****

ATM Forum Document Number: ATM Forum/98-0151

****

Title: A Definition of Generalized Fairness and its Support in Switch Algorithms.

****

Abstract:
In this contribution we give a general definition of fairness and show how this can achieve various fairness definitions, such as those mentioned in the ATM Forum TM 4.0 Specifications [1]. We discuss how a pricing policy can be mapped to general fairness. The general fairness can be achieved by calculating the $ExcessFairshare$ for each VC. We show how a switch algorithm can be modified to support the general fairness by using the $ExcessFairshare$. We use ERICA+ as an example switch algorithm and show how it can be modified to achieve the general fairness. Simulations results are presented to demonstrate that the modified switch algorithm achieves general fairness.

****

Source:
Bobby Vandalore, Sonia Fahmy, Raj Jain, Rohit Goyal, Mukul Goyal
The Ohio State University Department of Computer and Information Science
Columbus, OH 43210-1277

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

This work done in this contribution is sponsored by NASA Lewis Research Center.

****

Date: February 1998

****

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

****

****

1 Introduction

In ABR (available bit rate) service the users specify MCR (minimum cell rate) during connection set up. The ABR service gives guarantee that the ACR (allowed cell rate) is never less than MCR. When MCR is zero for all sources, the available bandwidth can be allocated equally among the competing sources. This allocation achieves max-min fairness. When MCRs are non-zero, other definitions of fairness allocate the excess bandwidth (which is available ABR capacity less the sum of MCRs) equally among sources, or proportional to MCRs, or proportional to a predetermined weight assigned for different sources.

In the real world, the users prefer to get a service which reflects the amount they are paying. The pricing policy requirements can be realized by mapping appropriately the weights associated with the sources.

We show how a switch schemes can support non-zero MCRs and achieve the generalized fairness. As an example, we show how the ERICA+ switch scheme can be modified to support generalized fairness.

The modified scheme is tested using simulations on various configurations. The simulations test the performance of the modified algorithm, using different weights using a simple configurations, with transient sources, a link bottleneck configuration, and a source bottlenecked configuration. The simulations show that the scheme realizes various fairness definitions in ATM TM 4.0 specifications, which are special cases of the generalized fairness.

Section 2 discusses the general fairness definition and shows how the various other definitions of fairness can be realized using this general definition. Section 4 shows how a switch scheme can achieve general fairness. Section 5 shows as an example, how ERICA+ is modified to support general fairness. Section 6 explains the simulation configurations and the parameters values used, and section 7 gives the simulation results.

2 General Fairness: Definition

Define the following parameters:

$A =$ Total available bandwidth for all ABR connections on a given link.

$U =$ Sum of bandwidth of underloaded connections which are bottlenecked elsewhere.

$B =$ A - U, excess bandwidth, to be shared by connections bottlenecked on this link.

$N_a =$ Number of active connections

$N_u =$ Number of active connections bottlenecked elsewhere.

$n = N_a - N_u$, number of active connections bottlenecked on this link.

$M =$ Sum of MCRs of active connections within bottlenecked on this link.

$B(i) =$ Generalized Fair Allocation for connection i.

$MCR(i) =$ MCR of connection i.

$w(i) =$ preassigned weight associated with the connection i.

The generalized fair allocation is defined as follows:

$$B(i) = MCR(i) + \frac{w(i)(B-M)}{\sum_{j=1}^{n} w(j)}$$

Note that this definition of fairness is different from the weighted allocation given as an example fairness criterion in ATM TM 4.0 specifications. In the above definition, only the excess bandwidth is allocated proportional to weights. This above definition ensures the allocation is at least MCR.

2.1 Mapping TM 4.0 Fairness to Generalized Fairness

Here we show how the different fairness criteria mentioned in ATM TM 4.0 specification, can be realized based on the above fairness.

1. **Max-Min:** In this case MCRs are zero and the bandwidth is shared equally.

   $$B(i) = B/n$$

   This is a special case of generalized fairness with $MCR(i) = 0$, and w(i) = c, where c is a constant.

2. **MCR plus equal share:** The excess bandwith is shared equally.

   $$B(i) = MCR(i) + (B-M)/n$$

   by assigning equal weights we achieve the above fairness.

3. **Proportional to MCR:** The allocation is proportional to its MCR.

   $$B(i) = \frac{B \times MCR(i)}{M} = \frac{(M+B-M)MCR(i)}{M} = MCR(i) + \frac{(B-M)MCR(i)}{M}$$

   By assigning w(i) = MCR(i) we can achieve the above fairness.

3 Relationship to Pricing/Charging Policies

Consider a very small interval T of time. The charge C that a customer pays for using a network during this interval is a function of the number of bits W that the network transported successfully:

$$C = f(W, R)$$

Where, $R = W/T$ is the average rate.

It is reasonable to assume that $f()$ is a non-decreasing function of W. That is, those sending more bits do not pay less. The function $f()$ should also be a non-increasing function of time T or equivalently a non-decreasing function of rate R.

For economy of scale, it is important that the cost per bit does not increase as the number of bits goes up. That is, C/W is a non-decreasing function of W.

Mathematically, we have three requirements:

$$\partial C/\partial W \geq 0$$

$$\partial C/\partial R \geq 0$$

$$\partial (C/W)/dW \leq 0$$

One simple function that satisfies all these requirements is:

$$C = c + wW + rR$$

Here, c is the fixed cost per connection; w is the cost per bit; and r is the cost per Mbps. In general, c, w, and r can take any non-negative value.

