

A Flexible Multipoint-to-Point Traffic Control Algorithm for ABR Services in ATM Networks

Uyen T. Nguyen and Irene Katzela
Department of Computer Science, University of Toronto
Toronto, Ontario M5S 3G4 Canada

Abstract

Traffic control for point-to-multipoint ABR services in ATM networks has been studied extensively, but this is not the case for multipoint-to-point (mp-p) traffic control. No consensus has been reached concerning the issue of fairness of bandwidth allocation among multipoint connections, or between point-to-point and multipoint connections. Existing mp-p traffic control schemes for ABR services either support only a limited set of fairness definitions, or are not effective. In this paper, we define a new type of fairness, and propose a flexible multipoint-to-point traffic control scheme for ABR services. The proposed scheme supports cell merging at both virtual path (VP) and virtual connection (VC) levels, and implements all existing types of fairness currently defined in the literature for mp-p communications. This flexibility allows the network to meet various bandwidth requirements of different application streams. It also permits switch vendors to quickly configure their switches to meet clients' requirements, despite the lack of consensus on the issue of fairness definition. While being flexible, the new algorithm is also fair, fast and robust. Simulation results show that the proposed scheme is fair, and performs as well as, or better than existing mp-p traffic control schemes. It is also easy to incorporate the new algorithm into a multipoint-to-multipoint traffic control scheme such as the framework proposed by Ren, Siu and Suzuki.

1 Introduction

Multipoint communications support the exchange of information among a set of participants. Multipoint communication is by definition more greedy in bandwidth than point-to-point (p-p) communication since a multipoint connection involves multiple senders and/or multiple receivers. Therefore, the research in the area of traffic control for multipoint communications is very important, useful, and more challenging compared to the p-p case.

Traffic control for point-to-multipoint ABR services in ATM networks has been studied extensively [15, 16, 17, 18, 19]. Multipoint-to-point traffic control for ABR, on the other hand, has not been investigated in depth. No consensus has been reached concerning the issue of fairness of bandwidth allocation among multipoint connections, or between point-to-point and multipoint connections. Existing mp-p traffic control schemes for ABR services [9, 10, 11, 12] either support only a limited set of fairness definitions, or are not effective.

In this paper, we introduce a new type of fairness, and propose a flexible mp-p traffic control scheme for

ABR services, the NK¹ algorithm. The proposed algorithm supports cell merging at both virtual path (VP) and virtual connection (VC) levels. More importantly, it implements all existing types of fairness currently defined in the literature for mp-p communications [6, 11]. This flexibility allows ATM networks to meet various bandwidth requirements of a wide range of ABR applications. Using the NK algorithm, it is easy to select a desired fairness definition simply by setting a few parameters. This advantage enables switch vendors to quickly configure their switches to meet clients' requirements, despite the lack of consensus on the issue of fairness definition. The proposed algorithm is not only flexible, but also fair, fast and robust. Simulation results show that the NK algorithm is fair, and performs as well as, or better than existing mp-p traffic control schemes [10]. Furthermore, the new algorithm can be applied easily to a multipoint-to-multipoint (mp-mp) traffic control scheme such as the framework proposed by Ren, Siu and Suzuki [9]. The paper is organized as follows. In section 2, we discuss the issues involved in the design of a mp-p

¹ Named using the first letters of the authors' last names. We will use this naming convention throughout the paper to denote different mp-p traffic control schemes.

traffic control scheme. We also introduce a new type of fairness in this section. Section 3 provides a literature survey on mp-p ABR traffic control. The proposed mp-p algorithm is described in section 4. Simulation results are presented in section 5. Section 6 concludes the paper

2 Design Issues of Multipoint-to-Point Traffic Control

The design of a mp-p traffic control scheme must address the following four issues: the cell interleaving problem of AAL5 (ATM Adaptation Layer 5); the fairness definition; the explicit rate calculation algorithm at switches; and the processing of resource management (RM) cells.

2.1 Cell interleaving problem of AAL5

AAL5 does not have any provision within its cell format for the interleaving of cells from different AAL5 packets on a single connection. In a mp-p group, data cells from various sources will eventually be merged into one single channel, and thus require a mechanism to preserve the integrity of packets from one another. Among the proposed solutions to the cell interleaving problem of AAL5 [6], VP merge and VC merge are the most popular techniques. With VP merge, the virtual path identification (VPI) field of ATM cells is used to identify the virtual connections (VCs), while the virtual connection identification (VCI) field is used to distinguish among the sources of the same multipoint-to-point VC. With VC merge, both VPI and VCI fields of ATM cells are used to identify the VCs. To prevent the interleaving of cells of packets originating from different sources on the same mp-p VC, cells of other packets are buffered at the switch until all cells of the current packet go through.

2.2 Fairness Definitions

For unicast² connections, the most commonly used definition is *max-min fairness* [5]. The definition of fairness becomes more complicated when mp-p connections are involved since a mp-p VC is shared by several sources. Fahmy, *et al.* have introduced the following four types of fairness [6]:

- **Source-based:** divides bandwidth fairly among active sources, ignoring whether a source belongs to a p-p or mp-p connection.

