CHAPTER 9

THE VIRTUAL SOURCE/VIRTUAL DESTINATION (VS/VD) FEATURE: DESIGN CONSIDERATIONS

One of the architectural features in the ABR specification is the Virtual Source/Virtual Destination (VS/VD) option. This option allows a switch to divide an end-to-end ABR connection into separately controlled ABR segments by acting like a destination on one segment, and like a source on the other. The coupling in the VS/VD switch between the two ABR control segments is implementation specific. In this section, we model a VS/VD ATM switch and study the issues in designing coupling between ABR segments. We identify a number of implementation options for the coupling. We show that a good choice significantly improves the stability and transient performance of the system and reduces the buffer requirements at the switches.

As mentioned, the VS/VD option allows a switch to divide an ABR connection into separately controlled ABR segments. On one segment, the switch behaves as a destination end system, i.e., it receives data and turns around resource management (RM) cells (which carry rate feedback) to the source end system. On the other segment the switch behaves as a source end system, i.e., it controls the transmission rate of every virtual circuit (VC) and schedules the sending of data and RM cells. We

call such a switch a "VS/VD switch". In effect, the end-to-end control is replaced by segment-by-segment control as shown in Figure 9.1.

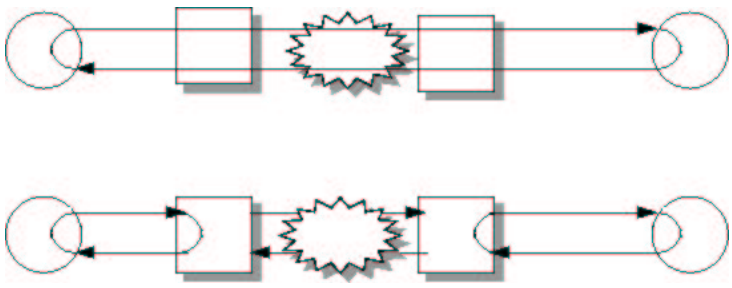


Figure 9.1: End-to-End Control vs VS/VD Control

One advantage of the segment-by-segment control is that it isolates different networks from each other. One example is a proprietary network like frame-relay or circuit-switched network between two ABR segments, which allows end-to-end ABR connection setup across the proprietary network and forwards ATM packets between the ABR segments (signaling support for this possibility is yet to be considered by the ATM Forum). Another example is the interface point between a satellite network and a LAN. The gateway switches at the edge of a satellite network can implement VS/VD to isolate downstream workgroup switches from the effects of the long delay satellite paths (like long queues). A second advantage of segment-by-segment control is that the segments have shorter feedback loops which can potentially improve performance because feedback is given faster to the sources whenever new traffic bursts are seen. The VS/VD option requires the implementation of per-VC queueing and scheduling at the switch.

The goal of this study is find answers to the following questions:

- Do VS/VD switches really improve ABR performance?
- What changes to switch algorithms are required to operate in VS/VD environments?
- Are there any side-effects of having multiple control loops in series?

Specifically, we study the requirements to implement the ERICA algorithm in a VS/VD switch. We describe our switch model and the use of the ERICA algorithm in sections 9.1 and 9.2. The VS/VD design options are listed and evaluated in sections 9.3 and 9.4, and summarized in section 9.6.

9.1 Switch Queue Structure

In this section, we first present a simple switch queue model for the non-VS/VD switch and later extend it to a VS/VD switch by introducing per-VC queues. The flow of data, forward RM (FRM) and backward RM (BRM) cells is also closely examined.

9.1.1 A Non-VS/VD Switch

A minimal non-VS/VD switch has a separate FIFO queue for each of the different service classes (ABR, UBR etc.). We refer to these queues as "per-class" queues. The ABR switch rate allocation algorithm is implemented at every ABR class queue. This model of a non-VS/VD switch based network with per-class queues is illustrated in Figure 9.2.

Besides the switch, the figure shows a source end system, S, and a destination end system, D, each having per-VC queues to control rates of individual VCs. For example, ABR VCs control their Allowed Cell Rates (ACRs) based upon network feedback.

