Design and Analysis of Queue Control Functions for Explicit Rate Switch Schemes

Bobby Vandalore, Raj Jain, Rohit Goyal, Sonia Fahmy
The Ohio State University
Department of Computer and Information Science
Columbus, OH 43210-1277
E-mail: {vandalor, jain, goyal, fahmy}@cis.ohio-state.edu

New Address: Raj Jain, jain@wustl.edu, http://www.cse.wustl.edu/~jain/

Abstract

ABR rate allocation schemes can achieve high link utilizations by maintaining non-zero (small) queues in the steady state, and draining queues when the sources do not have data to send. The queue length (and queuing delays) can be controlled if part of the available bandwidth is used for draining queues in the event of queue build up. A simple threshold function can allocate such bandwidth to drain queues. Better control of the queues, and hence delay, can be achieved using more sophisticated queue control functions. We study the design and analysis of several such queue control functions: the step, linear, hyperbolic and inverse hyperbolic functions. Analytical explanation and simulation results consistent with analysis are presented. From the study, we conclude that the inverse hyperbolic is the best queue control function. To reduce complexity, the linear function can be used since it performs satisfactorily in most cases.

1. Introduction

Asynchronous transfer mode (ATM) is the chosen technology for implementing the broadband integrated services digital network (B-ISDN). The available bit rate (ABR) service provided by ATM can be used to transport data traffic giving minimum rate guarantees. ABR uses closed loop feedback to control the source rates. The source periodically sends a resource management (RM) cell to gather information from the network [7]. The RM cells are turned around at the destination. The switches along the path indicate the rate which they can currently support. When the source receives the returning RM cells, it adjusts its allowed cell rate based on the explicit rate indicated in the RM cell. There are several explicit rate ABR switch schemes ([6, 2, 1, 3, 5]) that can compute appropriate explicit rates and indicate them in RM cells. This study uses the ERICA+ switch scheme [5], but it also examines how to use a queue control function to improve the performance of any scheme.

The goals of rate allocation schemes include maintaining high link utilizations, small queuing delays, low cell loss, and fairness among competing sources. In order to support (low quality) video sources over the ABR service, it is also desirable that, in the steady state, the rates and queuing delays not be highly variable. One way to achieve high utilization and low queuing delay is to vary the target ABR rate as a function of the queue lengths.

In this paper, we study several queue control functions which satisfy the above requirements. We present an analytical explanation for the performance of these functions, and give simulation results consistent with the analysis. The convergence time and behavior of the queues in the steady state are used as metrics to evaluate the queue control functions.

2. Queue Control Functions

In this section, the relationship between the queue length and queue control function is derived. This section also discusses the design considerations of queue control functions. The following terms are used in the discussion:

N number of sources.

t_s averaging interval, i.e., the interval in which measurements are made, and feedback to the sources is calculated at the switch (see [5] for more details).

$r_i(t)$ rate of source i.

$er_i(t)$ explicit rate of source i calculated at the switch.

t_p propagation time from the source to switch.

t_f feedback delay (twice t_p).

0-8186-9014-3/98 $10.00 © 1998 IEEE

Proceedings of IC3N'98, pp. 780-786, (7 pages), October 1998

R_{link} link rate.

R_l available ABR capacity.

$Q(t)$ switch queue length (in cells).

$R(t)$ aggregate *input rate* seen at switch, $R(t) = \sum_{i=1,N} r_i(t)$.

$C(t)$ (conversion function) number of cells transmitted in time t at link rate. $C(t) = (R_l \times t)/424$ if R_l is given in Mbps.

$X(t)$ denotes that X is a function of time. We assume that the explicit rate switch scheme operates at the output port and gives feedback to source i according to the following equation: Target rate $(er_i) = f(Q) \times F(r_i, R_{link}, HPR)$, where HPR is the total rate of high priority traffic such as VBR and CBR. For simplicity, we ignore higher priority traffic such as CBR and VBR, and assume that R_l is same as the link rate R_{link}. The function $f(Q)$ is the queue control function. The queue length increases if $f(Q) > 1$; remains constant if $f(Q) = 1$; and decreases if $f(Q) < 1$. Hence, by using an appropriate function $f(Q)$, the queue length can be controlled.

