therefore, we propose an adaptive *P* controller for adaptive bandwidth provisioning systems. The system block diagram with the proposed adaptive *P* controller is shown in Fig. 10, in which the *K* estimation block performs the estimation algorithm as described in (11) and (12). The rule for adjusting the parameter K_P of the *P* controller is given below.

$$K_P(n+1) = G_{ol} / \widetilde{K}(n) . \qquad (16)$$

in which, G_{ol} is the designed open-loop gain of the system, which is determined by how much steady state error is acceptable in the system output.

The performance of the proposed adaptive *P* controller is shown in Fig. 11 and Fig. 12 below. In this simulation, the designed open-loop gain is 9, so that the steady state errors should be less than 10%.

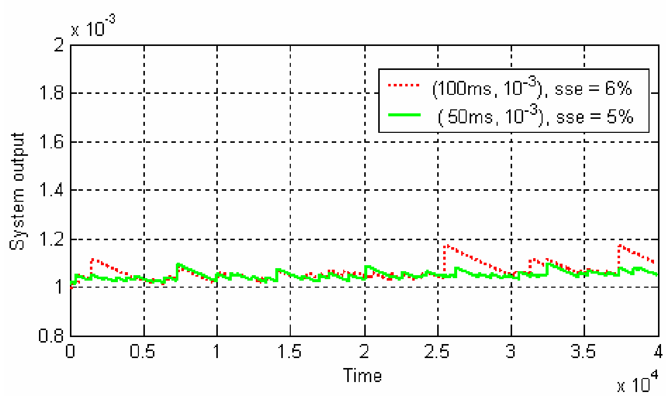

Figure 11. System output with different QoS requirements using adaptive *P* controller.

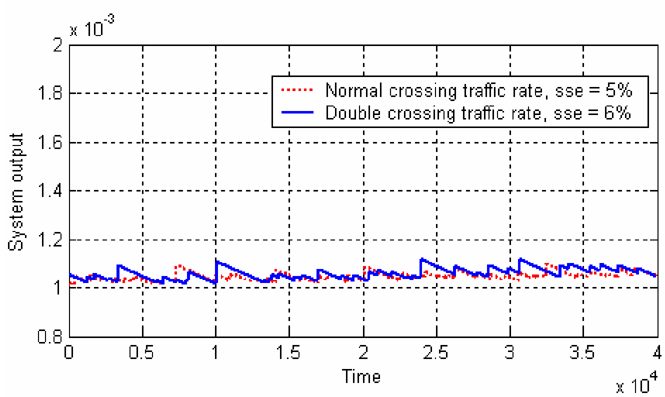

Figure 12. System output with different network conditions using adaptive *P* controller.

It can be seen that, despite different end-to-end QoS requirements and different network conditions, the output of the system maintains similar performances, and the actual steady state errors of the system output meet the design expectation. This confirms that our proposed adaptive *P* controller is able to adjust itself to different values of *K*.

IV. Adaptive *PI* Controller

In the previous section, it is shown that our proposed adaptive *P* controller can adapt itself to different network traffic and QoS requirements. But there is an observable steady state error in the system output using adaptive *P* controller. From our feedback control system model, we can tell that this is determined by the fact that the adaptive bandwidth provisioning system is a first order system. Therefore, from control theory, using *P* controllers cannot remove the steady state error from the system output, no matter what parameter values are used and no matter whether the controller is adaptive or non-adaptive.

In order to eliminate the steady state error in the system output, we propose an adaptive *PI* controller in this section. The block diagram of the system with adaptive *PI* controller is the same as the one with adaptive *P* controller shown in Fig. 10. The only difference is that the transfer function of the controller is changed from K_P to $K_{PI} \cdot [1+\gamma(z+1)/(z-1)]$. In our simulation we set $\gamma = 0.001$.

