
ATM Forum Document Number: ATM Forum/97-1087

Title: Design and Analysis of Queue Control Function for Switch Schemes

Abstract:

The main goals of a switch scheme are high utilization, low queuing delay and fairness. To achieve high utilization the switch scheme can maintain non-zero (small) queues in steady state which can be used if the sources do not have data to send. It is very important to design and analyze the queue control function which is used in such a scheme. In this contribution we study various queue control functions and present analytical explanation of its behavior and simulation results. From the study, we conclude that a simple linear queue control function performs satisfactorily.

Source:

Bobby Vandalore, Raj Jain, Rohit Goyal, Sonia Fahmy
The Ohio State University Department of Computer and Information Science

Raj Jain is now at Washington University in Saint Louis, jain@cse.wustl.edu <http://www.cse.wustl.edu/~jain/>

The presentation of this contribution at ATM Forum is sponsored by NASA Lewis Research Center.

Date: December 1997

Distribution: ATM Forum Technical Working Group Members (AF-TM)

Notice:

This contribution has been prepared to assist the ATM Forum. It is offered to the Forum as a basis for discussion and is not a binding proposal on the part of any of the contributing organizations. The statements are subject to change in form and content after further study. Specifically, the contributors reserve the right to add to, amend or modify the statements contained herein.

1 Introduction

The goals of rate allocation schemes are maintaining high utilization, small queue delay and cell loss and fairness among competing sources. In order to support (low quality) video sources over ABR (Available Bit Rate) service, it is also desirable that in steady state the rates and queuing delay be constant.

One way to achieve high utilization and low queuing delay is to vary the target rate as a function of queue length. The function should be a decreasing function of queue length. The function should also be *simple* so that it can be implemented in hardware.

In this contribution, we study several *queue control* functions which satisfy the above needs. We present analytical explanation for performance of these functions. Then we present simulation results which are consistent with the analysis. The various trade-offs between the queue control functions is studied using appropriate metrics. The *ERICA+* [6] switch scheme is used in the simulation.

2 Switch Scheme Model

There are many ABR switch schemes ([1, 2, 3, 4, 6]) This section gives an overview of the switching scheme model on which this study is based.

- An ABR switch scheme achieves the goals by giving explicit feedback to the sources to adjust their source rates. These are usually known as *Explicit Rate Feedback* switches. The other common switch model is the *Explicit Forward Congestion Indication (EFCI)* switch. We assume that an Explicit Rate Feedback switch is used.
- One way to achieve high utilization (100%) and control queuing delay by quick draining of queues is to vary the target ABR rate dynamically. During steady state, the target ABR rate is 100% while it is lower during transient overloads. Higher overloads result in even lower target rates (thereby draining the queues faster). In other words:

$$\text{Target rate} = f(\text{queue length}) \times \text{function}(\text{current rate, link rate, HPR rate})$$

The HPR rate is the total rate of higher priority classes like VBR (variable bit rate) and CBR (constant bit rate). The “f(queue length)” has to be a decreasing function of the queue length. The switch scheme uses the above queue control function to adjust the allocated rate depending on the current switch queue size.

- The switch measures the load, queue length and gives explicit feedback of target rate at fixed intervals. This interval is called the “averaging interval”. The measurements are done using the *FRM* cells and the feedback is given using the *BRM* cells. We assume that only one feedback is given in each averaging interval to the sources. This avoids unnecessary conflicting feedbacks to the sources.

The *ERICA+* algorithm used in this study fits the above model.

3 Queue control functions

In this section the relationship between the queue length and queue control function is presented for the above switch model. Then various queue control functions to achieve the desirable goals are presented.

The following terms are used in the discussion:

N number of sources.

t_s “averaging interval”, the period at which feedback to the sources is calculated at the switch.

$CCR_i(t)$ current rate of source i .

$ACR_i(t)$ allowed cell rate calculated at switch.

t_p propagation time from the source to switch.

t_f feedback delay is twice t_p .

R_l link rate (for simplicity, assume all links have same rate)

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

$R_i(t)$ aggregate *input rate* seen at switch. $R_i(t) = \sum_{i=1}^N CCR_i(t)$

$C(t)$ (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.

Note : $X(t)$ denotes that X is a function of time.

3.1 Queue Length Function

The current rate is seen at the switch after t_p time, so $CCR_i(t - t_p)$ is rate of source i seen at the switch. The sources adjust their rates based on the feedback information of the switches, ie., $CCR_i(t) = ACR_i(t - t_p)$.

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

$$Q(t) = Q(t - t_s) + \left(\sum_{i=1}^N CCR_i(t - t_p) - R_l \right) C(t_s)$$

$$Q(t) = Q(t - t_s) + \left(\sum_{i=1}^N ACR_i(t - t_f) - R_l \right) C(t_s)$$

$$Q(t) = Q(t - t_s) + (R_i(t) - R_l)t_s$$

The switch scheme tries to adjust the input rate $R_i(t)$ to match output rate depending on current queue size, ie., $R_i(t) = f(Q(t)) \times \text{Available ABR Capacity}$, if we assume no HPR then $R_i(t) = f(Q(t)) \times R_l$. Hence,

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

and $(f(Q(t - t_s)) - 1)R_l$ is the rate at which the queue changes.

3.2 Explicit Rate

The sources adjust their rates $ACR(t)$ based on explicit rate feedback from the switch. The source rates lag from the explicit rate by T_f . Hence $ACR(t)$ (source rate) can be expressed using the following function;

$$ACR(t) = f(Q(t - t_f)) \times F(ACR(t - t_f), \text{Link Rate}, \text{HP rate})$$

For simplicity we assume there is no HPR traffic (Note in the presence of bursty VBR sources there might not be any steady state of the system). So the above function becomes

$$ACR(t) = f(Q(t - t_f)) \times F(ACR(t - t_f), \text{Link Rate})$$

For our *ERICA+* scheme the above function is as follows

$$ACR(t) = f(Q(t - t_f)) \times \max\left(\frac{ACR(t - t_f) \times \text{Link Rate}}{\text{Input Rate}}, \frac{\text{Link Rate}}{n}\right)$$

where *Input Rate* is the ABR input rate measured at the switch. The other terms used in *ERICA+* are ignored since this is the only term which has the queue control function. The scheme tries to match the input rate to the link rate, by over allocating the rates if the queue is small. If queues are large then they are drained quickly by using part of the link capacity. The function $f(Q)$ is a fraction which modifies the link rate to achieve the above.