In the presence of MCR, the above discussion can be generalized to:

$$C = f(W, R, L)$$

Where, L is the MCR. All arguments given above for R apply to L also except that the customers requesting larger L possibly pay more. One possible function is:

$$C = c + wW + rR + mL$$

where, m is dollars per Mbps of MCR. In effect, the customer pays $r + m$ dollars per Mbps upto L and then pays only r dollars per Mbps for all the extra bandwidth he/she gets over and above L.

Consider two users with MCRs L_1 and L_2. Suppose their allocated rates are R_1 and R_2 and, thus, they transmit W_1 and W_2 bits, respectively. Their costs are:

$$C_1 = c + wW_1 + rR_1 + mL_1$$

$$C_2 = c + wW_2 + rR_2 + mL_2$$

Cost per bit (C/W) should be a decreasing function of bits W. Thus, if $W_1 \geq W_2$:

$$C_1/W_1 \leq C_2/W_2$$

$$c/W_1 + w + rR_1/W_1 + mL_1/W_1 \leq c/W_2 + w + rR_2/W_2 + mL_2/W_2$$

Since $R_i = W_i/T$, we have:

$$c/(R_1T) + w + r/T + mL_1/(R_1T) \leq c/(R_2T) + w + r/T + mL_2/(R_2T)$$

$$c/R_1 + mL_1/R_1 \leq c/R_2 + mL_2/R_2$$

$$(c + mL_1)/(c + mL_2) \leq R_1/R_2$$

$$(a + L_1)/(a + L_2) \leq R_1/R_2$$

Where a $(=c/m)$ is the ratio of the fixed cost and cost per unit of MCR.

Note that the allocated rates should either be proportional to a+MCR or be a non-decreasing function of MCR. This is the policy we have chosen for this contribution.

4 Achieving General Fairness in Switch Algorithms

A typical ABR switch scheme calculates the excess bandwidth capacity available for best effort ABR after reserving bandwidth for providing MCR guarantee and higher priority classes such as VBR and CBR. The switch fairly divides the excess bandwidth among the connections bottlenecked at that link. Therefore, the ACR can be represented by the following equation.

$$ACR(i) = MCR(i) + \mathrm{E}xcessFairshare(i)$$

$ExcessFairshare$ is the amount of bandwidth allocated over the MCR in a fair manner.

In the case of general fairness, the $ExcessFairshare$ term is given by:

$$ExcessFairshare(i) = \frac{w(i)(B - M)}{\sum_{j=1}^{n} w(j)}$$

If the network is near steady state, then the above allocation enables the sources to attain the generalized fairness. To achieve the generalized fairness, the ATM TM 4.0 specification mentions that the value of $(ACR - MCR)$ can be used. We have to ensure the $(ACR - MCR)$ converges to the $ExcessFairshare$. We use the notion of *activity level* to achieve the above [4]. A connection's *activity level* $(AL(i))$ is defined as follows.

$$AL(i) = minimum(1, \frac{SourceRate(i) - MCR(i)}{ExcessFairshare(i)})$$

$SourceRate(i)$ is the rate at which the source is currently transmitting data. Note that, $SourceRate(i)$ is the $ACR(i)$ given as the feedback rate earlier by the switch. The activity level indicates how much of the $ExcessFairshare$ is actually being used by the connection. The activity level attains the value of one when the $ExcessFairshare$ is used by the link. The weights used in the generalized fairness, assume that the links use their $ExcessFairshare$, but this might not be case. By multiplying the weights by the activity level, and using these as the weights in calculating the $ExcessFairshare$ we can make sure that the rates converge to the generalized fairness allocation. Therefore, the $ExcessFairshare$ share term is

$$ExcessFairshare(i) = \frac{w(i)AL(i)(B - M)}{\sum_{j=1}^{n} w(j)AL(j)}$$

An switch algorithm can use the above $ExcessFairshare$ term to achieve general fairness. In the next section we show how the ERICA+ switching algorithm is be modified to achieve general fairness.

5 Example Modifications to A Switch Algorithm

The ERICA+ algorithm operates at each output port of a switch. The switch periodically monitors the load on each link and determines a load factor (z), the available ABR capacity, and number of currently active sources or VCs (N). The measurement period is the "Averaging Interval". These measurements are used to calculate the feedback rate which is indicated in the BRM (backward RM) cells. The measurements are done in the forward direction and the feedback is given int the backward direction. The complete description of the ERICA+ algorithm can be obtained from [2].

The ERICA+ algorithm uses the term $FairShare$ which is the bottleneck link capacity divided by the active number of VCs. It also uses $MaxAllocPrevious$ term, which is the maximum allocation in the previous "Averaging Interval". This term is used to achieve Max-min fairness. We modify the algorithm by replacing the $FairShare$ term by $fairshare(i)$ and adding the MCR(i). The keys steps in ERICA+ which are modified to achieve the general fairness shown below:

End of Averaging Interval:

$$\begin{aligned}
\text{Total ABR Capacity} &\leftarrow \text{Link Capacity} - \text{VBR Capacity} \\
&\quad - \sum_{i=0}^{n} \min(SourceRate(i), MCR(i)) && (1) \\
\text{Target ABR Capacity} &\leftarrow Fraction \times \text{Total ABR Capacity} && (2) \\
\text{Input Rate} &\leftarrow \text{ABR Input Rate} - \sum_{i=0}^{n} min(SourceRate(i), MCR(i)) && (3) \\
z &\leftarrow \frac{\text{Input Rate}}{\text{Target ABR Capacity}} && (4)
\end{aligned}$$

$$ExcessFairshare(i) \leftarrow \frac{(\text{Target ABR Capacity})w(i)AL(i)}{\sum_{j=1}^{n} w(j)AL(j)} \tag{5}$$

$$\tag{6}$$

The $Fraction$ term is dependent on the queue length. Its value is one for small queue lengths and drops sharply as queue length increases. When the $Fraction$ is less than one, $(1 - Fraction) \times TotalABRCapacity$ is used to drain the queues. ERICA+ uses an hyperbolic function calculating value of the $Fraction$.