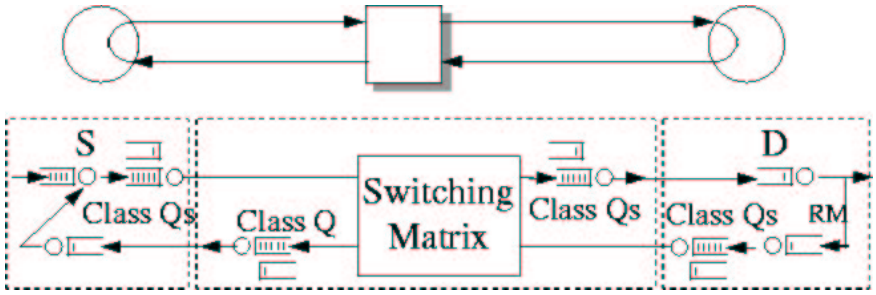


Figure 9.2: Per-class queues in a non-VS/VD switch

We assume that the source/destination per-VC queues feed into corresponding per-class queues (as shown in the figure) which in turn feed to the link. This assumption is not necessary in practice, but simplifies the presentation of the model. The contention for link access between cells from different per-class queues (at the switch, the source and the destination) is resolved through appropriate scheduling.

9.1.2 A VS/VD Switch

The VS/VD switch implements the source and the destination end system functionality in addition to the normal switch functionality. Therefore, like any source and destination end-system, it requires per-VC queues to control the rates of individual VCs. The switch queue structure is now more similar to the source/destination structure where we have per-VC queues feeding into the per-class queues before each link. This switch queue structure and a unidirectional VC operating on it is shown in Figure 9.3.

The VS/VD switch has two parts. The part known as the Virtual Destination (VD) forwards the data cells from the first segment ("previous loop") to the per-VC queue at the Virtual Source (VS) of the second segment ("next loop"). The other part

or the Virtual Source (of the second segment) sends out the data cells and generates FRM cells as specifed in the source end system rules.

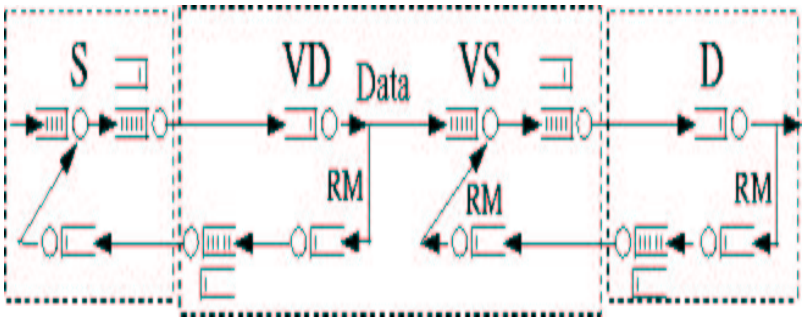


Figure 9.3: Per-VC and per-class queues in a VS/VD switch (a)

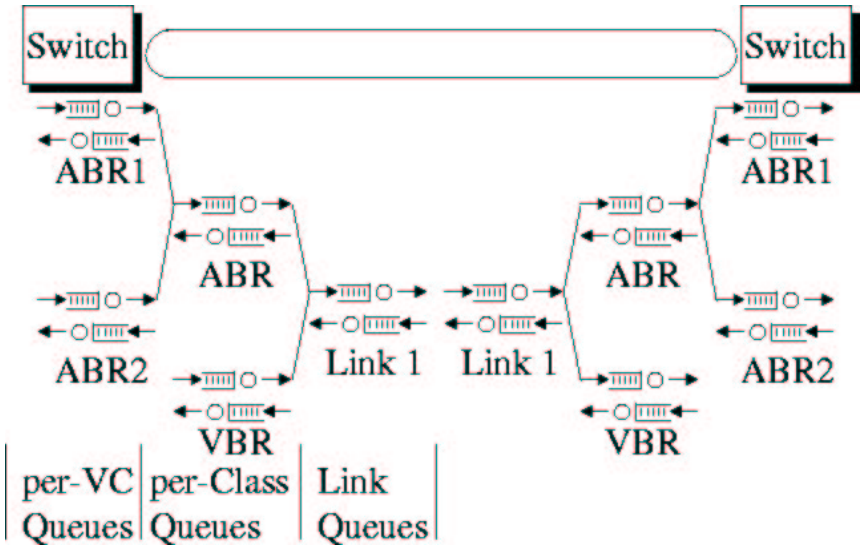


Figure 9.4: Per-VC and per-class queues in a VS/VD switch (b)

The switch also needs to implement the switch congestion control algorithm and calculate the allocations for VCs depending upon its bottleneck rate. A question

which arises is where the rate calculations are done and how the feedback is given to the sources. We postpone the discussion of this question to later sections.

9.1.3 A VS/VD Switch with Unidirectional Data Flow

The actions of the VS/VD switch upon receiving RM cells are as follows. The VD of the previous loop turns around FRM cells as BRM cells to the VS on the same segment (as specified in the destination end system rules (see chapter 2)). Additionally, when the FRM cells are turned around, the switch may decrease the value of the explicit rate (ER) field to account for the bottleneck rate of the next link and the ER from the subsequent ABR segments.