2.1. Queue Length Function

To simplify the analysis, we assume that the propagation delay t_p from source to switch switch is the same for all sources. Due to propagation delay t_p, the rate seen at the switch at time t is the same as the source rate at time $t - t_p$. The sources adjust their rates to the explicit rate indicated in the BRM cells. Due to propagation delay, the sources adjust their rate at time t to the explicit rate indicated at time $t - t_p$.

In one averaging interval t_s, $Q(t)$ is drained by $R_l \times C(t_s)$ cells. The queue builds up at *input rate*. Hence, $Q(t)$ can be expressed as follows :

$Q(t) = Q(t - t_s) + (\sum_{i=1,N} r_i(t - t_p) - R_l)C(t_s) \rightarrow$
$Q(t) = Q(t - t_s) + (\sum_{i=1,N} er_i(t - t_f) - R_l)C(t_s) \rightarrow$
$Q(t) = Q(t - t_s) + (R(t) - R_l)C(t_s)$

The switch scheme tries to adjust the input rate $R(t)$ to match the output rate depending on the current queue size, i.e., $R(t) = f(Q(t)) \times$ Available ABR Capacity. We assume no higher priority traffic, so $R(t) = f(Q(t)) \times R_l$. Hence,

$$Q(t) = Q(t - t_s) + (f(Q(t - t_s)) - 1)R_l C(t_s)$$

The sources adjust their rates $r_i(t)$ based on the explicit rate feedback from the switch. The feedback reaches the switch after time t_p. Hence, $r_i(t)$ (source rate) can be expressed using the following equation:

$r_i(t) = er_i(t - t_p) = f(Q(t - t_f)) \times F(r_i(t - 2t_p), R_{link})$.

Since $t_f = 2t_p$, we get: $r_i(t) = f(Q(t - t_f)) \times F(r_i(t - t_f), R_{link})$.

For the *ERICA+* scheme, the above function is as follows: $r_i(t) = f(Q(t - t_f)) \times max(\frac{r_i(t-t_f) \times R_{link}}{R(t-t_f)}, \frac{R_{link}}{n})$.

For other schemes, the following modification can be done to incorporate queue control. Let $er_A(t)$ be the explicit rate calculated by an algorithm A. Then queue control can be incorporated by adding the following as last step in the algorithm A: $er_A(t) = max(f(Q(t - t_f)) \times er_A(t - t_f), PCR)$, where PCR is the peak cell rate. It can be shown that if the algorithm A converges to max-min rates, then the modified algorithm also converges to the max-min rate.

2.2. Design of a Queue Control Function

The design considerations for the queue control functions are as follows:

1. If the queue length is very small, it should be increased, so that the scheme can maintain small queues which can drained when the link is under utilized. This implies that $f(Q)$ should be greater than one for small queue lengths.

2. In the steady state, we want the queue length to be almost constant, and the computed rate to be the max-min fair rate. The function $Q(t)$ satisfies the goal of constant queue length if $f(Q) = 1$ in the steady state. If the switch scheme is fair, the steady state rates will be max-min fair.

3. If the queue is large, then part of the link capacity must be used to drain the queues. Hence $f(Q)$ should be less than one. It is desirable not to use all the capacity to drain the queue. Therefore, there is a minimum threshold, queue drain limit factor (QDLF), for $f(Q)$.

4. The $f(Q)$ function has to be continuous. Discontinuities imply sudden changes which give rise to oscillations of rates and queues.

The queue control function with above properties will be of the form

$$f(Q) \begin{cases} > 1 & 0 \leq Q \leq Q_0 \\ = 1 & Q_0 < Q \leq Q_1 \\ < 1 & Q_1 < Q \leq Q_2 \\ = \text{QDLF} & Q_2 < Q < \infty \end{cases}$$

where $Q_0 < Q_1 < Q_2 < \infty$. The potential candidates are the step, linear, hyperbolic and inverse hyperbolic functions (see table 1). All these functions evaluate to 1 in the range $Q_0 < Q \leq Q_1$ and $QDLF$ in the range $Q_2 < Q < \infty$. The step function can, in general, have n steps. We use $n = 4$. The linear function can be implemented in an efficient manner, using shift operations, if m_a and m_b are of the form $1/2^k$ and the queue length is counted in terms of Q_0. The hyperbolic and inverse hyperbolic functions take more time to calculate, since they have a division operation. For large values of h_a, the hyperbolic function