- **VC/source-based:** first allocates the available bandwidth fairly among the VCs, and then divides the bandwidth for each VC fairly among the active sources in the VC.
- **Flow-based:** divides bandwidth fairly among active flows, where a flow is made up of cells of a VC coming on an input link.
- **VC/flow-based:** first allocates the available bandwidth fairly among the VCs, and then divides the bandwidth for each VC fairly among the active flows in the VC.

Wang and Schwartz have suggested the concept of “essential fairness” to define fairness between multicast (point-to-multipoint) and unicast TCP connections [7]. Moh and Chen extend the essential fairness definition to mp-p communications of ABR traffic in ATM networks [11]. This extended essential fairness requires every source to be assigned a weight. The weight of any unicast source is one. The weight of a source of a mp-p connection i is w_i , where $0 < w_i \leq 1$. Let b be the bandwidth of a unicast source. Then the bandwidth of a source of mp-p connection i is $w_i \times b$. We name the extended essential fairness *weighted-source-based* since each mp-p source is now given a weight. For two different mp-p connections i and j , we can have $w_i \neq w_j$. Note that when $w_i = 1$, *weighted-source-based* becomes *source-based* fairness.

When the number of sources sharing an mp-p VC is large, VC/source-based definition can be unfair to multipoint-to-point sources, and so can source-based definition to unicast sources. *Weighted-source-based* fairness provides a compromise between these two extreme definitions

We extend the concept of *weighted-source-based* fairness to flows. Similarly, each flow is assigned a weight. Unicast flows have weight equal to one. Each flow of a mp-p connection i has weight f_i , where $0 < f_i \leq 1$. If the bandwidth of a unicast flow is b , then the bandwidth of a flow of mp-p connection i is $f_i \times b$. This fairness definition is named *weighted-flow-based*. Two different mp-p connections i and j can have $f_i \neq f_j$. When $f_i = 1$, we have *flow-based* fairness. *Weighted-flow-based* fairness offers a trade-off between the two extreme cases of *flow-based* and *VC/flow-based* fairness.

2.3 Explicit Rate Calculation Algorithm

Given the max-min fairness definition, several explicit rate calculation algorithms have been proposed for unicast communications [4]. Typical rate calculation algorithms are the MIT scheme [1], EPRCA [2], ERICA [3], and FMMRA [4]. A unicast rate calculation algorithm can be modified and extended to support

² The terms “unicast” and “point-to-point” are used interchangeably in this paper

multipoint connections. In fact, the multipoint-to-point traffic control schemes presented in [10] and [11] are based on ERICA algorithm [3].

The choice of VC merge or VP merge also affects the design of a multipoint rate calculation algorithm. With VC merge implementation, cells of different sources in the same mp-p VC are not distinguishable. Therefore, with VC merge, the rate calculation algorithm must avoid distinguishing between rates of different sources. Such per-source accounting, however, is possible with VP merge because the VCI field is used to distinguish among cells of different sources in the same mp-p VC.

2.4 Processing of RM cells

There exist two methods of processing RM cells. In the ($FRM \rightarrow BRM$) method, an FRM cell arriving at the switch triggers it to return a BRM to the source [8]. In the ($BRM \rightarrow BRM$) method, when a BRM cell arrives at a switch from downstream, the switch sends a BRM to the source [9]. The latter method is more complex to implement with mp-p connections, but works effectively when extended to multipoint-to-multipoint traffic control [9].

3 Related Work

The first work on mp-p traffic control was published by Ren, Siu, and Suzuki [8]. The proposed scheme, RSS algorithm, assumes VC merge, and source-based fairness. No rate calculation algorithm was proposed. Instead, the authors claim that any unicast algorithm can be applied directly to their mp-p scheme without modifications, and if the unicast algorithm is fair then the resulting mp-p algorithm is also fair. In reality, there exist two problems with this mp-p traffic control scheme: Firstly, using VC merge, sources of the same mp-p group at a merging point are not distinguishable. If a unicast rate calculation algorithm uses per-source accounting, it cannot be applied directly to the proposed mp-p scheme. Secondly, simulation results suggested that the above mp-p algorithm is not fair, even if the original unicast algorithm is fair [11].

In [9], Ren, Siu, and Suzuki propose a framework for mp-mp traffic control. This mp-mp framework is the combination of a p-mp traffic control algorithm [15] and an mp-p algorithm. The mp-p algorithm is taken from the authors' previous work [8], and thus still suffers from the above drawbacks. However, a fair and correct mp-p algorithm can be applied to this framework to obtain an effective mp-mp traffic control scheme.

Fahmy, Jain, *et al.* introduced a mp-p traffic control scheme [10] based on the rate calculation algorithm ERICA [3]. Although this scheme is fair and fast, its application can be limited since it supports only one type of fairness: source-based fairness. We call this scheme FJ algorithm.

Another mp-p traffic control scheme based on ERICA algorithm was suggested by Moh and Chen [11]. We call this scheme MC algorithm. This algorithm implements only source-based and weighted-source-based fairness. The MC algorithm employs the basic ERICA algorithm, which does not use ER_{max} , the maximum explicit rate posted on the BRM cells processed during a measurement interval, for computing new explicit rates [3]. It has been shown that there are cases in which the basic ERICA algorithm does not converge to fair allocation [3, 4]. The same unfairness problem exists in the MC algorithm since it uses the basic ERICA algorithm. ER_{max} can be used to remedy the problem, but it must be scaled to the weight of an mp-p connection when the new explicit rate of the connection is calculated.