3.3 Design of Queue Control Function

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

- If queue length is very small it should be increased, so that the scheme can maintain some small queue which can be used when link is under utilized. This implies that $f(Q)$ should be greater than one.
- In steady state we desire constant queue length and target rate to be the max-min fairness rate. The function $Q(t)$ satisfies this goal if $f(Q) = 1$ in steady state.
- If queue is large then part of the link capacity is 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 $f(Q)$.

- The $f(Q)$ function has to be continuous. Discontinuities imply sudden changes which give rise to oscillations.

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 following three functions are possible candidates.

Step function

The step function has multiple thresholds (See figure 1). This is most simplest to implement in hardware (lookup table).

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

where $s_a > 1$ and $\text{QDLF} < s_b < 1$ are step parameters. In general it can have n steps. In the above case $n = 4$.

Linear function

The fraction $f(Q)$ has linear relationship with queue length. (See figure 1)

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

where m_b and m_a are slope of the linear portions. This 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 .

Hyperbolic function

The fraction $f(Q)$ is a hyperbolic function of the queue length. (See figure 1)

$$f(Q) = \begin{cases} = \frac{h_b Q_0}{(h_b - 1)Q + Q_0} & 0 \leq Q \leq Q_0 \\ = 1 & Q_0 < Q \leq Q_1 \\ = \frac{h_a Q_1}{(h_a - 1)Q + Q_1} & Q_1 < Q \leq Q_2 \\ = \text{QDLF} & Q_2 < Q < \infty \end{cases}$$

where h_a and h_b are parameters which control degree of curvature of the hyperbolic function. This function takes more time to calculate, since it has a division operation. For high value of h_a the hyperbolic function becomes similar to step function. For h_a value near 1, the hyperbolic function approaches the linear function.

Note: $f(Q_2) = QDLF$, so Q_2 can be expressed in terms of $QDLF$ and a parameter in the case of linear and hyperbolic functions.

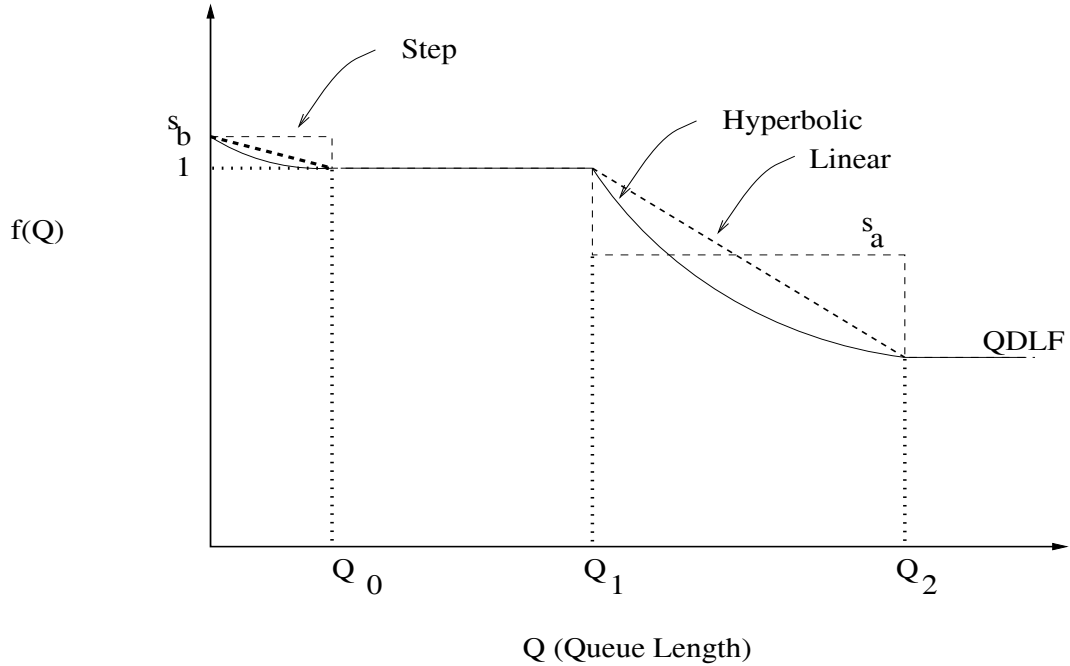


Figure 1: Queue Control Functions

4 Metrics

To compare the performance of the queue control function the following metrics are chosen.

Convergence Time: The time the scheme takes to converge to steady state. To find the convergence time, the variance and standard deviation of desired variable are calculated between $(i \times t_k, (i+1) \times t_k)$ for $i = 0, 1, \dots$, where t_k ($= 100\text{ms}$) 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) value of the variable versus time are plotted. From the graph the convergence time can be calculated.

Standard Deviation: The standard deviation of various quantities like ACRs, queue length and utilization is calculated. In order to separate the oscillations before steady state from affecting the measurement, the variance is measured both before and after steady state is achieved.

Visual inspection of the graphs also gives a good idea about the convergence time and the variations.

5 Analytical Explanation

In this section we analyze the behavior of the proposed queue control functions. We assume a simple configuration in our analysis. N infinite ABR sources (always has data to send) are sending data to N ABR destinations (See figure 2). The performance study under more stressful conditions is done by simulation using the Generic Fairness configuration - 2 [7] in the simulations section.