When a BRM is received:

$$\text{VCShare} \leftarrow \frac{SourceRate(i) - MCR(i)}{z} \tag{7}$$

$$\text{ER} \leftarrow MCR(i) + \text{Max (ExcessFairshare(i), VCShare)} \tag{8}$$

$$\text{ER\_in\_RM\_Cell} \leftarrow \text{Min(ER\_in\_RM\_Cell,ER,Target ABR Capacity)} \tag{9}$$

The $VCShare$ is used to achieve an unit overload. When the network reaches steady state the $VCShare$ term converges to $ExcessFairshare(i)$ term achieving generalized fairness criterion.

6 Simulation Configurations

We use different configurations to test the performance of the modified algorithm. We assume that the sources are greedy, i.e., they have infinite amount of data to send, and always send data at ACR. In all configurations the data traffic is only one way, from source to destination. All the link bandwidths are 149.76 (155.52 less the SONET overhead), expect in the GFC-2 configuration.

6.1 Three Sources

This is a simple configuration in which three sources send to three destinations over a two switchs and a bottleneck link. See figure 1. Only sources send data. This configuration is used to demonstrate that the modified switchs algorithm can achieve the general fairness for the various set of weight assignments.

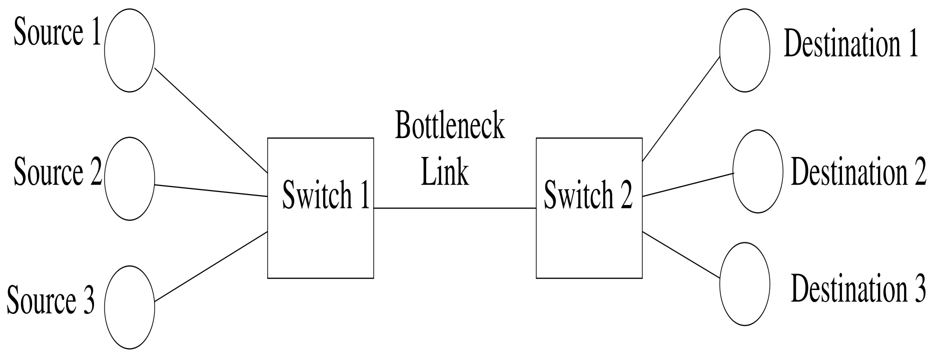


Figure 1: N Sources - N Destinations Configuration

6.2 Source Bottleneck

In this configuration, the source S1, is bottlenecked to rate (10 Mbps), which below its fairshare (50 Mbps). This configuration tests whether the fairness criterion can be achieved in the presence of source bottleneck.

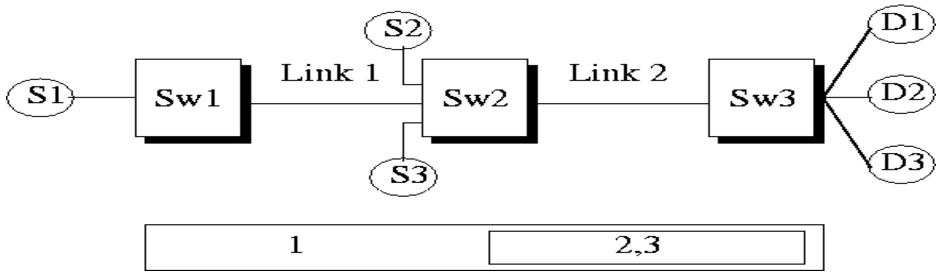


Figure 2: 3 Sources - Bottleneck Configuration

6.3 Generic Fairness Configuration - 2 (GFC-2)

This configuration is a combination of upstream and parking lot configuration (See Figure 3). In the configuration all the links are bottlenecked links. This configuration is explained in [3].