Another work related to multipoint flow control is presented in [12]. The author considers the multipoint-to-multipoint case, but does not address the mp-p case. A multipoint-to-multipoint connection with N members is implemented as N point-to-multipoint connections, each having one sender and $N-1$ receivers. This scheme is thus not scalable as N grows.

References [13, 14] are also related to multipoint communications, but they focus on multipoint architectures, not traffic control.

The proposed NK algorithm implements all existing fairness definitions suggested in the literature for multipoint-to-point communications [6, 11]. **Table 1** summarizes the types of fairness supported by the RSS, FJ, MC and NK algorithms respectively. The NK algorithm is not only flexible, but also fair, fast and robust. Its implementation is described in the next section.

Table 1 Types of fairness supported by different mp-p traffic control algorithms

Fairness Type	RSS	FJ	MC	NK
Source-based	x	x	x	x
VC/source-based				x
Weighted-source-based			x	x
Flow-based				x
VC/flow-based				x
Weighted-flow-based				x

4 The NK Multipoint-to-Point Traffic Control Algorithm

In this section, we first provide an overview of the NK algorithm, and then describe the algorithm in details.

4.1 Overview

The NK mp-p traffic control scheme supports both VC merge and VP merge. RM cells are processed using the ($BRM \rightarrow BRM$) method since it allows us to extend the algorithm to the multipoint-to-multipoint case later. The NK algorithm implements all existing fairness definitions suggested in the literature, namely source-based, VC/source-based, flow-based, and VC/flow-based fairness proposed in [6], weighted-source-based fairness defined in [11], and weighted-flow-based fairness introduced in section 2.2.

We have selected the unicast algorithm FMMRA [4] as the base for the new mp-p rate calculation algorithm. FMMRA has $O(1)$ computation complexity and $O(N)$ storage complexity, comparable with the best switch algorithms such as ERICA [3]. We have chosen FMMRA since it is fair, fast and robust. These advantages result from the following characteristics of the FMMRA algorithm:

- The fair share is used for computing new explicit rates. The fair share of a link can be defined roughly as the target ABR capacity divided by the number of active ABR sources traversing the link.
- The traffic load on the link is used to correct the computation of the fair share to be allocated to each active VC. Using the traffic load as a correction factor permits the explicit rate (ER) calculation to converge rapidly to fair share when the number of active ABR sources is not known, or when bursty sources are present.
- The switch keeps track of the maximum ER over all BRM cells, ER_{max} , and uses ER_{max} together with the traffic load in setting the explicit rate. The advantage of using ER_{max} with respect to using individual CCRs (as the MC algorithm does) is that in this way all connections have their rates allocated from a common reference. Once the connections are coupled to the same reference, by using the load, the ER is guaranteed to converge to the fair share.
- The queue length is used as an additional indicator of congestion, in order to improve congestion control, and maintain the buffer requirements at a desired level.
- The ER field in the RM cells is set in both forward and backward directions. This speeds up the convergence time since the ER set in FRM cells captures upstream bottlenecks. When the ER is

computed in the backward direction, a source will not be given more bandwidth than the amount set by its upstream switches, avoiding unnecessary rate oscillations.

- FMMRA is robust since its performance and accuracy are independent of the rate increase (RIF) and decrease (RDF) parameters [20]. Moreover, FMMRA does not use CCR values in RM cells, and thus is not sensitive to CCR errors.

We now describe the extensions to the FMMRA algorithm that are needed to accommodate mp-p connections. Let $W_{s,i}$ be the weight of an mp-p connection i at switch s . Depending on the desired type of fairness, $W_{s,i}$ is computed as follows:

- **VC/source-based** and **VC/flow-based**: $W_{s,i} = 1$.
- **Source-based**: $W_{s,i} = N_{s,i}$, where $N_{s,i}$ is the number of sources of connection i merging at switch s .
- **Weighted-source-based**: $W_{s,i} = N_{s,i} \times w_i$, where w_i is the weight of a source of connection i .
- **Flow-based**: $W_{s,i} = F_{s,i}$, where $F_{s,i}$ is the number of flows of connection i merging at switch s .
- **Weighted-flow-based**: $W_{s,i} = F_{s,i} \times f_i$, where f_i is the weight of a flow of connection i .

The weight of any p-p connection is one. The FMMRA algorithm is modified so that, at switch s , mp-p connection i is allocated $W_{s,i}$ times bandwidth of a p-p connection, if i is not bottlenecked elsewhere. In other words, each connection is offered an amount of bandwidth proportional to its weight. If connection i is bottlenecked elsewhere, the offered amount is reduced to the bottlenecked bandwidth. Note that appropriate switch variables (e.g., the fair share, ER_{max}) are scaled to $W_{s,i}$ when the new explicit rate of connection i is calculated. We call this modified version of the FMMRA algorithm “FMMRA with weighted connections” or FMMRAW for short.

4.2 Detailed Description of The NK Algorithm

Point-to-point connections do not require any special processing or variables, except that FMMRAW is used to calculate new explicit rates instead of the original FMMRA algorithm. This is because FMMRAW takes into account the weights of mp-p connections when computing new explicit rates.