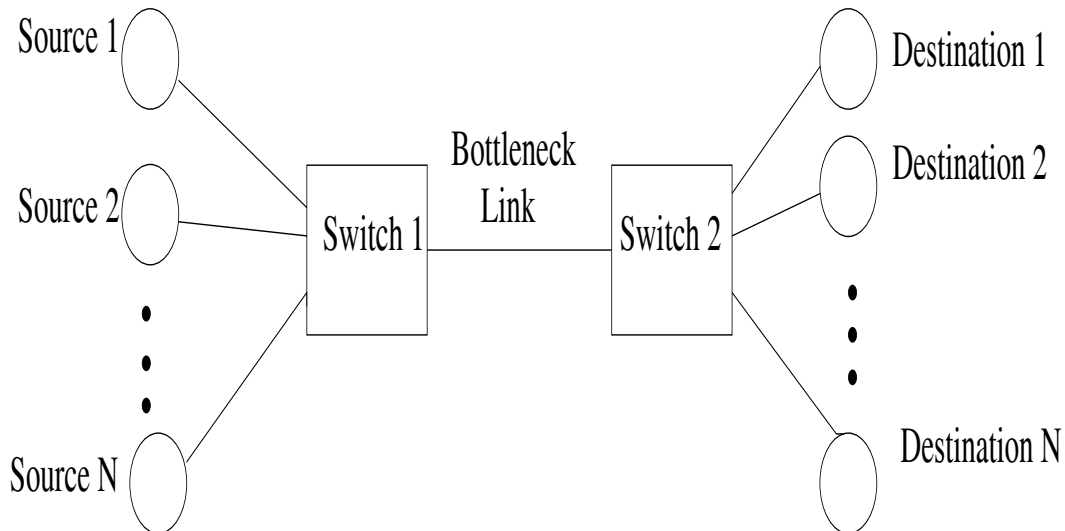


Figure 2: N Sources - N Destinations Configuration

In the beginning, the queue lengths grow depending on the initial ICR (initial cell rate). 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. The switch initially estimates that the link is under utilized, so it asks the source to increase their rates. But this gives rise to overloaded condition and increases the switch queue lengths. When the queue length crosses Q_2 the queues are quickly drained by using $(1-QDLF)$ fraction of link capacity. In the meantime the feedback control loop is established, and the switch gives reliable feedback to the sources. The feedback information tries to match the input rate to output rate. As the input rate approaches output rate the oscillations die down and the network reaches steady state. In steady state the rates and the queue lengths remain constant.

This behavior of the system is independent of the queue control function used, since all of them have $f(Q) = QDLF$ when $Q(t) > Q_2$. So, in this analysis we assume that the initial convergence period is over and the network is near the steady state.

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

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

5.1 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 asking them to decrease their rate. The queue grows for t_f time and it can be expressed as follows:

$$Q(t) = Q(t - t_f) + (s_b - 1) \times R_l \lfloor \frac{t_f}{t_s} \rfloor C(t_s)$$

if t_f is a multiple of t_s the above simplifies to

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

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

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

Therefore for the system to achieve the steady state the parameter Q_0 , should be small and Q_1 should be such that $Q_1 > Q_0 + (s_b - 1) \times R_l C(t_f)$ is satisfied. Since step function has discontinuities, it is very sensitive to queue length value near the thresholds and steady state might not be reached if the parameters are not set properly. If parameters are not set properly, then the queue grows from a value below Q_0 for t_f time crosses Q_1 and decreases for t_f time to a value less than Q_0 and this pattern repeats.

5.2 Linear Function

If $Q(t - t_f) < Q_0$, then $f(Q(t)) > 1$. Similar to the step function the queue keeps growing for t_f time with 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 there are no sudden changes in the slope. Therefore the oscillations are less compared to the step function. If the system is near steady state, then the oscillations decrease, queue length becomes Q_1 and system reaches steady state.

5.3 Hyperbolic Function

The analysis for this case is similar to above. If h_a and h_b parameter are close to one (typical values are $h_a = 1.15, h_b = 1.05$) the hyperbolic function has similar behavior as the linear function. If h_a is high then the hyperbolic function is close to the step function. Since hyperbolic function has a larger curvature initially and then smooths out, $f(Q)$ value will be smaller when Q_1 threshold is crossed compared to the linear function. Hence the fluctuations in the rates are more, but the queue draining is faster.

6 Simulation: Configuration and Parameters

In this section the two configurations used in the simulations are explained.

6.1 Simple Configuration: N Source - N Destinations

In this configuration: (See figure 2)

- N infinite sources send data to N destinations
- The traffic is one way
- The initial value of ICR are chosen randomly in the range (0,link rate)
- All links are of length 1000 Km, which corresponds to a propagation delay of 5 ms at 149.76 Mbps
- All links have a bandwidth of 149.76 Mbps (after accounting for SONET overhead).
- The sources start at random time between (0, t_{RTT}), where t_{RTT} is the round trip time. $t_{RTT} = 30$ ms for the above configuration.

6.2 Generic Fairness Configuration - 2 (GFC-2)

This configuration (See figure 3) was used to test the performance of the queue control functions and the switch scheme under more stressful conditions. The value of link distance D was chosen to be 1000 Km. This configuration and the expected max-min fairness rate for the different VC's are given in [7]

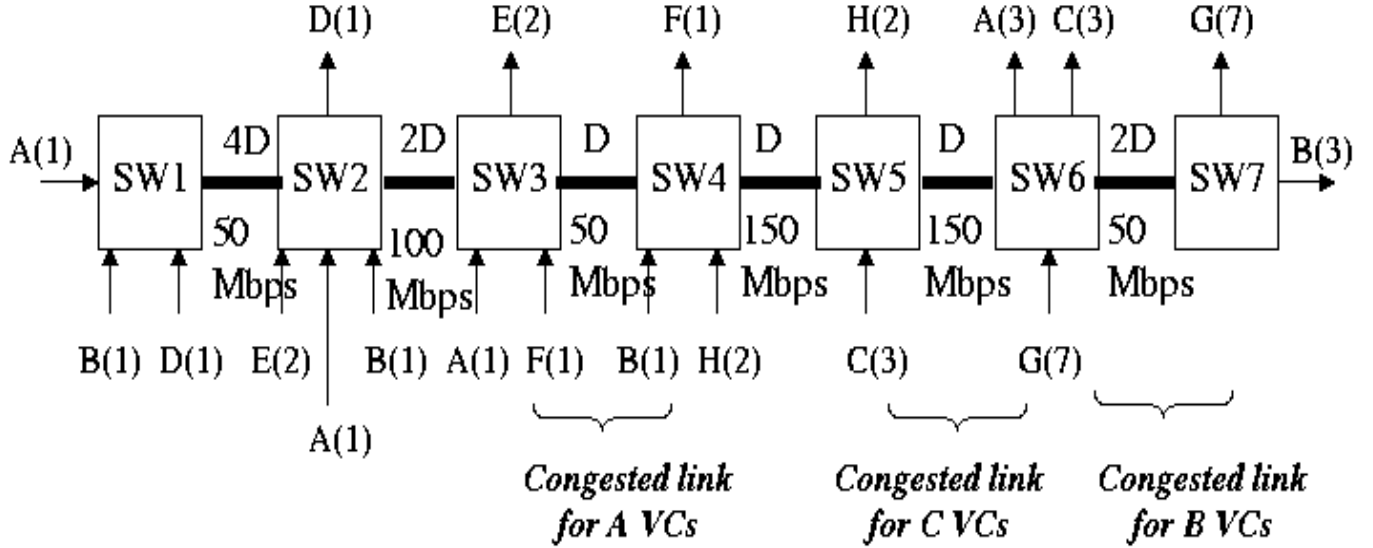
7 Simulation: Results

In this section the simulation results using the above two configurations are given. The graphs of rates, queue length and utilization are given. The tables and the graphs are used to study the performance of different queue control functions. In the simulations for both configurations, $QDLF$ was chosen to be 0.5.