Similar to the adaptive *P* controller case, the value of K_{PI} is adjusted as:

$$K_{PI}(n+1) = G_{ol} / \widetilde{K}(n) . \qquad (19)$$

The results of using adaptive *PI* controller are shown in Fig. 13 and Fig. 14. Compared with Fig. 11 and Fig. 12, we can see that in addition to the ability of adjusting itself to different traffic and different QoS requirements as the adaptive *P* controller does, adaptive *PI* controller successfully removes the steady state error from the system output.

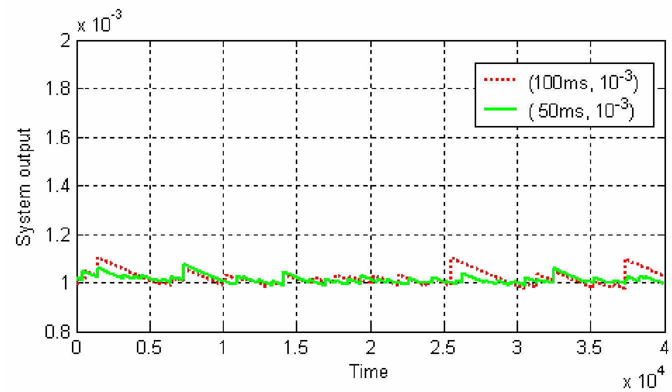

Figure 13. System output with different QoS requirements using adaptive *PI* controller.

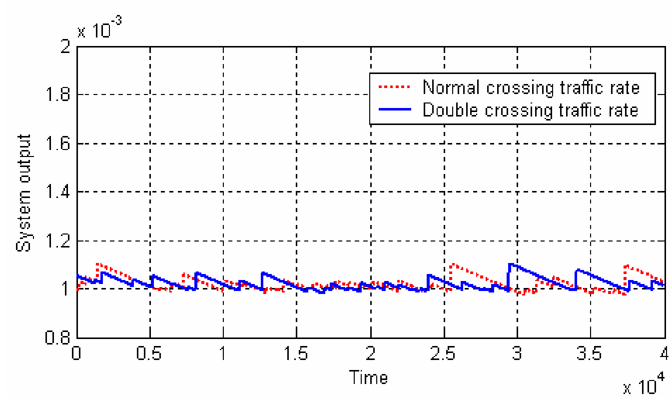

Figure 14. System output with different network conditions using adaptive *PI* controller.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

V. CONCLUSIONS

This paper studies end-to-end multiple-link adaptive bandwidth provisioning to provide guaranteed statistical end-to-end QoS for traffic flows in bandwidth sharing Generalized Processor Sharing (GPS) IP networks.

The main contribution of this paper is that it presents a linearized feedback control system model for the proposed end-to-end multiple-link adaptive bandwidth provisioning scheme. This system model is verified by simulations and is shown to match well the actual dynamics of adaptive bandwidth provisioning systems.

Based on this feedback control system model, an adaptive *P* controller and an adaptive *PI* controller are proposed. It is shown that compared with the simple non-adaptive *P* controllers, the adaptive *P* and adaptive *PI* controllers perform better in handling different traffic conditions and different QoS requirements.

Because the design of the adaptive *P* controller and adaptive *PI* controller are based on our presented linearized feedback control system model, their performances can serve as good demonstrations of the usefulness and effectiveness of our model.

Some possible future research directions include using the adaptive bandwidth provisioning system model presented in this paper to study other issues such as optimality, nonlinearity and robustness of the system; investigating the possibility of using other low pass filters instead of the EWMA filter used in this paper; and assessing the performance of other types of controllers in adaptive bandwidth provisioning systems.

In addition, there are some other issues that need to be further investigated when implementing the proposed end-to-end multiple-link adaptive bandwidth provisioning scheme in real networks, such as QoS measurement, end-to-end feedback realization, protocol design, etc. We leave these issues to future studies.