When the VS at the next loop receives a BRM cell, the ACR of the per-VC queue at the VS is updated using the ER field in the BRM (ER of the subsequent ABR segments) as specified in the source end system rules). Additionally, the ER value of the subsequent ABR segments needs to be made known to the VD of the first segment. One way of doing this is for the VD of the first segment to use the ACR of the VC in the VS of the next segment while turning around FRM cells.

The model can be extended to multiple unidirectional VCs in a straightforward way. Figure 9.5 shows two unidirectional VCs, VC1 and VC2, between the same source S and destination D which go from Link1 to Link2 on a VS/VD switch. Observe that there is a separate VS and VD control for each VC. We omit non-ABR queues in this and subsequent figures.

9.1.4 Bi-directional Data Flow

Bi-directional flow in a VS/VD switch (Figure 9.6) is again a simple extension to the above model. The data on the previous loop VD is forwarded to the next loop VS.

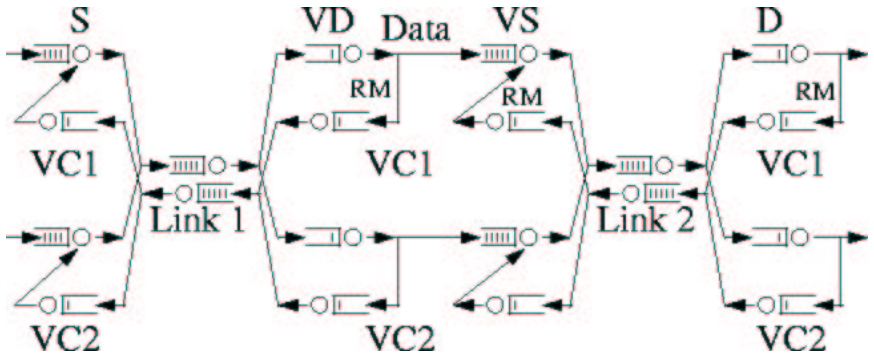


Figure 9.5: Multiple unidirectional VCs in a VS/VD switch

FRMs are turned around by the previous loop VD to the previous loop VS. BRMs are processed by the next loop VS to update the corresponding ACRs.

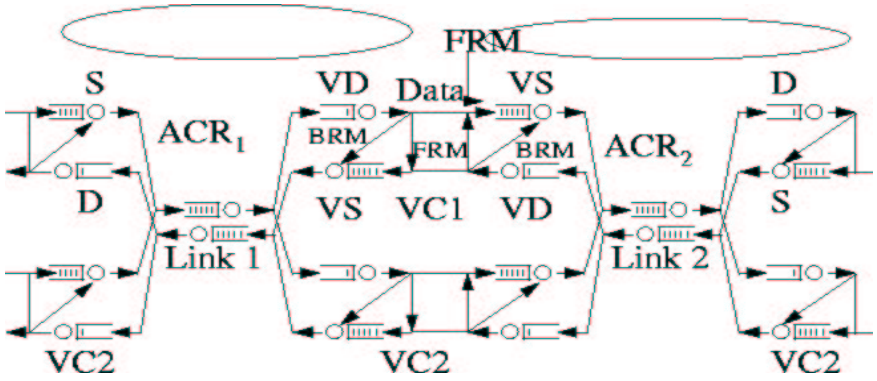


Figure 9.6: Multiple bi-directional VCs in a VS/VD switch

We will discuss the rates and allocations of *VC1 only*. VC1 has two ACRs: ACR_1 in the reverse direction on Link1 and ACR_2 in the forward direction on Link2. Henceforth, the subscript 1 refers to the "previous loop" variables and subscript 2 to the "next loop" variables of VC1.

9.2 The ERICA Switch Scheme: Renotated

In this section, we introduce some new notation for the ERICA algorithm which we use later in this section to explain its implementation in a VS/VD switch.

The ERICA target rate is set as follows:

Target Rate = Target Utilization $\times$ Link Rate - VBR (high priority) Rate.

ERICA measures the input rate to the ABR queue and the number of active ABR sources.

To achieve fairness, the VC's Allocation (VA) has a component:

$VA_{fairness}$ = Target Rate / Number of Active VCs

To achieve efficiency, the VC's Allocation (VA) has a component:

$VA_{efficiency}$ = VC's Current Cell Rate / Overload, where Overload = Input Rate / Target Rate;

Finally, the VC's allocation on this link (VAL) is calculated as:

VAL = Max{ $VA_{efficiency}$, $VA_{fairness}$ } = Function{ Input Rate, VC's current rate }

We use this basic algorithm to illustrate the VS/VD implementation. The implementation of the full scheme can be derived as a simple extension to the description given in this section.