7.1 Simple Configuration: Results

The table 1 shows the performance for different step values (parameters) of the step queue control functions as the queue threshold Q_1 is varied. The mean bottleneck link queue length, its standard deviation before one second and after one second (last two columns) are shown in the table. Note that Q_2 is fixed given the $QDLF$ and other parameters of the *linear* and *hyperbolic* functions. Number of sources $N = 3$.

The following things can be observed from the table 1.



Note: Entry/exit links of length D , speed 150 Mbps

Figure 3: Generic Fairness Configuration - 2

- The step function never converges entirely. 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 and hyperbolic function reach steady state. The standard deviation after one second is very small.
- As Q_1 increases the convergence time increases for linear and hyperbolic functions
- For $Q_1 = 2Q_0$, the linear function converged. The value of $f(Q)$ for hyperbolic function value is less compared to that of 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 are oscillating near the target value.
- For $Q_1 = 8Q_0$, the convergence time for hyperbolic function is more than linear.

The graphs 4(b), 5(a), 6(a) show the ACR rate of the three sources.

The mean and standard deviation of the rates and the queue lengths are calculated for every 100 milliseconds. These are shown figures 4(b), 5(b), 6(b) for VC rates and in figures 4(d), 5(d), 6(d) for the queue lengths. From these graphs the converging time can be estimated. In steady state the oscillations are small, the standard deviation is small compared to mean. So the quantity (mean + standard deviation) has value close to the mean in the steady state.

Table 1: Simple Configuration: Results

Queue Control	a param	b param	Q_1	Q_2	Convg time(secs)	Mean Q(cells)	Std Dev (bef 1 sec)	Std Dev (after 1 sec)
Step	0.75	1.01	4 Q_0	26 Q_0	-	252.93	552.21017	501.60
	0.90	1.01	4 Q_0	26 Q_0	-	98.04	651.82	241.43
	0.90	1.05	4 Q_0	26 Q_0	-	663.63	1226.70	840.36
	0.95	1.01	4 Q_0	26 Q_0	-	251.51	816.62	393.26
	0.95	1.05	4 Q_0	26 Q_0	-	124.11	805.32	240.04
	0.95	1.01	2 Q_0	26 Q_0	-	896.90	1386.87	1036.66
	0.95	1.01	8 Q_0	26 Q_0	-	483.20	1001.54	644.73
Linear	1/16	1/16	2 Q_0	26 Q_0	0.20	311.85	335.61	0.69
	1/16	1/16	4 Q_0	26 Q_0	0.32	403.52	457.90	0.69
	1/16	1/16	8 Q_0	26 Q_0	0.61	402.85	622.02	0.69
Hyperbolic	1.15	1.05	2 Q_0	26 Q_0	-	509.94	423.89	205.65
	1.15	1.05	4 Q_0	26 Q_0	0.32	214.19	500.14	0.86
	1.15	1.05	8 Q_0	26 Q_0	0.82	220.96	862.25	0.63

The (e) graph shows the utilization for the bottleneck link.

For the step function there is oscillation in all the quantities (rates, queue and utilization). 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* and *hyperbolic* queue control function fulfill the desired goal. This is consistent with the analytical explanation given in the previous section.

7.2 GFC-2 Configuration: Results

The following parameters were used in the simulations for this configuration.

- Thresholds: $Q_0 = 176$, $Q_1 = 4 \times Q_0$, $Q_2 = 26 \times Q_0$, $QDLF = 0.5$
- Step: $s_a = 0.95$, $s_b = 1.01$
- Linear: $m_a = 1/16$, $m_b = 1/16$
- Hyperbolic: $h_a = 1.15$, $m_b = 1.05$

The table 2 shows the performance for three queue control functions. The table shows the $H(1)$ VC's mean rate, switch queue length for SW5 and its convergence time, 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* functions compared to *step* function. This is also evident from the graphs which are explained in the next section.

Table 2: GFC-2 Configuration: Results

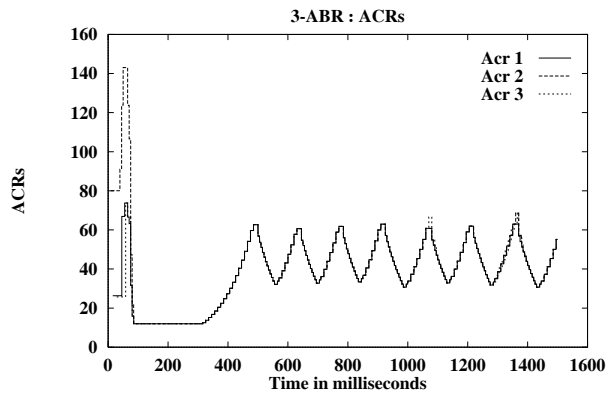
Queue Control	Quantity	Convergence Time (secs)	Mean	Std Dev (before 1 sec)	Std Dev (after 1 sec)
Step	H(1) ACR	-	72.81	18.4	4.46
	SW5 Queue	-	284.28	878.63	281.85
Linear	H(1) ACR	1.25	52.46	14.38	1.08
	SW5 Queue	1.3	455.46	1043.71	220.42
Hyperbolic	H(1) ACR	1.45	52.77	13.57	0.58
	SW5 Queue	1.3	361.32	968.27	201.86

7.3 GFC-2 Configuration: Graphs

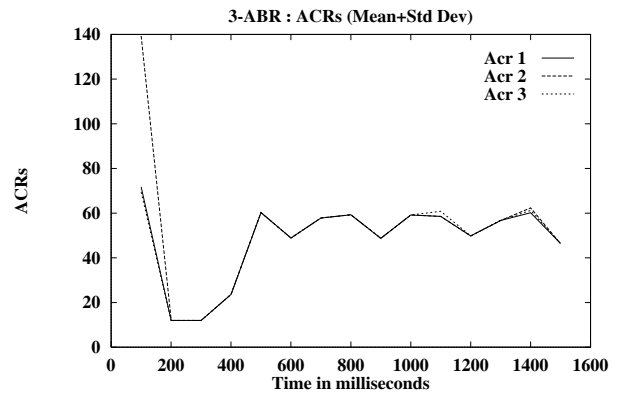
The graphs 7, 8, 9 were obtained by simulating the GFC-2 configuration using the *step*, *linear* and *hyperbolic* queue control functions respectively. Figures 7(a), 8(a), 9(a) show the ACR rate for one VC of each of A through H type VCs versus time.