Name	$0 \leq Q \leq Q_0$	$Q_1 < Q \leq Q_2$
Step	s_a	s_b
Linear	$1 - m_b \frac{(Q-Q_0)}{Q_0}$	$1 - m_a \frac{(Q-Q_1)}{Q_1}$
Hyperbolic	$\frac{\lambda_b Q_0}{(\lambda_b-1)Q+Q_0}$	$\frac{h_a Q_1}{(h_a-1)Q+Q_1}$
Inv. Hyperbolic	$\frac{\lambda_b Q_0}{(\lambda_b-1)Q+Q_0}$	$\frac{\lambda_a Q_1}{(\lambda_a+1)Q_1-Q}$

Table 1. Queue control functions (f(Q))

becomes similar to step function. For an h_a value close to 1, the hyperbolic function approaches the linear function. The inverse hyperbolic function is continuous and smooth both at Q_0 and Q_1.

The curve used in the control function in the underload region is called the "a-curve" and the one used in the overload region is called the "b-curve". The parameters used in the a-curve s_a, m_a, h_a, λ_a are called *a-parameters*; s_b, m_b, h_b, λ_b are called the *b-parameters*. Since all the functions are continuous, at Q_2 we have the equation $f(Q_2) = QDLF$. So, Q_2 can be expressed in terms of $QDLF$ and *a-parameters* for linear, hyperbolic and inverse hyperbolic functions.

3. Metrics

To compare the performance of the queue control functions, the *convergence time* metric is used. Convergence time is defined as the time the scheme takes to converge to steady state. To compute the convergence time, the variance and standard deviation of the required variables are calculated between ($i \times t_k$,$(i+1) \times t_k$) for $i = 0, 1, \ldots$, where t_k (100 ms) is a small time interval. Initially, the standard deviation is large due to oscillations. The convergence time is $i \times t_k$, after which the variance is small. Also the graphs of (mean + standard deviation) values of the variable versus time are plotted (these graphs are given in [9], which is an extended version of this paper). From these graphs, the convergence time can be calculated.

4. Analytical Explanation

In this section, we analyze the behavior of the proposed queue control functions. We assume a simple configuration in our analysis: N persistent ABR sources (always having data to send) are sending data to N ABR destinations (figure 1). A performance study under more strenuous conditions is done through simulation on the Generic Fairness configuration - 2 [8] in section 6.

The change in queue length in an averaging interval t_s is given by:

$$\Delta Q = (f(Q(t - t_f)) - 1) \times R_l \times C(t_s)$$

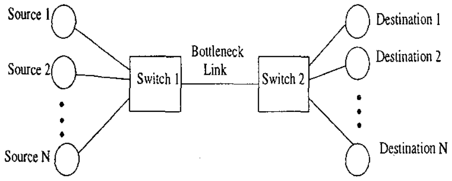


Figure 1. N Sources - N Destinations Configuration

Initial behavior: In the beginning, the queue lengths grow depending on the initial cell rate (ICR) value. So the maximum queue depends on the ICR and round trip time and is independent of the queue control function used. The feedback information reaches the sources and the sources adjust their rates accordingly.

Under-utilization: The switch scheme initially estimates that the link is under utilized. The queue control function evaluates to $f(Q) > 1$, during under utilization ($Q < Q_0$). Thus the switch gives feedback to the sources to increase their rates, which increases queue length at the switch. The rate of increase of the queue length is dictated by the b-curve of the queue control function.

Overload: Either due to the increase in the source rates or the high ICRs, an overload condition occurs and queues at the switch grow. Initially, the feedback information is not accurate; hence, the queue might grow beyond the Q_2 threshold. In such a heavy overload condition, the queues are quickly drained by using the (1-QDLF) fraction of link capacity. In the meantime, the feedback control loop is established, and the switch gives reliable feedback to the sources. For the mild overload region, i.e., queue length in range of $Q_1 \leq Q \leq Q_2$, the a-curve is the queue control function. The value of $f(Q)$ is less than one in this region, which effectively decreases the queue length.

Steady state: The feedback tries to match the input rate to the output rate. As the input rate approaches the output rate, the oscillations die down and the network reaches the steady state. In the steady state, the rates and the queue lengths remain constant since $f(Q) = 1$.