Multipoint-to-point connections, however, need additional variables at switches. Every mp-p connection at a switch requires three additional arrays; each array has B elements, where $B \geq 1$ is the number of branches of the mp-p connection merging at this switch. The three arrays are:

- **receivedFRM**: $receivedFRM[k]$ is set to true if an FRM cell has been received from branch k

since the last BRM cell was returned along branch k .

- *numberSources*³: *numberSources[k]* indicates the number of sources in this mp-p connection whose data is carried by branch k .
- *savedER*: *savedER[k]* stores the explicit rate provided by field ER of FRM cells. It helps to capture upstream bottleneck of branch k , if any, allowing faster convergence to the optimal operating rate.

We consider the case of **source-based**, **VC/source-based** and **weighted-source-based** fairness first, and discuss the implementation of the other fairness definitions based on this case.

When an FRM cell arrives at the switch from branch k of an mp-p connection i , the switch sets variable *receivedFRM[k]* of connection i to true, and stores the value of field ER of the FRM cell into *savedER[k]*. The switch then uses FMMRAW algorithm and array *savedER* to compute the new explicit rate of the connection. Field ER of the FRM cell is set to the new explicit rate, and the FRM cell is forwarded downstream. The pseudo-code is as follows:

```

receivedFRM[k] = true;
savedER[k] = FRM.ER;
Apply FMMRAW to compute offeredRate;
sumER = 0;
for every branch  $j$  of connection  $i$ 
    sumER = sumER + savedER[j];
newER = min (sumER, offeredRate);
FRM.ER = newER;
Forward the FRM cell downstream;

```

Note that *sumER* is the total bandwidth from the merging branches of connection i , and set by upstream switches. The computed *offeredRate* is the fair rate that the switch is willing to offer to connection i . This amount is reduced to *sumER* if i has any upstream bottlenecks.

When a BRM cell arrives at the switch from an mp-p connection i , the switch computes the new explicit rate using FMMRAW algorithm, and field ER of the BRM cell. The allocated bandwidth is then divided among the branches of the connection merging at this point.

³ One way to maintain variables *numberSources* is as follows. When a source joins the mp-p group, it sends a “join” signal to the destination. Each switch on the source-to-destination path increments variable *numberSources* of the corresponding incoming branch. A similar procedure is performed when a source leaves the mp-p group; that is, variable *numberSources* is decremented upon receiving a “leave” signal. Another method for maintaining variables *numberSources* is suggested in [11].

Each branch is offered an amount of bandwidth proportional to the number of sources whose data it carries. A BRM cell is returned to each branch k whose *receivedFRM[k]* is true. Following is the pseudo-code:

```

Apply FMMRAW to compute offeredRate;
newER = min (BRM.ER, offeredRate);
N = 0;
for every branch  $k$  of connection  $i$ 
    N = N + numberSources[k];
for every branch  $k$  of connection  $i$ 
    if receivedFRM[k] = true then
        Generate a BRM cell;
        BRM.ER = min (savedER[k],
            newER * numberSources[k] / N);
        Send BRM cell upstream along branch  $k$ ;
        receivedFRM[k] = false;
    endif;

```

Note that if branch k has an upstream bottleneck, which is captured in *savedER[k]*, then its equal share is reduced to *savedER[k]*.

To support **flow-based**, **VC/flow-based** and **weighted-flow-based** fairness, we simply set array variables *numberSources* of all branches of mp-p connections to one. The same algorithm as described above is applied. After a BRM cell is received from an mp-p connection i , and the new explicit rate is computed, the bandwidth allocated to i is effectively divided equally among its branches (or flows), resulting in flow-based fairness within the mp-p connection. Note that in the NK algorithm it is easy to switch from one fairness definition to another simply by using the appropriate formula to compute weights of connections as shown in section 4.1, and setting array variables *numberSources* of mp-p connections.

5 Simulation Results

Fig. 1 shows the configuration used for simulation. In this configuration, multipoint-to-point sources S_1 , S_2 and S_3 send data to destination D_{123} , while unicast source S_u sends to destination D_u . In the experiments presented in this paper, we use the following setting. Each link has an ABR capacity of 150 Mbps. The distance between switches is 1000 km. Terminal-switch distance is 1km. VC merge is simulated, and the packet size is 500 bytes. The sources are persistent, and start sending data at the same time. The duration of each experiment is 1000 msec. Given the above configuration and link bandwidths, **Table 2** lists the sources' rates for each type of fairness. We conducted three experiments to evaluate the effectiveness, performance, and robustness of the NK algorithm.