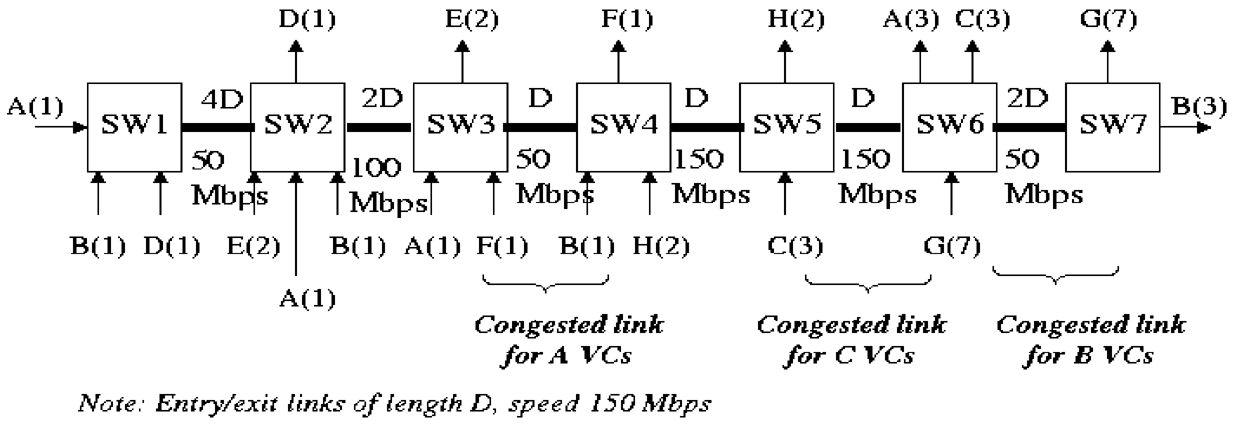


Figure 3: Generic Fairness Configuration - 2

6.4 Simulation Parameters

The parameters values used in the different configurations is given in Table 1. The "Averaging Interval" is the period for which the switch monitors various parameters. Feedback is given based on these monitored values. The ERICA+ algorithm uses dynamic queue control to vary the available ABR capacity dependent on queue size. At steady state the queue length remains constant. The

Table 1: Simulation Parameter Values

Configuration Name	Link Distance	Averaging interval	Target Delay
Three Sources	1000 Km	5 ms	1.5 ms
Source Bottleneck	1000 Km	5 ms	1.5 ms
GFC-2	1000 Km	15 ms	1.5 ms

Table 2: Three sources configuration simulation results

Case Number	Src Num	mcr	a	weight function	Expected fair share	Actual share
1	1	0	∞	1	49.92	49.92
	2	0	∞	1	49.92	49.92
	3	0	∞	1	49.92	49.92
2	1	10	∞	1	29.92	29.92
	2	30	∞	1	49.92	49.92
	3	50	∞	1	69.92	69.92
3	1	10	5	15	18.53	16.64
	2	30	5	35	49.92	49.92
	3	50	5	55	81.30	81.30

"Target Delay" parameter specifies the desired delay due to this constant queue length at steady state.

7 Simulation Results

In this section we give the simulation results for the different configurations.

7.1 Three Sources

Simulations using a number of weight functions were done using the simple three sources configuration to demonstrate that general fairness is achieved in all these cases. The ICRs of the sources were set to the (50,40,55) in all the simulations. The results of these cases are given in Table 2. The Figure 2 shows the ACRs of the three sources and the queue length to bottleneck link at switch-1.

The following can be observed from the Table 2

- Case 1: a = ∞, MCRs = 0. All weights are equal so the allocation (149.76/3) = 49.92 for each connection. This is allocation is the same as max-min fair allocation.

Table 3: Three sources transient configuration simulation results

Case Number	Src Num	mcr	a	weight function	Expected fairshare (non-trans.)	Actual (non-trans) share	Expected fairshare (trans.)	Actual (trans.) share
1	1	0	∞	1	74.88	74.83	49.92	49.92
	2	0	∞	1	-	-	49.92	49.92
	3	0	∞	1	74.88	74.83	49.92	49.92
2	1	10	∞	1	54.88	54.88	29.92	29.83
	2	30	∞	1	-	-	49.92	49.92
	3	50	∞	1	94.88	95.81	69.92	70.93
3	1	10	5	15	29.92	29.23	18.53	18.53
	2	30	5	35	-	-	49.92	49.92
	3	50	5	55	119.84	120.71	81.30	81.94

- Case 2: a = ∞, MCRs $\neq$ 0. The left over capacity 149.76 - (10 + 30 + 50) = 59.76 is divided equally among the three sources. So the allocation is (10 + 19.92, 30 + 19.92, 50 + 19.92) = (29.92,39.92,69.92).

- Case 3: a = 5, MCRs $\neq$ 1. Hence, the weight function is 5 + MCR. The left over capacity 59.76 Mbps, is divided proportional to (15,35,55). Hence the allocation is (10 + 15/105 $\times$ 59.76, 30 + 35/105 $\times$ 59.76, 50 + 55/105 $\times$ 59.76) = (16.64, 49.92, 83.2).

7.2 Three Sources: Transient

In these simulations the same simple three source configuration is used. The source 1 and source 3 transmit data throughout the simulation period. The source S2 is a transient source, which starts transmitting at 400 ms and stops at 800 ms. The total simulation time is 1200 ms. The same parameters values from the case's 1, 2 and 3 of the previous sections were used in these simulations. The results of these simulations are given in Table 3. The (non-trans.) columns give the allocation when source 2 is not present, i.e., between 0ms to 400ms and between 800ms to 1200 ms. The (trans.) columns give allocation when the transient source 2 is present, i.e., between 400 ms to 800 ms.

The graphs for these three simulations are shown in figure 5. The graphs show the ACRs of the three sources and the bottleneck link utilization. It can be seen both from the Table 3 and the graphs that the switch algorithm does converge to the general fairness allocation even in the presence of transient sources. Note the link utilization is high throughout the simulation. The width of dip in the utilization graph, when the transient sources goes away, indicates the reponsiveness of the algorithm. This shows that the algorithm is tolerant of transient sources and responds quickly to changing demands.