For all four queue control functions, $f(Q) > 1$ in the range $Q_0 \leq Q < Q_1$, and $f(Q) < 1$ in the range $Q_1 \leq Q < Q_2$. The change in queue length depends on the value of $f(Q) - 1$. Hence, the change in the queue length is an additive increase for the underutilized region [4]. The queue length converges to a value in between Q_0 and Q_1. An interesting point to note is that the amount of increase and the amount of decrease itself is dictated by the current queue length and the difference between the current queue length and the steady state queue length.

4.1. Step Function

If $Q(t - t_f) < Q_0$, then $f(Q(t - t_f) = s_b > 1$, so the queue grows till feedback information is passed to the sources indicating a rate decrease. The queue grows for t_f seconds as follows:

$$Q(t) = Q(t - t_f) + (s_b - 1) \times R_l C(t_f)$$

If the condition $Q_0 < Q(t) < Q_1$ is satisfied and the input rate matches the output rate, then the steady state is achieved, and the queue remains at this constant length.

If $Q_1 < Q(t) < Q_2$, then $Q(t)$ starts decreasing with slope $-(1 - s_a)$. This decrease also takes place for t_f time, if the queue length is between Q_0 and Q_1, and if the input rate is close to the output rate then again the steady state is achieved.

Therefore, for the system to achieve the steady state, the value of the queue length after one feedback interval should be within the range Q_0 and Q_1. This requirement is satisfied if the condition $Q_1 > Q_0 + (s_b - 1) \times R_l C(t_f)$ holds. Since the step function has discontinuities, it is very sensitive to the queue length value near the thresholds and the steady state might not be reached if the parameters are not set properly. In such a case, the queue grows from a value below Q_0 for duration t_f seconds, crosses Q_1 and decreases for t_f seconds to a value less than Q_0. This pattern repeats giving rise to periodic oscillations.

4.2. Linear Function

If $Q(t - t_f) < Q_0$, then $f(Q(t)) > 1$. As with the step function, the queue keeps growing for t_f time with a slope of $(f(Q(t - t_f)) - 1) \times R_l$. But unlike the step function, the slope now depends on the value of queue length. After t_f seconds, if the queue $Q(t) > Q_1$, the queue length starts decreasing with a slope of $(f(Q(t) - 1) \times R_l$. The slope now depends on the value of the queue length, so the there are no sudden changes in the slope. Therefore, the oscillations are fewer compared to the step function. If the system is near steady state, then the oscillations decrease, queue length reaches a value between Q_0 and Q_1, and the system reaches steady state.

4.3. Hyperbolic Function

The analysis for this case is similar to the above. If the h_a and h_b parameters are close to one (typical values are $h_a = 1.15, h_b = 1.05$), the hyperbolic function has similar behavior to the linear function. If h_a is high, then the hyperbolic function is close to the step function. Since the hyperbolic function has a larger curvature initially and then smoothes out, $f(Q_1 + \delta Q)$ value will be smaller than the corresponding $f(Q_1 + \delta Q)$ value obtained using the linear function. Hence, the fluctuations in the rates are more, but the queue draining is faster.

4.4. Inverse Hyperbolic Function

The behavior of the system is the same as in the case of the hyperbolic function for underload and the steady state range of queue lengths. For overload, $Q_2 \leq Q\infty$, the queue decreases linearly with slope $QDLF$. For mild overload, $Q_1 \leq Q \leq Q_2$, the queue decreases hyperbolically (inverse function of f(Q)'s a-curve). Since the inverse hyperbolic function is continuous and smooth near Q_1, it gives rise to less oscillations compared to other cases.

5. Simulation Configuration and Parameters

This section describes the two configurations used in the simulations. In all simulations, no higher priority (CBR and VBR) traffic is present.

5.1. Simple Configuration

In this configuration (figure 1), N infinite sources send data to N destinations. The traffic is unidirectional from source to destination. The initial values of ICR are chosen randomly in the range $(0, R_{link})$. All links are of length 1000 km, which corresponds to a propagation delay of 5 ms. All links have a bandwidth of 149.76 Mbps (after accounting for SONET overhead). The sources start transmission at a random time between $(0, t_{RTT})$, where t_{RTT} is the round trip time. t_{RTT} = 30 ms for this above configuration.

5.2. Generic Fairness Configuration-2 (GFC-2)

This configuration was used to test the performance of the queue control functions and the switch scheme under more stressful conditions. The value of the link distance D was chosen to be 1000 km. In this configuration the, expected max-min fairness rate for the different VC's are: A VCs = 10 Mbps, B VCs = 5 Mbps, C VCs = 35 Mbps, D VC = 35 Mbps, E VCs = 35 Mbps, F VC = 10 Mbps, G VCs = 25 Mbps, H VCs = 52.5 Mbps. This configuration is explained in detail in [8].