Fig. 1 Simulation configuration

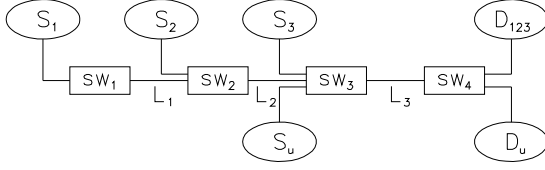


Table 2 Fair rates of the sources in Mbps

Fairness Type	S_1	S_2	S_3	S_u
Source	37.50	37.50	37.50	37.50
VC/source	25.00	25.00	25.00	75.00
Weighted-source	34.62	34.62	34.62	46.15
Flow	25.00	25.00	50.00	50.00
VC/flow	18.75	18.75	37.50	75.00
Weighted-flow	22.50	22.50	45.00	60.00

5.1 Effectiveness and Performance of The NK Algorithm

In the first experiment, we evaluate the effectiveness and performance of the NK algorithm when all the sources have a low initial cell rate (ICR) of 25 Mbps. **Fig. 2** shows the allowed cell rates (ACRs) of the sources for each type of fairness. We can see that the sources' rates converge to fair allocation quickly, within 120 msec or less. The sources' rates are allocated properly according to each fairness definition, as listed in Table 2. **Fig. 3** shows the lengths of the queues at switches sw_1 , sw_2 and sw_3 for each type of fairness. The queues at sw_1 and sw_2 are very small, because sources S_1 and S_2 do not use links L_1 and L_2 fully due to the bottleneck at switch sw_3 . The queue occupancy at switch sw_3 remains stable, and oscillates very little after the sources reach their optimal operating rates.

5.2 Robustness of The NK Algorithm

In the second experiment, we increase the sources' ICR to 100 Mbps in order to test the robustness of the NK algorithm. The results in **Fig. 4** and **Fig. 5** show that the algorithm adjusts well to high traffic demand, and allows fast convergence to fair allocation. Fig. 4 shows that the sources reach their fair rates within 250 msec or less, depending on the fairness type. In each case in Fig. 5, the queue at switch sw_3 is overloaded at the beginning, but quickly reduces to stable occupancy

when the sources' bandwidths converge to fair rates. During the stable operating state, the queue length oscillations are very small.

5.3 Comparing The FJ and NK Algorithms

In the third experiment, we compare the FJ algorithm [10] with the source-based case of the NK algorithm (the FJ algorithm supports only source-based fairness). We use the same configuration and simulation setting as in the above two experiments, and run the FJ algorithm. **Fig. 6** shows the sources' ACRs and the lengths of the queues at switches when the sources have an ICR of 25 Mbps. **Fig. 7** shows the results when the sources' ICR is 100 Mbps. Comparing Fig. 2a with Fig. 6a, we see that the NK algorithm converges faster: the convergence time is 105 msec versus 260 msec. Using the NK algorithm, switch sw_3 is also less congested at the beginning: the maximum queue length at sw_3 is 1025 cells versus 3500 cells, as shown in Fig. 3a and 6b respectively. With the NK algorithm, the fluctuations of source rates and queue lengths at the beginning are also smaller. For the experiment in which the sources' ICR is 100 Mbps, we have similar observations when comparing Fig. 4a with Fig. 7a, and Fig. 5a with Fig. 7b.

The NK algorithm offers better performance due to the following two reasons: Firstly, the NK algorithm uses the fair share in the computation of new explicit rates. Secondly, the algorithm also computes new explicit rates on the forward path of RM cells, and updates field ER of FRM cells. These two features help to speed up convergence time, and to minimize the fluctuations of source rates and queue lengths. The FJ algorithm does not implement the above two features.

5.4 Other Experiments

In the configuration in Fig. 1, the bottlenecked link is closer to the destinations. We also carried out the above three experiments with another configuration in which there are more sources, the bottlenecked link is closer to the sources, and the sources have different ICRs. The evaluations drawn from the simulation results of these experiments are consistent with the above observations. The proposed algorithm permits the sources to reach their fair rates quickly. The oscillations of source rates and queue lengths are small. When compared with the FJ algorithm, the source-based case of the NK algorithm performs as well as, or better than the FJ algorithm. This set of experiments is not presented here due to space limitation.

6 Conclusion

Increasing demands for ATM multipoint services have called for efficient multipoint traffic control protocols, including those supporting ABR applications. We have added a new fairness definition for ABR multipoint traffic, and proposed a flexible traffic control scheme for ABR mp-p services, the NK algorithm. The new scheme allows the use of both VC merge and VP merge, and accommodates all existing types of fairness defined for mp-p communications. This flexibility enables the network to meet various bandwidth requirements of a wide range of ABR applications. It also provides switch vendors with more implementation choices for supporting mp-p communications in their switches. Moreover, the NK algorithm is fair, fast, and robust. Simulation results have confirmed the advantages of the proposed algorithm. The NK algorithm can be easily incorporated into an mp-mp traffic control scheme such as the framework suggested by Ren, Siu and Suzuki.