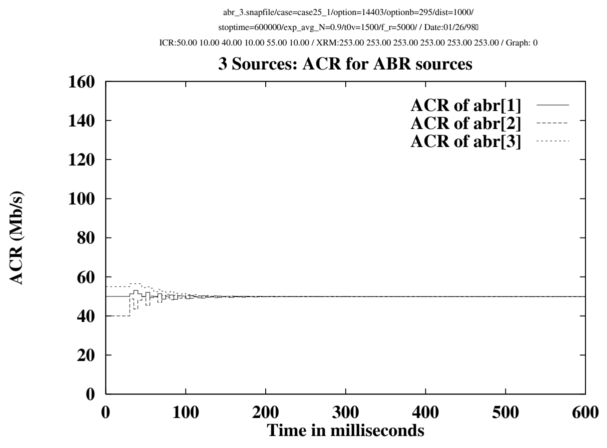
Table 4: Three sources bottleneck configuration simulation results

Case Number	Src Num	mcr	a	weight function	Expected fair share	Using CCR in RM cell	Using Measured CCR
1	1	0	∞	1	49.92	49.85	49.92
	2	0	∞	1	49.92	49.92	49.92
	3	0	∞	1	49.92	49.92	49.92
2	1	10	∞	1	29.92	-	29.62
	2	30	∞	1	49.92	-	49.60
	3	50	∞	1	69.92	-	71.03
3	1	10	5	15	18.53	-	18.42
	2	30	5	35	49.92	-	49.92
	3	50	5	35	81.30	-	81.93

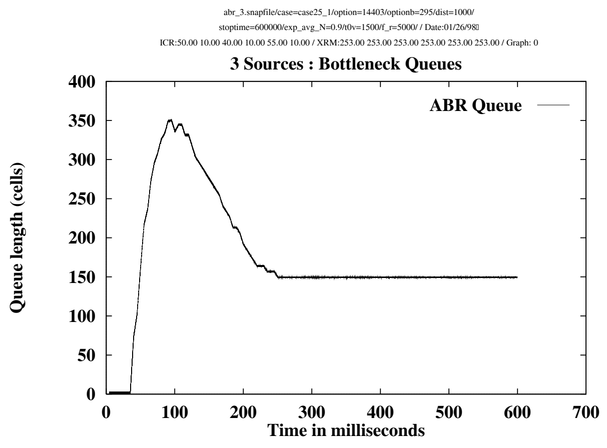
7.3 Source Bottleneck

The case 1, 2 and 3 of section 7.1 were simulated using the three sources bottleneck configuration. In these simulations the source S1 is bottlenecked at 10 Mbps, i.e., it always transmits data at rate of atmost 10 Mbps, irrespective of its ACR (and ICR). The initial ICRs were set to 50, 30, 110. The load on the bottleneck link is near unity. If the switch algorithm uses the CCR value indicated in the RM cell as the source rate the switch cannot estimate the correct value of source rate of the bottleneck source. But if the switch uses measured source rate then it can correctly estimate the bottlenecked source's rate. Table 4 shows the results both when the switch uses the CCR field and when it measure's the source rate. The correct fairness is achieved when the measured source rates are used.

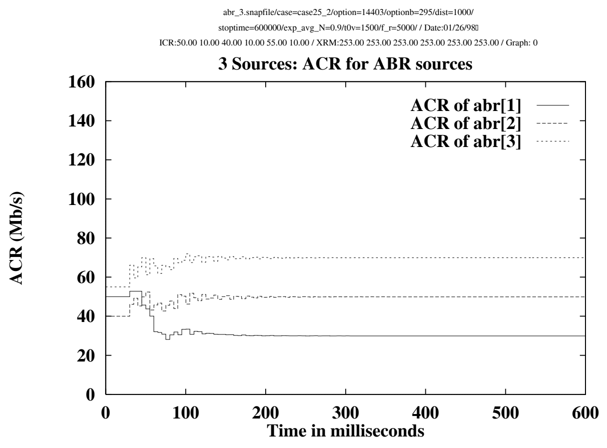
The graphs for the simulations are given in Figure 6. The switch algorithm uses queue control, to dynamically use part of available ABR capacity to drain the queues. When the queue is large the available ABR capacity is only a fraction of actual capacity. So, the algorithm takes sometime before converging to the correct fairness values. When the CCR value from the RM cells is used, the algorithm does not converge in case 2 and case 3. In case 1, it converged since the bottleneck source's rate (CCR) had the correct value of 50 which is the same allocation it would get in the fair allocation.



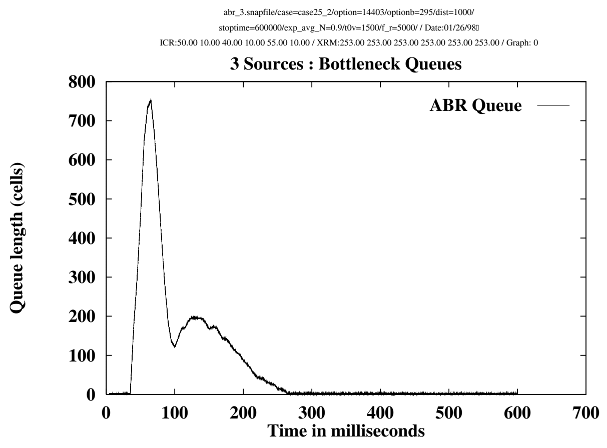
(a)