6. Simulation Results

In this section, the simulation results using the above two configurations are given. The rates and queue lengths are shown for each configuration, and are used to study and compare the performance of different queue control functions. In the simulations for both configurations, $QDLF$ was chosen to be 0.5.

Table 2. Simple Configuration: Results

Queue	a	b	Q_1	Q_2	Conv	Avg
Control	val	val			(secs)	QLen
Simple	0.75	1.01	4	26	-	252.93
	0.90	1.01	4	26	-	98.04
	0.90	1.05	4	26	-	663.63
	0.95	1.01	4	26	-	251.51
	0.95	1.05	4	26	-	124.11
	0.95	1.01	2	26	-	896.90
	0.95	1.01	8	26	-	483.20
Linear	1/16	1/16	2	26	0.20	311.85
	1/16	1/16	4	26	0.32	403.52
	1/16	1/16	8	26	0.61	402.85
Hyperbolic	1.15	1.05	2	26	-	509.94
	1.15	1.05	4	26	0.32	214.19
	1.15	1.05	8	26	0.82	220.96
Inv. Hyperbolic	36	1.05	2	26	0.22	313.17
	16.5	1.05	4	26	0.25	209.50
	6.75	1.05	8	26	1.12	202.27

6.1. Simple Configuration

Table 2 shows the performance for different step values of the step queue control function as the queue threshold Q_1 is varied. Note that Q_2 is fixed given $QDLF$ and other parameters of the *linear* and *hyperbolic* functions. The number of sources is 3. The following can be observed from table2.

- The step function never converges: the values are fluctuating near the target values, so the standard deviation after one second is lower than the standard deviation in the first second.
- The linear, hyperbolic and inverse hyperbolic functions reach steady state after one second. As Q_1 increases, the convergence time increases.
- For $Q_1 = 2Q_0$, the linear function converged. The value of f(Q) for the hyperbolic function is less than that of the linear function, so the queue is drained faster and Q becomes less than Q_0. Therefore, for the hyperbolic function, the queue length and rate values oscillate near the target value. This is mainly because the hyperbolic function is not smooth (not differentiable) near Q_1. The inverse hyperbolic is smooth at Q_1 so the queue lengths do converge in this case.
- For $Q_1 = 8Q_0$, the convergence time for the hyperbolic and inverse hyperbolic functions is more than the linear function. This happens because both the hyperbolic and inverse hyperbolic functions vary slowly near the steady state. The queue length variance is more for the inverse hyperbolic function.

Graphs 2(a), 2(c), 3(a), and 3(c) show the ACR rate of the three sources using step, linear, hyperbolic and inverse hyperbolic queue control functions respectively. Graphs 2(b), 2(d), 3(b), and 3(d) show the corresponding queue lengths.

The mean and standard deviation of the rates and the queue lengths are calculated for every 100 milliseconds. In steady state, the oscillations are small, so the standard deviation is small compared to mean. The quantity (mean + standard deviation) has a value close to the mean in the steady state. The convergence time was calculated based on the (mean + standard deviation) graphs (given in [9]).

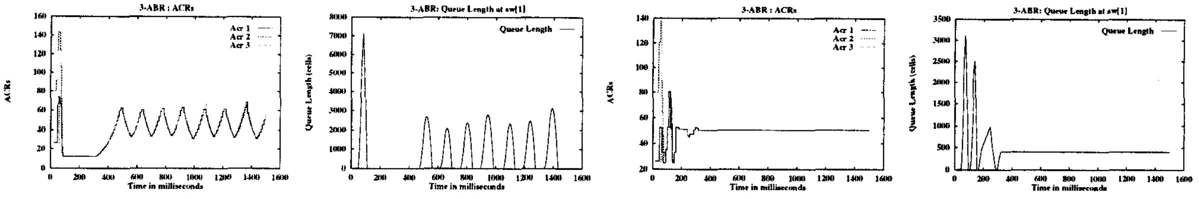
For the step function, there is oscillation in both rates and queue length. For linear and hyperbolic functions, the oscillations die down and the system reaches steady state. In steady state, the rate and queue length are constant and utilization is 100%. Hence, the *linear*, *hyperbolic* and *inverse hyperbolic* queue control functions fulfill the desired goal. This is consistent with the analytical explanation given in the previous section. In all the cases when the queue length and the rates converge, the queue length is non-zero and hence the utilization at steady state is 100%.