7 References

- [1] A. Charny, D. Clark, and R. Jain, "Congestion Control with Explicit Rate Indication," *Proceedings of ICC '95*, June 1995.
- [2] L. Roberts, "Enhanced PRCA (Proportional Rate-Control Algorithm)," ATM Forum 94-0735R1, August 1994.
- [3] Raj Jain et al., "ERICA Switch Algorithm: A Complete Description," ATM Forum 96-1172, August 1996.
- [4] F.M. Chiussi, et al., "Explicit Rate ABR Scheme Using Traffic Load as Congestion Indicator," *Proceedings of 6th International Conference on Computer Communications and Networks (IC3N'97)*, Las Vegas, Nevada, September 1997.
- [5] J. M. Jaffe, "Bottleneck Flow Control," *IEEE Transactions on Communications*, vol. 29, no. 7, July 1980.
- [6] Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, Sastri Kota, Pradeep Samudra, "Fairness for ABR multipoint-to-point connections," ATM Forum 97-0832, September 1997.
- [7] H. Wang and M. Schwartz, "Achieving bounded fairness for multicast and TCP traffic in the Internet," *Proceedings of ACM SIGCOMM '98*, September 1998.
- [8] W. Ren, K.-Y. Siu, and H. Suzuki, "Multipoint-to-Point ABR Service in ATM Networks," *IEEE International Conference on Communications*, Montreal, Canada, June 1997.
- [9] W. Ren, K.-Y. Siu, H. Suzuki, and M. Shinhara, "Multipoint-to-Multipoint ABR Service in ATM Networks," *Computer Networks and ISDN Systems*, Oct. 1998, vol. 30, no. 19, pp. 1793-1810.
- [10] Sonia Fahmy, et al., "A switch algorithm for ABR multipoint-to-point connections," ATM Forum 97-1085, December 1997.
- [11] W.M. Moh and Y. Chen, "Design and Evaluation of Multipoint-to-Point Multicast Flow Control," *Proceedings of SPIE '98 Conference on Performance and Control of Network Systems II*, vol. 3530, November 1998, pp. 143-154.
- [12] D. Cavendish and M. Gerla, "Rate Based Congestion Control for Many-to-Many Multicast ABR Traffic," *Proceedings of SPIE '98 Conference on Performance and Control of Network Systems II*, vol. 3530, November 1998, pp. 122-130.
- [13] Matthias Grossglauser and K. K. Ramakrishnan, "SEAM: Scalable and Efficient ATM Multipoint-to-Multipoint Multicasting," *Proceedings of IEEE INFOCOM '97*, April 1997.
- [14] Eric Gauthier, Jean-Yves Le Boudec, and D. Dykeman, "SMART: A Many-to-Many Multicast Protocol for ATM," ATM Forum 96-1295, October 1996.
- [15] W. Ren, K.-Y. Siu, and H. Suzuki, "On the Performance of Congestion Control Algorithms for Multicast ABR Service in ATM," *Proceedings of IEEE ATM Workshop '96*, San Francisco, Aug. 1996.
- [16] T. Jiang, M. Ammar, and E. W. Zegura, "Inter-Receiver Fairness: A Novel Performance Measure for Multicast ABR Sessions," *Proceedings of the ACM SIGMETRICS*, June 1998.
- [17] W. Molody Moh, Y. Chen, and B. Niizawa, "Branch-Point Algorithms for Multicasting ATM ABR Protocols," *Proceedings of IEEE Workshop on ATM*, May 1998.
- [18] Sonia Fahmy, et al., "Feedback Consolidation Algorithms for ABR Point-to-Multipoint Connections in ATM Networks," *Proceedings of IEEE INFOCOM '98*, 1998, pp.1004-1013.
- [19] T. Jiang, E. W. Zegura, and M. Ammar, "Improved Consolidation Algorithms for Point-to-Multipoint ABR Service," *Proceedings of IEEE ATM Workshop '98*, Fairfax, VA, May 1998.
- [20] ATM Forum, "ATM Forum Traffic Management Specification Version 4.0," April 1996.

Fig. 2 ACRs of sources (ICR = 25 Mbps)