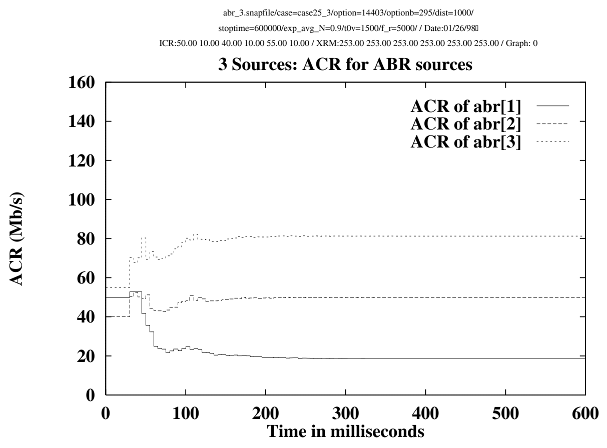
(b)



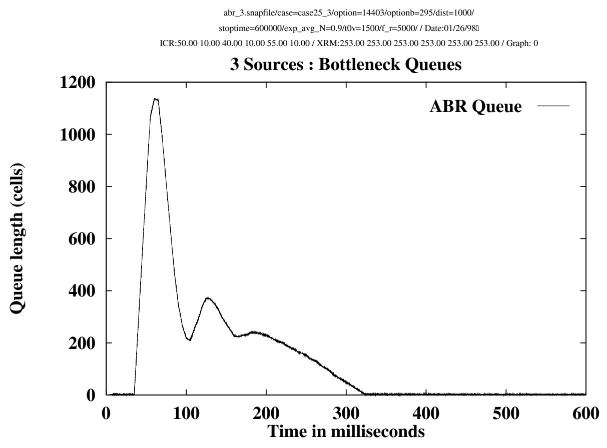
(c)



(d)

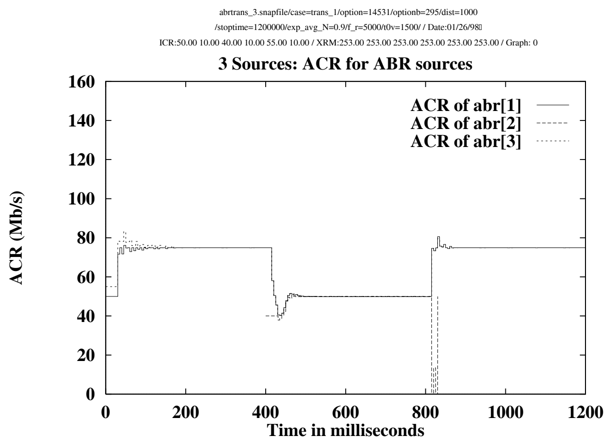


(e)

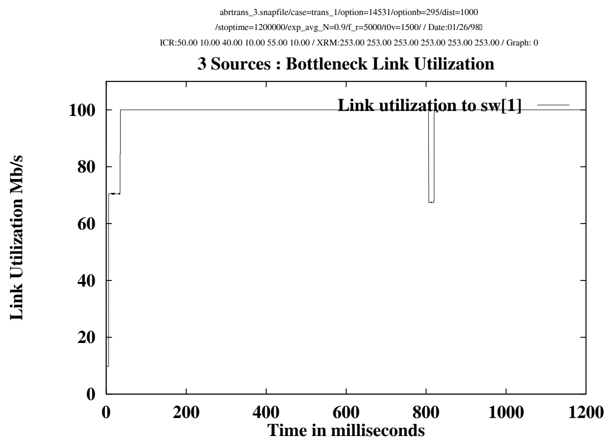


(f)

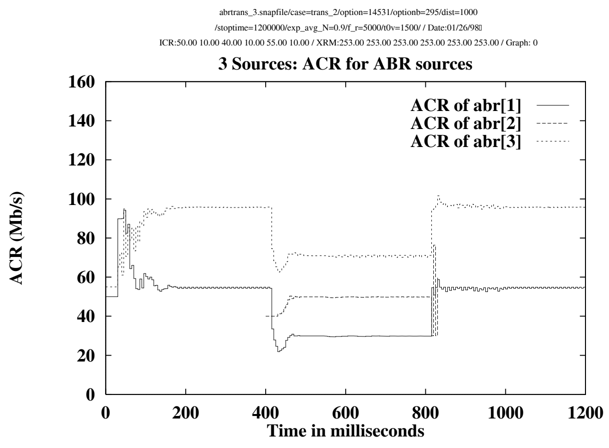
Figure 4: Three Sources: ACR graphs and Queue length of bottleneck link



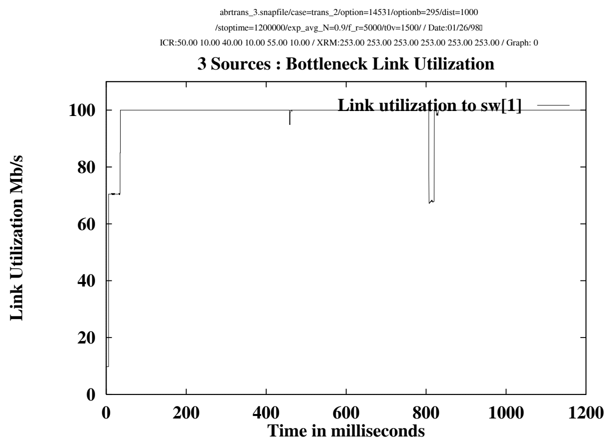
(a)