6.2. GFC-2 Configuration

The following parameters were used in the simulations for this configuration. The thresholds used were $Q_0 = 176$, $Q_1 = 4 \times Q_0$, $Q_2 = 26 \times Q_0$, $QDLF = 0.5$. The values of $a - param$ and $b - param$ were, for the step function, $s_a = 0.95$, $s_b = 1.01$; for the linear function $m_a = 1/16$, $m_b = 1/16$; for the hyperbolic function $h_a = 1.15$, $m_b = 1.05$; and for the inverse hyperbolic function $\lambda_a = 16.5$, $\lambda_b = 1.05$.

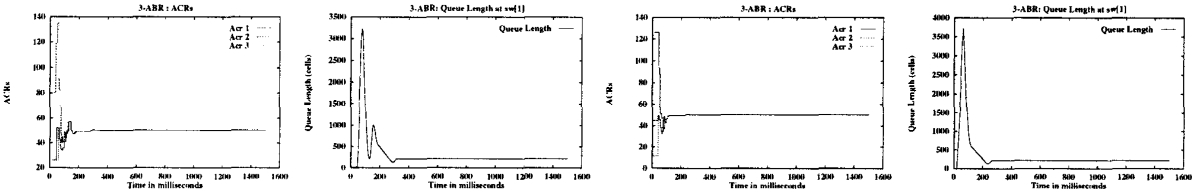
Table 3 shows the performance for the four queue control functions. The table shows the $H(1)$ VC's mean rate, switch queue length for SW5 and its convergence time, and standard deviation before one second and after one second. The queue length variation is present in all the three cases. The rate variation is much less in *linear* and *hyperbolic* and $inversehyperbolic$ functions compared to *step* function.

Figures 4 and 5 were obtained by simulating the GFC-2 configuration using the step, linear and hyperbolic and inverse hyperbolic queue control functions. Graphs 4(a), 4(c), 5(a) and 5(c) show the ACR rate for one VC of each of A through H type VCs versus time when different queue control functions are used. From these graphs, it can be seen that the expected rates are obtained when linear, hyperbolic and inverse hyperbolic functions are used for queue control. The (b) and (d) graphs have the queue length for all the switches. The maximum queue is due to the initial overload,



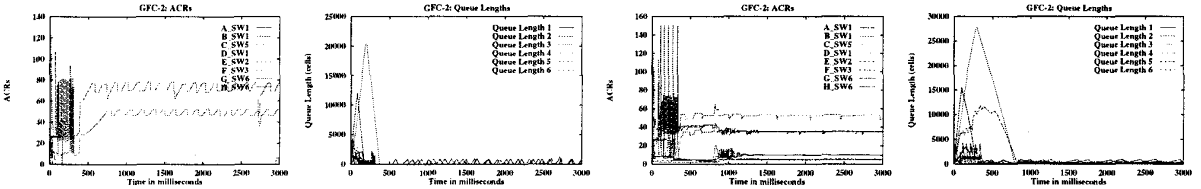
(a) ACR (b) Queue Length (c) ACR (d) Queue Length

Figure 2. Simple Configuration: Simple and Linear queue control functions



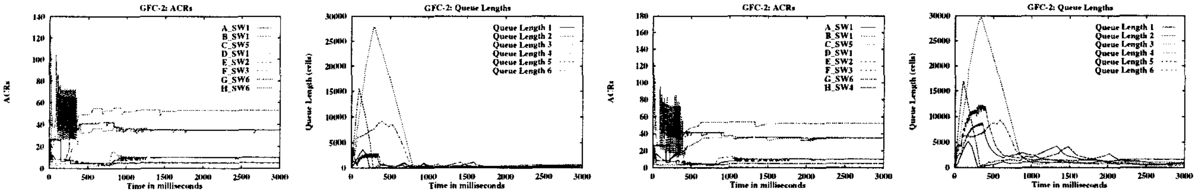
(a) ACR (b) Queue Length (c) ACR (d) Queue Length

Figure 3. Simple Configuration: Hyperbolic and Inv. Hyperbolic queue control function