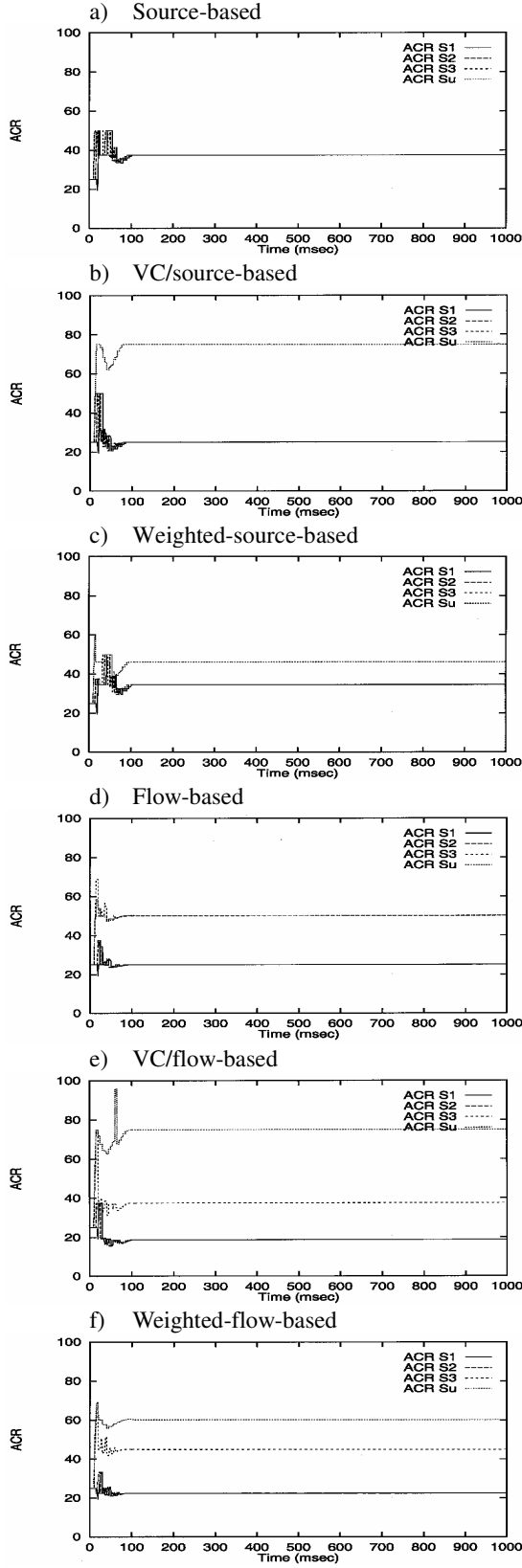


Fig. 3 Queue lengths (ICR = 25 Mbps)

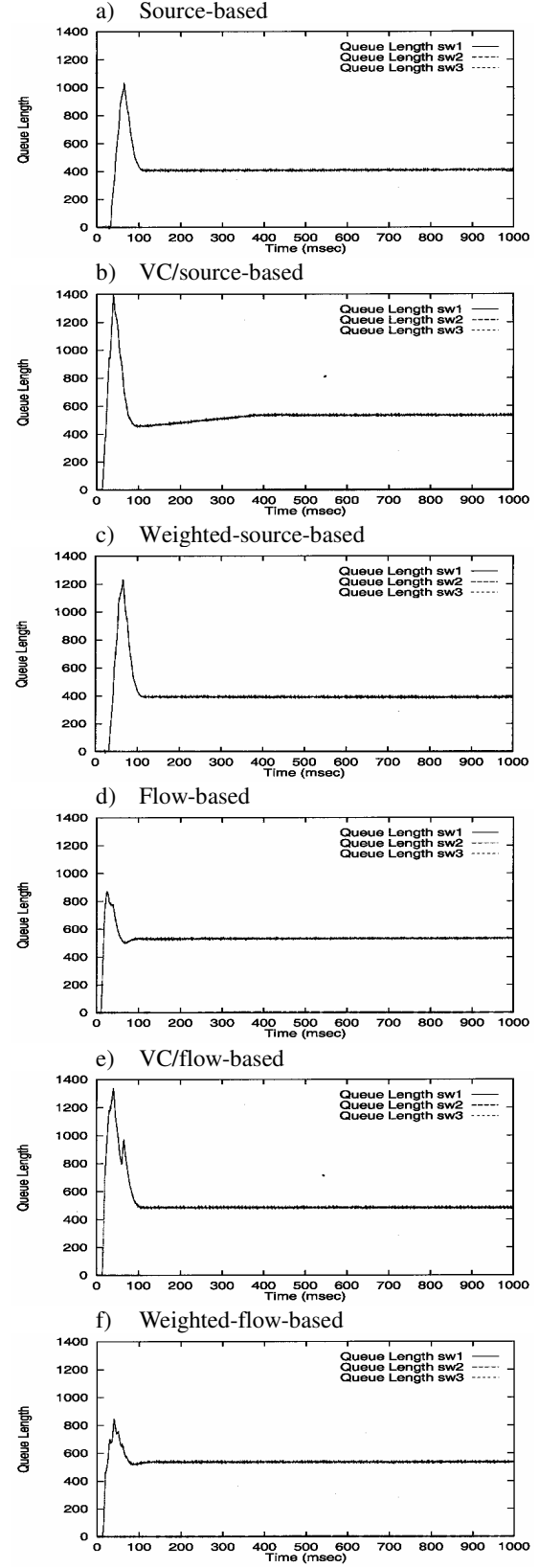


Fig. 4 ACRs of sources (ICR = 100 Mbps)