The (b) graphs have the queue length for all the switches. The maximum queue is due to the initial overload, before the first round trip time. Once the feedback control loop is established the $f(Q)$ value is $QDLF$ and queues are drained quickly. Again in 7(b) oscillations when step function is used are more compared to the oscillations when other two functions are used. The graphs 7(c), 8(c), 9(c) plot mean plus standard deviation for VC rates. The figures 7(d), 8(d), 9(d) plot corresponding (mean+standard deviation) graphs for the queue lengths. The graphs 7(e), 8(e), 9(e) give the utilization of all the links between the switches.

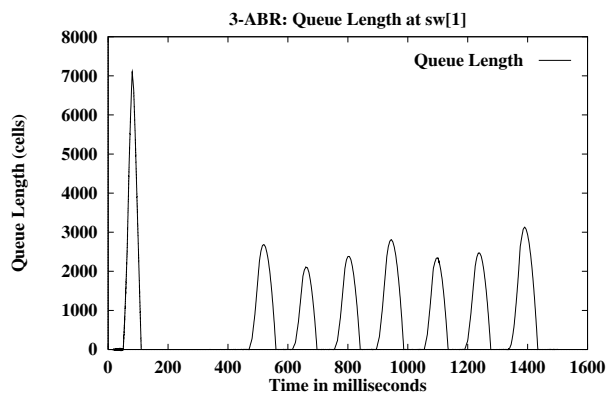
Note that in graphs when step function is used some of the VCs do not get their max-min fair share rates and the VCs near the fair share have considerable oscillations. The step function is very sensitive to queue length variation near the thresholds. Since the configuration is complex, with large number of VCs passing through each of the switches, the queue length and hence the rates vary. For the graphs 8(a),9(a) the oscillations are only present before steady state. The oscillations die down and the rates become steady since the function $f(Q)$ changes smoothly.



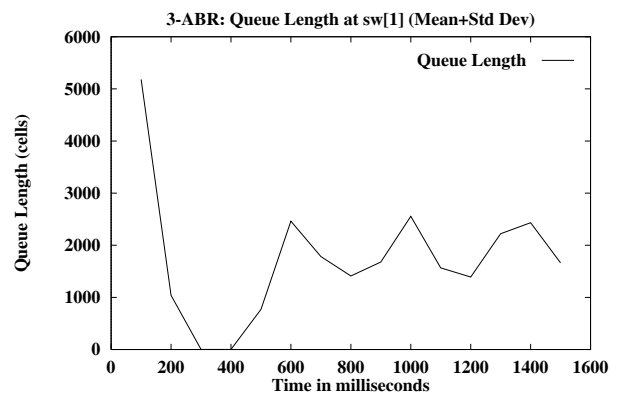
(a)



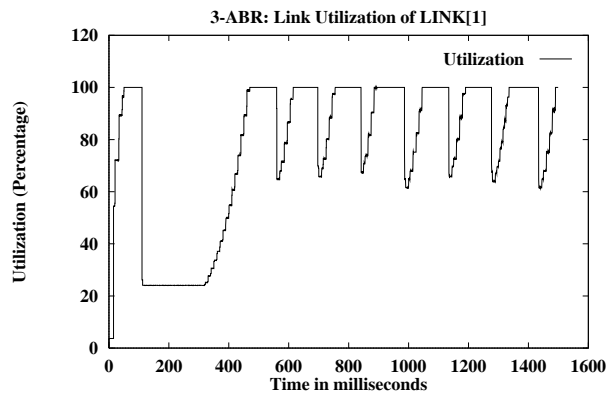
(b)



(c)



(d)



(e)

Figure 4: Simple Configuration: Rate, Queue and Utilization graphs: Step queue control function