(a) ACR (b) Queue Length (c) ACR (d) Queue Length

Figure 4. GFC-2 Configuration: Step and Linear queue control function



(a) ACR (b) Queue Length (c) ACR (d) Queue Length

Figure 5. GFC-2 Configuration: Hyperbolic and Inv. Hyperbolic queue control function

Table 3. GFC-2 Configuration: Results

Queue Control Step	Quantity	Convg. time(secs)	Mean
	H(1) ACR	-	72.81
	SW5 Queue	-	284.28
Linear	H(1) ACR	1.25	52.46
	SW5 Queue	1.3	455.46
Hyper	H(1) ACR	1.45	52.77
	SW5 Queue	1.3	361.32
Inv.Hyper	H(1) ACR	1.51	52.38
	SW5 Queue	2.0	1443.72

before the first round trip time. Once the feedback control loop is established, the $f(Q)$ value is $QDLF$ and queues are drained quickly.

When the step function (figure 4(a)) is used, the oscillations are more compared to the oscillations when other functions are used. Some of the VCs do not get their max-min fair share rates and the VCs near the fair share have considerable oscillations. The step function is very sensitive to queue length variation near the thresholds. Since the configuration is complex with large number of VCs passing through each of the switches, the queue length and hence the rates vary. For the graphs 4(c), 5(a) and 5(c) the oscillations are only present before steady state. The oscillations die down and the rates become steady since the function $f(Q)$ changes smoothly. The maximum queue length is same for all queue control functions since this depends only on the ICR. When the inverse hyperbolic function is used, the queues are larger since in this case the steady state queue length is near Q_1.

The simulation results obtained by using different queue control functions in the simple and the GFC-2 configurations are consistent with the analytical explanation. The step function is sensitive to queue thresholds (Q_0, Q_1, Q_2). The other functions are not sensitive to these queue thresholds. Small steady state queuing delay can be achieved by choosing nearby values for Q_0 and Q_1.

7. Conclusion

In this paper, we have considered the problem of designing a simple and robust queue control function for switch schemes. A switch scheme tries to maximize utilization, minimize queuing delay and give max-min fair rates to the sources. It is also desirable to have fewer oscillations in rates and queue lengths to support (low quality) video over ABR service. We assume a switch scheme model which dynamically adjusts the rate of the sources to match the output rate and drain large queues. The design considerations were discussed with analytical explanations. Four different queue control functions were analyzed. The choice of parameters for the queue control functions was both explored analytically and by simulation. The simulations showed that even in complex configurations (like GFC-2), the system behavior was consistent with the analytical explanation. When the step function is used, the system oscillates and does not converge in most cases. Both the analytical and the simulation results show that the inverse hyperbolic is the best queue control function, followed by the hyperbolic and the linear queue control functions. For lower implementation complexity, the linear function is recommended.

Acknowledgments

This research was sponsored in part by Rome Laboratory/C3BC Contract #F30602-96-C-0156.

References

[1] Y. Afek, Y. Mansour, and Z. Ostfeld. Phantom: A simple and effective flow control scheme. In *Proc. of the ACM SIGCOMM*, pages 169–182, August 1996.

[2] A. Arulambalam, X. Chen, and N. Ansari. Allocating fair rates for available bit rate service in atm networks. 34(11):92–100, November 1996.

[3] A. W. Barnhart. *Example Switch Algorithm for TM Spec.* ATM Forum/AF-TM 95-0195, 1995.

[4] D. Chiu and R. Jain. Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN*, 17(1):1–14, June 1989.

[5] S. Kalyanaraman, R. Jain, R. Goyal, S. Fahmy, and B. Vandalore. The erica switch algorithm for abr traffic management in atm networks. *Submitted to* IEEE/ACM Transactions on Networking, November 1997.

[6] L. Roberts. *Enhanced PCRA (Proportional Rate Control Algorithm).* ATM Forum/AF-TM 94-0735R1, 1994.

[7] S. S. Sathaye. *ATM Forum Traffic Management Specification Version 4.0.* Draft Specification ATM Forum/95-0013R11, ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.0000.pdf, 1996.

[8] R. J. Simcoe. *Test configurations for fairness and other tests.* ATM Forum/AF-TM 94-0557, 1994.

[9] B. Vandalore, R. Jain, R. Goyal, and S. Fahmy. Design and analysis of queue control functions for explicit rate switch schemes. Technical Report OSU-CISRC-8/98-TR35, The Ohio State University, 1998.