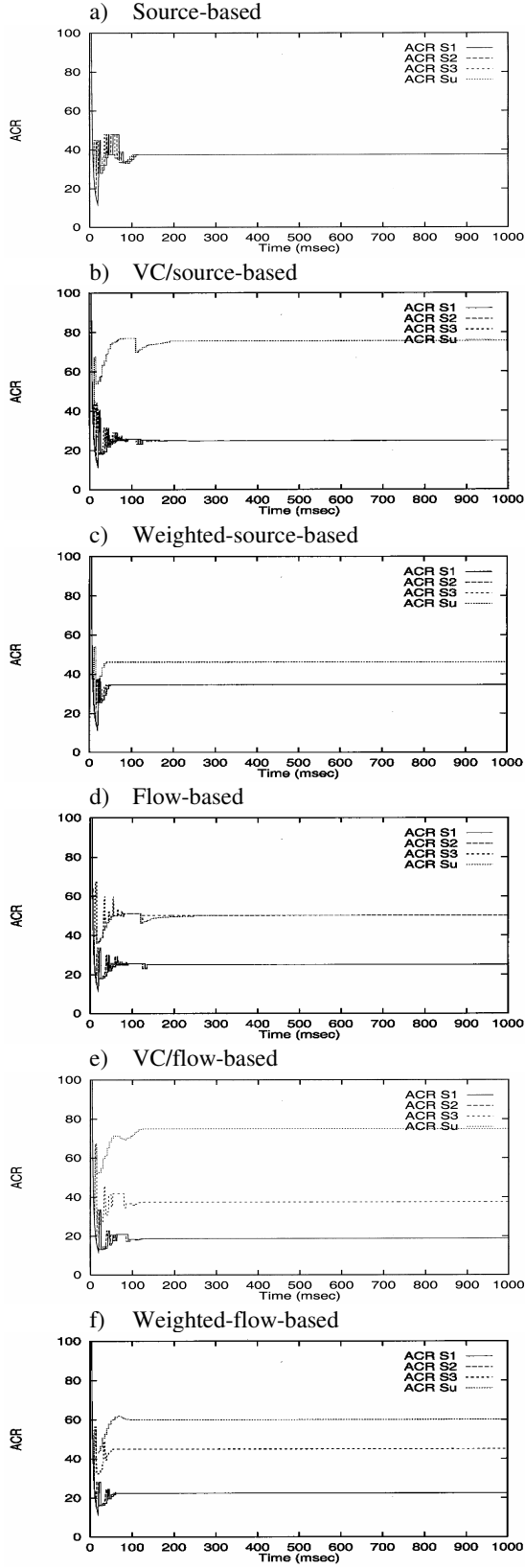


Fig. 5 Queue lengths (ICR = 100 Mbps)

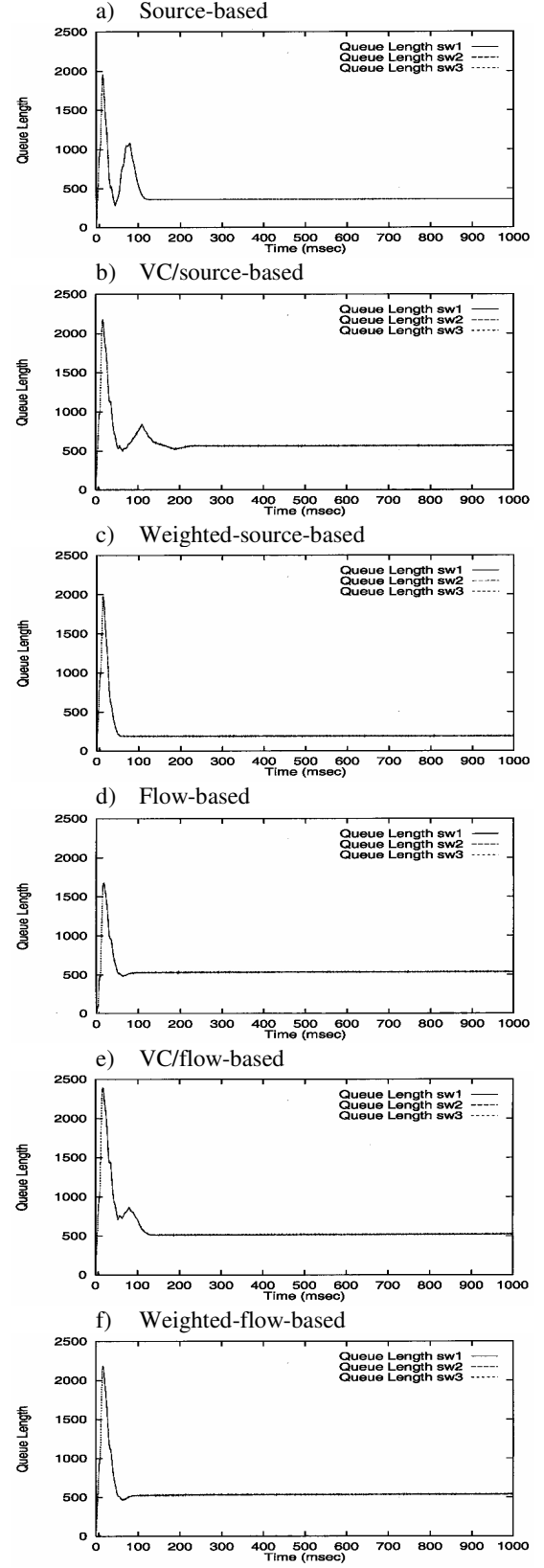


Fig. 6 The FJ algorithm (ICR = 25 Mbps)

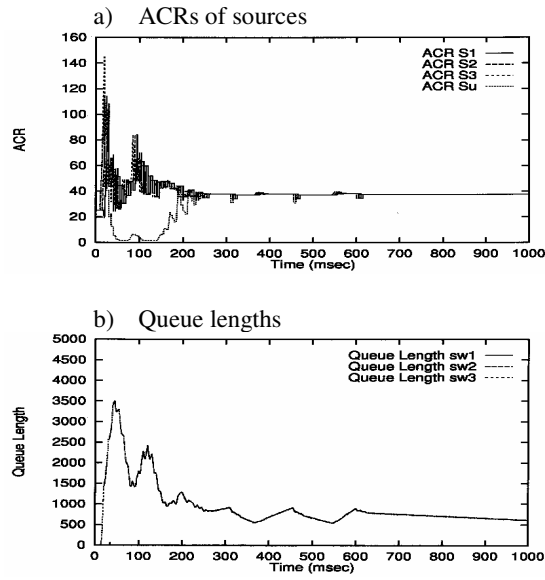


Fig. 7 The FJ algorithm (ICR = 100 Mbps)