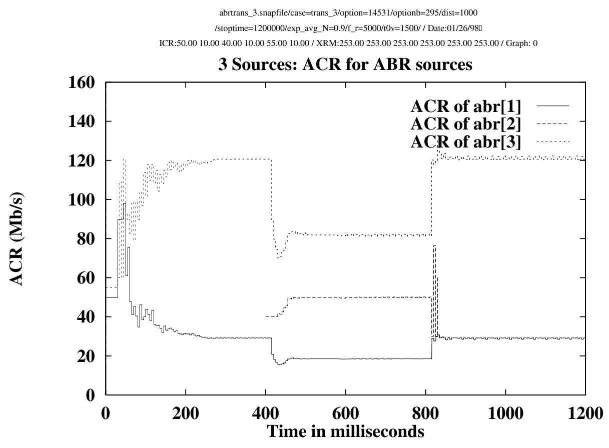
(b)



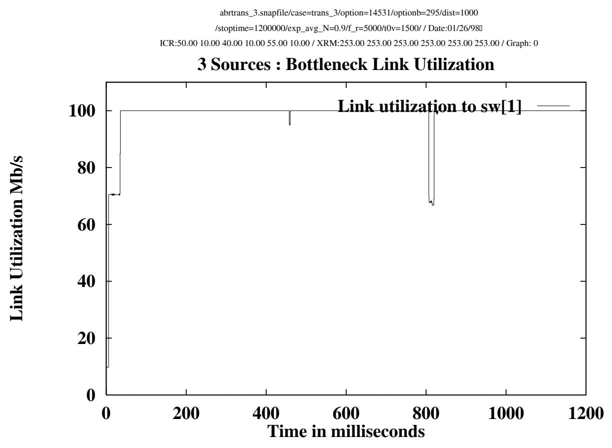
(c)



(d)

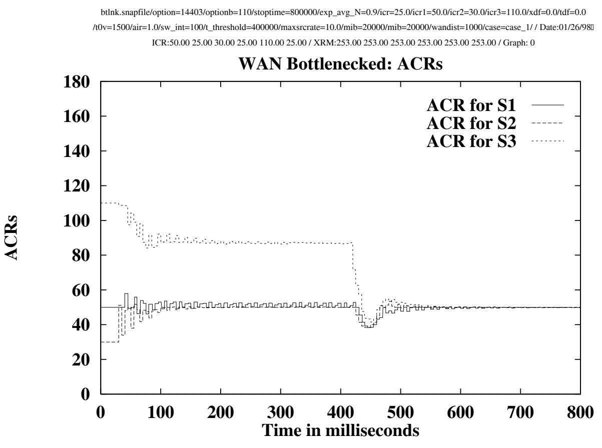


(e)

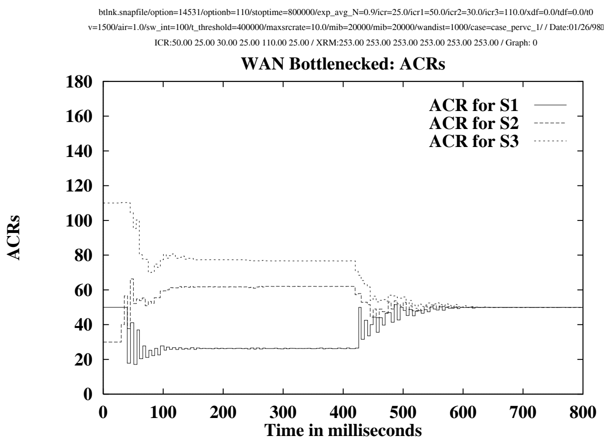


(f)

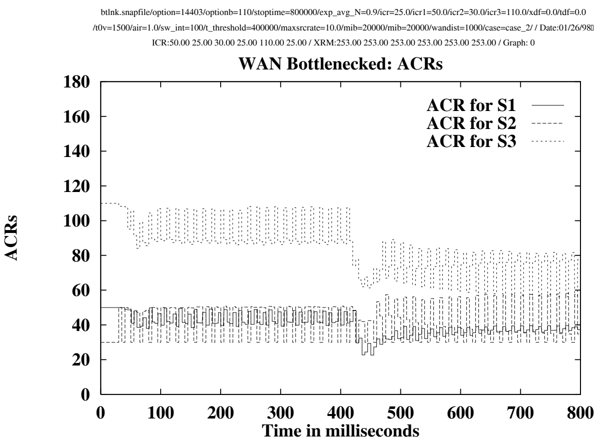
Figure 5: Three Sources (Transient) : ACR and Utilization graphs



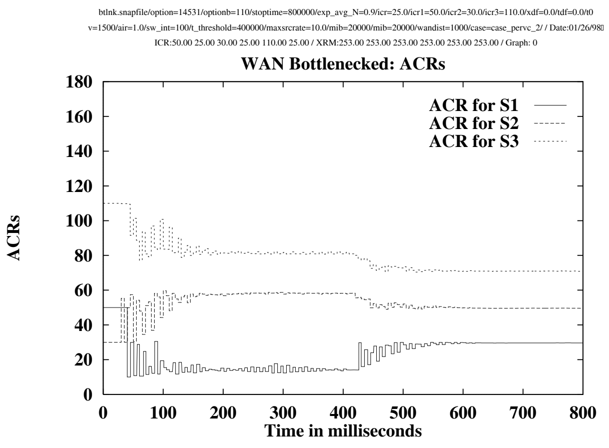
(a)