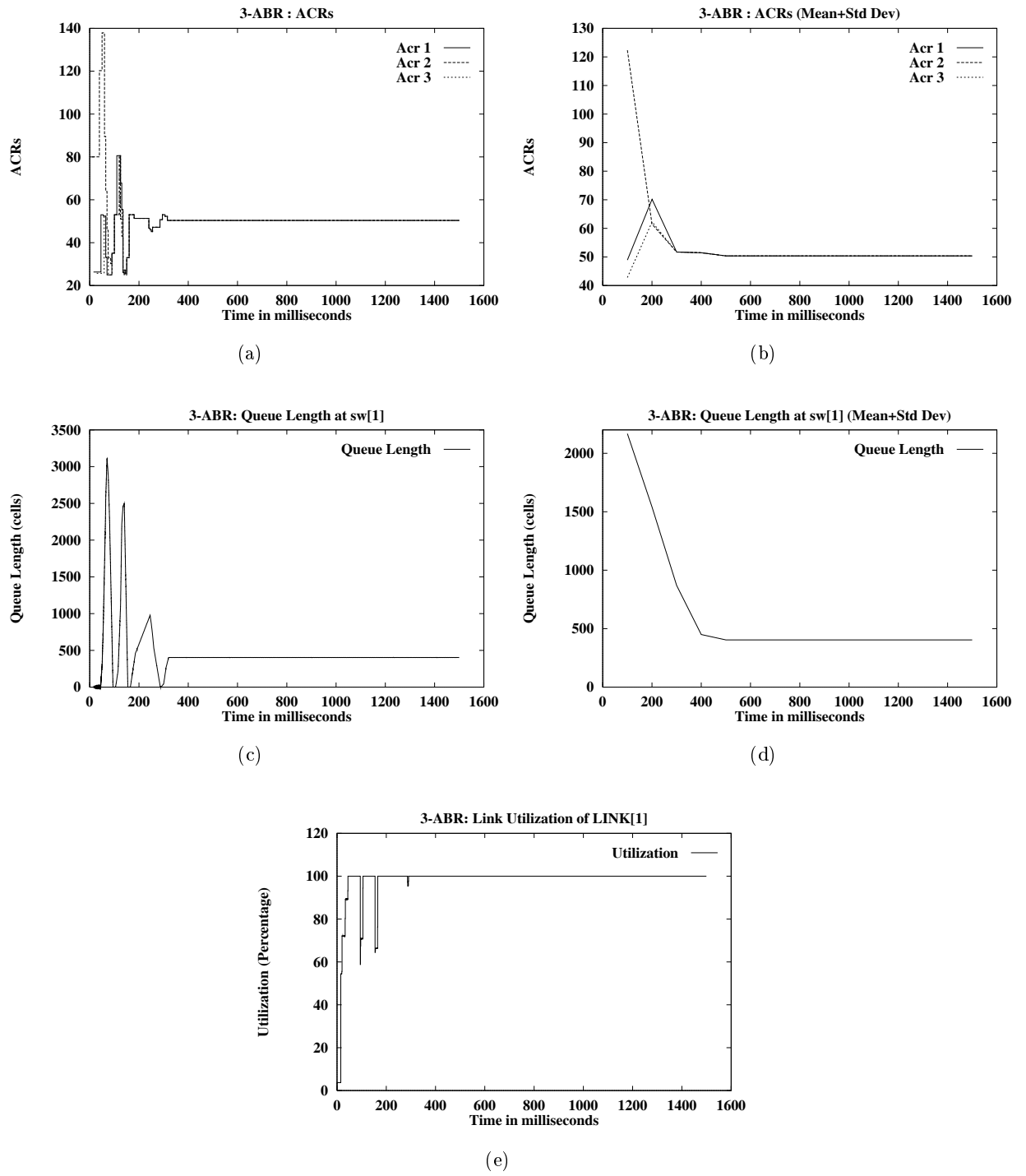
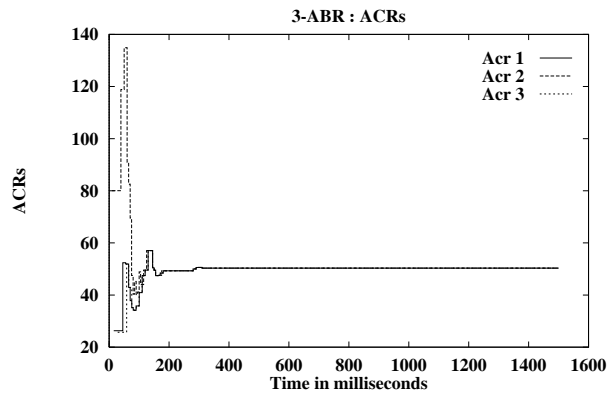
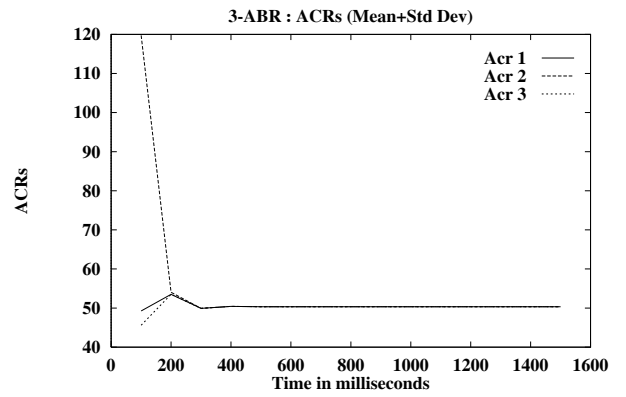


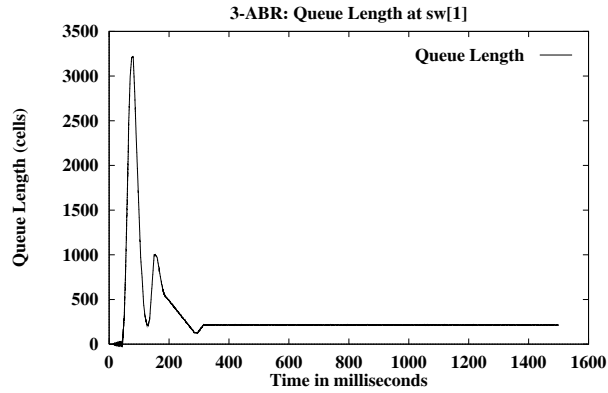
Figure 5: Simple Configuration: Rate, Queue and Utilization graphs: Linear queue control function



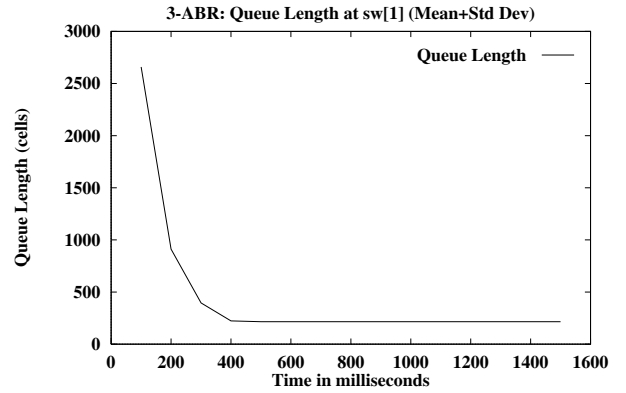
(a)



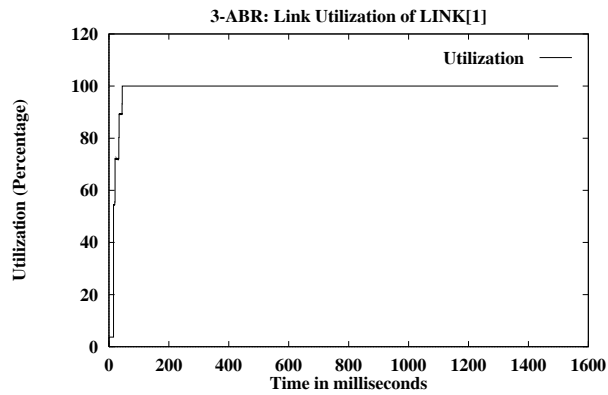
(b)



(c)



(d)



(e)

Figure 6: Simple Configuration: Rate, Queue and Utilization graphs: Hyperbolic queue control function