REFERENCES

[1] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case", IEEE/ACM Trans. Networking, vol. 1, pp. 344-357, June 1993.

[2] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case", IEEE/ACM Trans. Networking, vol.2, pp.137-150, April 1994.

[3] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, "A framework for integrated services operation over diffserv networks", RFC2998, November 2000.

[4] G. de Veciana and G. Kesidis, "Bandwidth allocation for multiple qualities of service using generalized processor sharing", IEEE Trans. on Information Theory, vol. 42, No. 1, 1996.

[5] Z. L. Zhang, D. Towsley, and J. Kurose "Statistical analysis of the gerneralized processor sharing scheduling disipline", IEEE Jounal on Selected Areas in Communications, vol. 13, 1995.

[6] Z. L. Zhang, "Large deviations and the generalized processor sharing scheduling for a two-queue system", Queueing Systems: Theory and Applications, vol. 28, 1997.

[7] Z. L. Zhang, "Large deviations and the generalized processor sharing scheduling for a multiple-queue system", Queueing Systems: Theory and Applications, vol. 28, 1998.

[8] S. Borst, O. Boxma and P. Jelenkovic, "Asymptotic behavior of generalized processor sharing with long-tailed traffic sources", IEEE INFOCOM 2000.

[9] M. van Uitert and S. Borst, "Generalized processor sharing networks fed by heavy-tailed traffic flows", IEEE INFOCOM 2001.

[10] Cisco Systems, "Cisco MPLS autobandwidth allocator for MPLS traffic engineering: A unique new feature of Cisco IOS software", White Paper, Jun 27, 2003.

[11] A. M. Adas, "Using adaptive linear prediction to support real-time VBR video under RCBR network service model", IEEE/ACM Transactions on Networking, Vol. 6, No. 5, 1998.

[12] H. Wang, C. Huang, and J. Yan. "Performance of prediction-based dynamic bandwidth provisioning", The 19th International Teletraffic Congress (ITC19), 2005.

[13] B. Krithikaivasan, K. Deka, and D. Medhi. "Adaptive bandwidth provisioning envelope based on discrete temporal network measurements", IEEE INFOCOM 2004.

[14] W. Cui and M. Bassiouni, "Virtual private network bandwidth management with traffic prediction", Computer Networks, vol. 42, no. 6, p. 765-778, August 2003.

[15] N. G. Duffield, P. Goyal, and A. Greenberg, "A flexible model for resource management in virtual private networks", ACM SIGCOMM'99, Cambridge, MA, Oct. 1999.

[16] H. T. Tran, and T. Ziegler. "On the adaptive bandwidth provisioning schemes", IEEE ICC 2004.

[17] K. Gopalan, T. Chiueh, and Y. Lin, "Probabilistic delay guarantees using delay distribution measurement", ACM Multimedia 2004.

[18] P. Siripongwutikorn, S. Banerjee, and D. Tipper, "Adaptive bandwidth control for efficient aggregate QoS provisioning,", IEEE GLOBECOM , Nov. 2002.

[19] E. W. Fulp, and D. S. Reeves, "On-line dynamic bandwidth allocation", IEEE International Conference on Network Protocols, 1997.

[20] G. Kesidis, "Bandwidth adjustments using online packet-level measurements", SPIE Conf. Performance and Control of Network Systems, Boston, MA, Sept. 1999.

[21] I. Hsu and J. Walrand, "Dynamic bandwidth allocation for ATM switches", Journal of Applied Probability, vol. 33, 1996

[22] W. Leland and D. Wilson, "High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection", IEEE INFOCOM, April 1991.

[23] Y.Wang and Q. Zhu, "Error control and concealment for video communication: A review", Proc. IEEE, vol. 86, pp. 974–997, May 1998.

[24] O. Yaron and M. Sidi, "Performance and stability of communication networks via robust exponential bounds", IEEE/ACM Trans. on Networking, vol. 1, pp. 372-385, 1993.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.