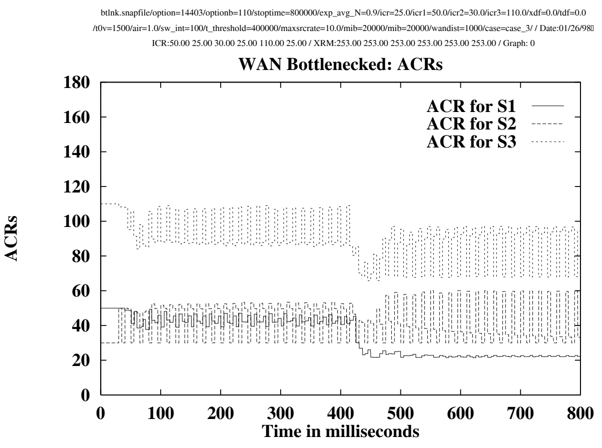
(b)



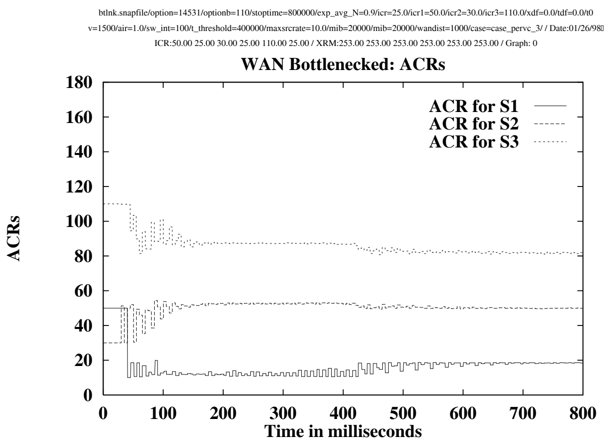
(c)



(d)

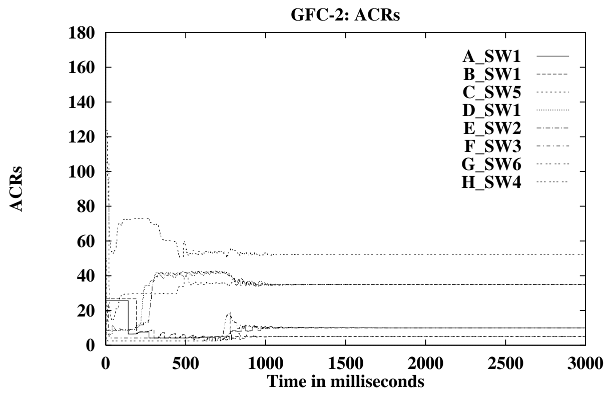


(e)

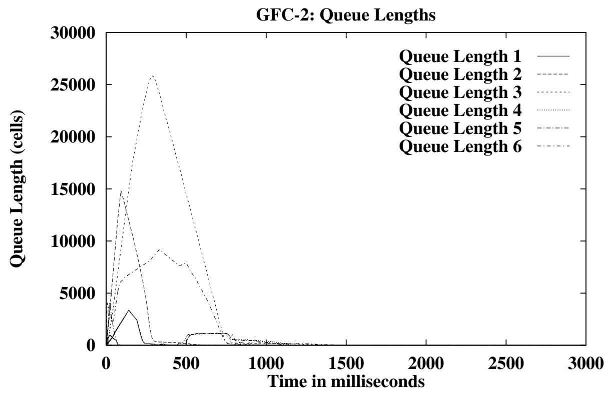


(f)

Figure 6: Three Sources Bottleneck: ACR graphs (with and without measuring Source Rate)



(a)



(b)

Figure 7: GFC-2 configuration: ACRs of A through H, VCs and Queue lengths at bottlenecks links

Table 5: GFC-2 configuration: simulation results

Case Number	VC type	Expected allocation	Actual Allocation
1	A	10	9.85
	B	5	4.97
(a = ∞)	C	35	35.56
	D	35	35.71
(all MCRs	E	35	35.34
are zero)	F	10	10.75
(same as	G	5	5.00
max-min)	H	52.5	51.95

7.4 Link Bottleneck: GFC-2

In this configuration each link is a bottleneck link. The Figure 7 (a) shows the ACR graphs for each type of VCs. Figure 7 (b) shows the queue length of all the bottleneck links (links between the switches). From the Figure and Table 5 it can be seen that the VCs converge to their expected fairshare. This shows that the algorithm works in the presence of link bottlenecks.

8 Conclusion

In this contribution, we have given a general definition of fairness, which inherently provides MCR guarantee and divides the excess bandwidth proportional to predetermined weights. Different fairness criterion such as max-min fairness, MCR plus equal share, proportional MCR can be realized as special cases of this general fairness. We showed how to realize a typical pricing policy by appropriate weight function. The general fairness can be achieved by using the $ExcessFairshare$

term in the switch algorithms. The weights are multiplied by the activity level when calculating the $ExcessFairshare$ to reflect the actual usage of the source.

We have shown how ERICA+ switch algorithm can be modified achieve this general fairness. The modified algorithm has been tested under different configuration using persisent sources. The simulations results show that the modified algorithm achieves the general fairness in all configurations. In source bottlenecked configuration the value of the CCR from the RM cell maybe incorrect. Hence, it is necessary to used the measured source rate in the presence of source bottlenecks.

References

[1] Shirish S. Sathaye. ATM Forum Traffic Management Specification Version 4.0 April 1996

[2] Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Sonia Fahmy, and Ram Viswanathan. ERICA switch algorithm: A complete description. ATM Forum/96-1172, August 1996.

[3] Robert J. Simcoe. Test configurations for fairness and other tests. ATM Forum/94-0557, July 1994.

[4] Sonia Fahmy, Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, and Bobby Vandalore. Determining the number of active ABR sources in switch algorithms. ATM Forum/98-0154, February 1998.