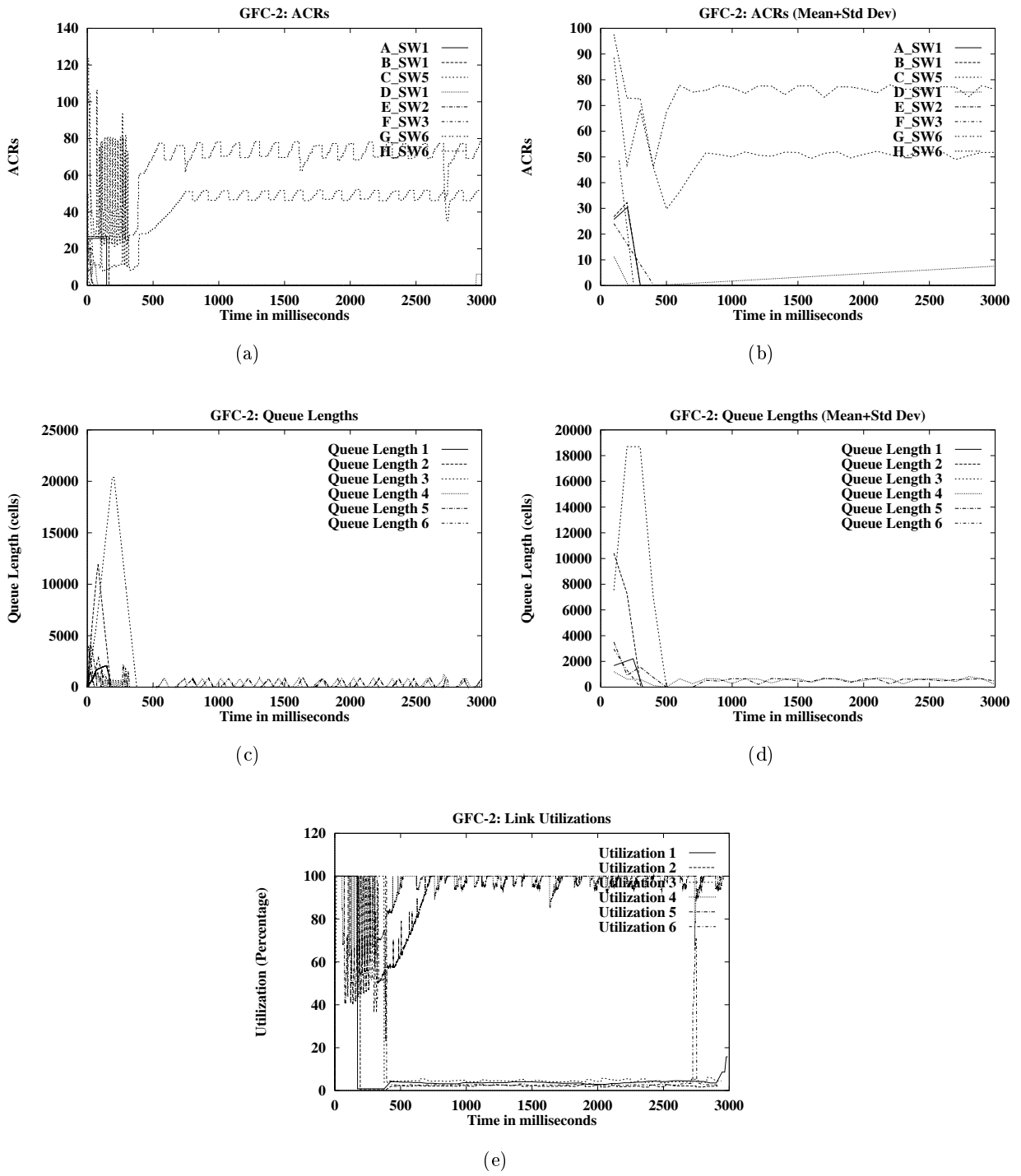


Figure 7: GFC-2 Configuration: Rate and Queue graphs for: Step queue control function

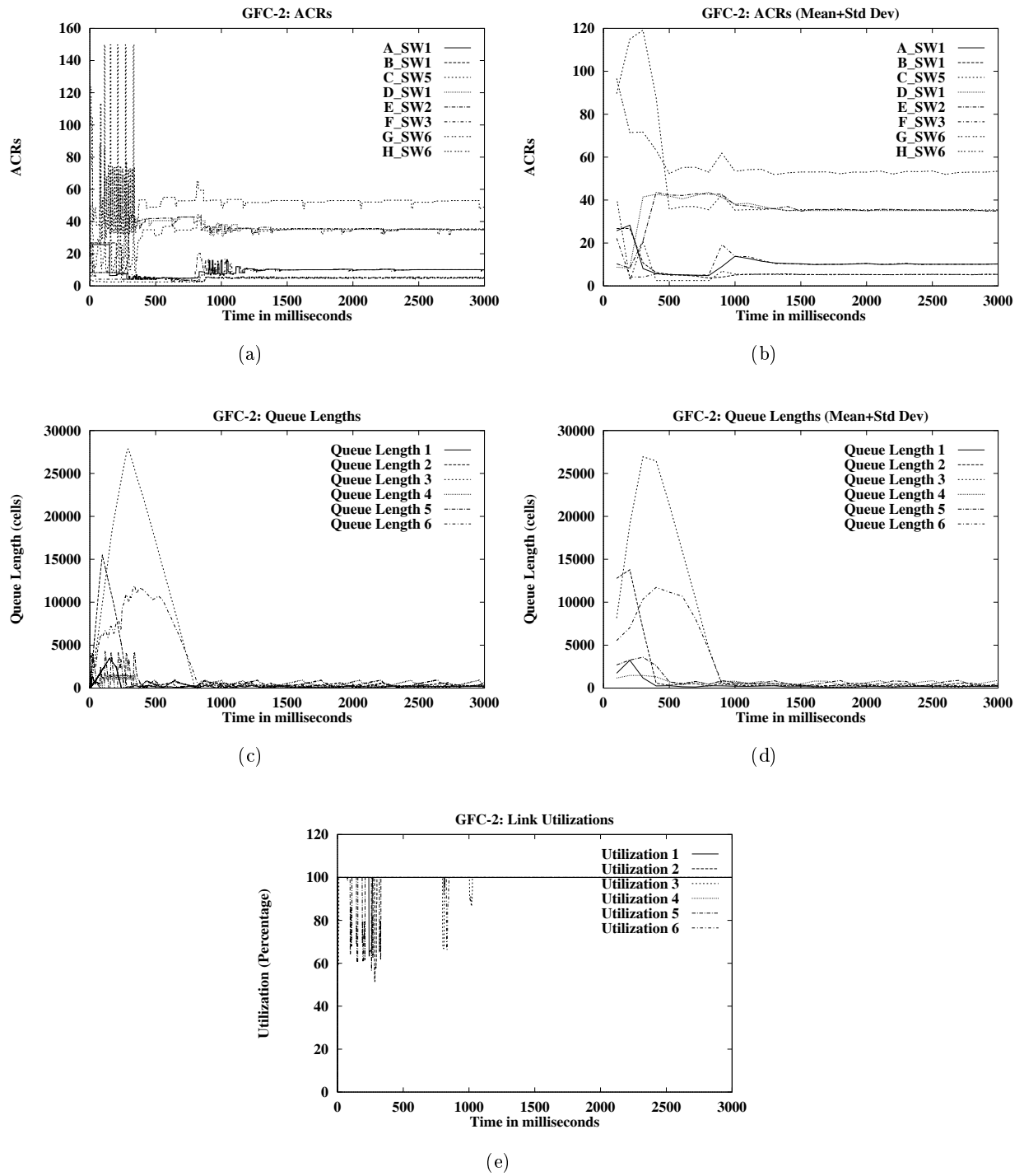


Figure 8: GFC-2 Configuration: Rate, Queue and Utilization graphs: Linear queue control function

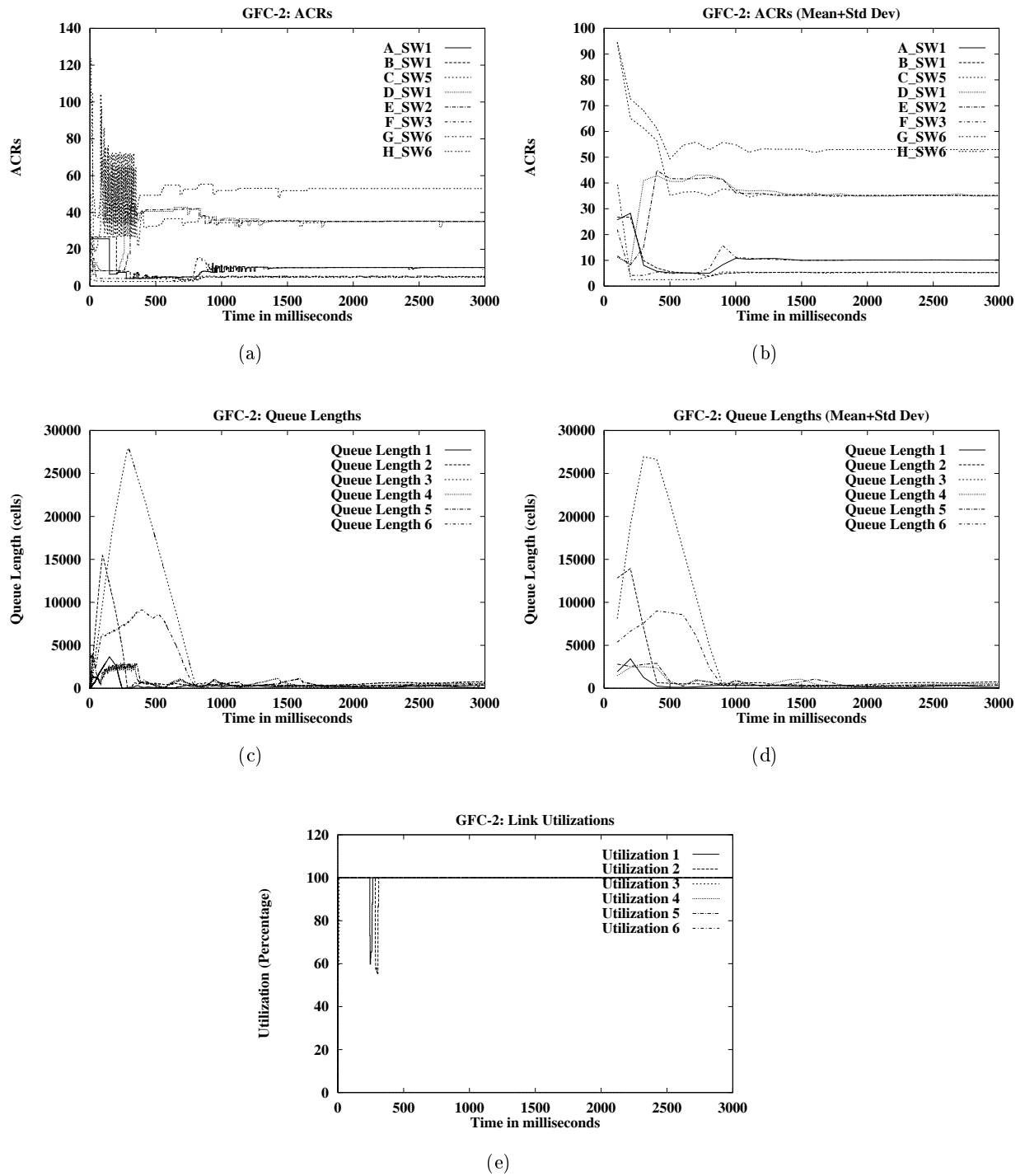


Figure 9: GFC-2 Configuration:: Rate, Queue and Utilization graphs: Hyperbolic queue control function

8 Conclusion

In this contribution we have considered the problem of designing a simple and good queue control function for switch schemes. A switch schemes tries to maximize utilization, minimize queuing delay and give max-min fair rates to the sources. It is also desirable to have less oscillations in rates and queue length 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. Three different queue control functions were considered. The choice of parameters for the queue control functions was explored analytically and by simulation. Simulation showed that even in complex configuration (like GFC-2) the system behavior was similar as explained analytically. Both from the analytical and the simulation results it can be concluded that the both hyperbolic and linear queue control function achieve the desired goals. For simpler implementation complexity the linear function is recommended.

References

- [1] Larry Roberts. Enhanced PRCA (Proportional Rate-Control Algorithm). ATM Forum 94-0735R1, August 1994.
- [2] Ambalavanar Arulambalam, Xiaoqiang Chen, and Nirwan Ansari. Allocating Fair Rates for Available Bit Rate Service in ATM Networks. *IEEE Communications Magazine*, 34(11):92–100, November 1996.
- [3] Yehuda Afek, Yishay Mansour, and Zvi Ostfeld. Phantom: A Simple and Effective Flow Control Scheme. In *Proceedings of the ACM SIGCOMM*, pages 169–182, August 1996.
- [4] A. W. Barnhart. Example Switch Algorithm for TM Spec. ATM Forum 95-0195, February 1995.
- [5] R. Jain A. Charny, D. D. Clark. Congestion control with explicit rate indication. In *Proceedings of ICC 95*, June 1995.
- [6] Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Sonia Fahmy, and Ram Viswanathan. ER-ICA switch algorithm: A complete description. ATM Forum/96-1172, August 1996.
- [7] Robert J. Simcoe. Test configurations for fairness and other tests. ATM Forum/94-0557, July 1994.