9.2.1 Rate Calculations in a non-VS/VD Switch

The non-VS/VD switch calculates the rate (VAL) for sources when the BRMs are processed in the reverse direction and enters it in the BRM field as follows:

ER in BRM = Min{ ER in BRM, VAL }

At the source end system, the ACR is updated as:

ACR = Function{ ER, VC's current ACR }

9.2.2 Rate Calculations in a VS/VD Switch

Figure 9.7 shows the rate calculations in a VS/VD switch. Specifically, the segment starting at Link2 ("next loop") returns an ER value, ER_2 in the BRM, and the FRM of the first segment ("previous loop") is turned around with an ER value of ER_1. The ERICA algorithm for the port to Link2 calculates a rate (VAL_2) as: VAL_2 = Function { Input Rate, VC's Current Rate }. The rate calculations at the VS and VD are as follows:

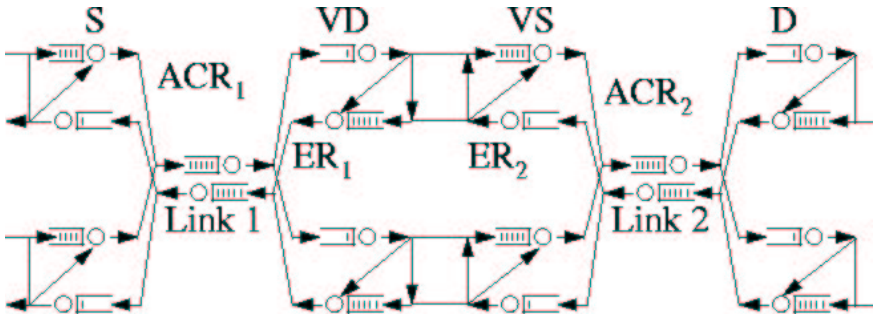


Figure 9.7: Rate calculations in VS/VD switches

- Destination algorithm for the *previous loop*:

  ER_1 = Min { ER_1, VAL_2, ACR_2 }

- Source Algorithm for the *next loop*:

  Optionally, ER_2 = Min { ER_2, VAL_2 }

  ACR_2 = Fn { ER_2, ACR_2 }

The unknowns in the above equations are the input rate and the VC's current rate. We shall see in the next section that there are several ways of measuring VC rates and input rates, combining the feedback from the next loop, and updating the ACR of the next loop. Note that though different switches may implement different algorithms, many measure quantities such as the VC's current rate and the ABR input rate.

9.3 VS/VD Switch Design Options

In this section, we aim at answering the following questions:

- What is a VC's current rate? (4 options)
- What is the input rate? (2 options)
- Does the congestion control actions at a link affect the next loop or the previous loop? (3 options)
- When is the VC's allocation at the link (VAL) calculated? (3 options)

We will enumerate the 72 ($= 4 \times 2 \times 3 \times 3$) option combinations and then study this state space for the best combination.

9.3.1 Measuring the VC's Current Rate

There are four methods to measure the VC's current rate:

1. The rate of the VC is declared by the source end system of the previous loop in the Current Cell Rate (CCR) field of the FRM cell (FRM1) received by the VD. This declared value can be used as the VC's rate.

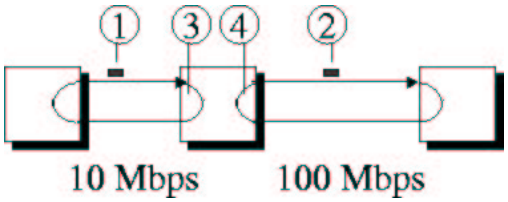


Figure 9.8: Four methods to measure the rate of a VC at the VS/VD switch

2. The VS to the next loop declares the CCR value of the FRM sent (FRM2) to be its ACR (ACR_2). This declared value can be used as the VC's rate.

3. The actual source rate in the *previous loop* can be measured. This rate is equal to the VC's input rate to the per-VC queue. This measured source rate can be used as the VC's rate.

4. The actual source rate in the *next loop* can be measured as the VC's input rate to the per-class queue (from the per-VC queue). This measured value can be used as the VC's rate.

Figure 9.8 illustrates where each method is applied (note the position of the numbers in circles).

9.3.2 Measuring the Input Rate at the Switch

Figure 9.9 (note the position of the numbers in circles) shows two methods of estimating the input rate for use in the switch algorithm calculations. These two methods are:

1. The input rate is the sum of input rates to the per-VC ABR queues.

2. The input rate is the aggregate input rate to the per-class ABR queue.

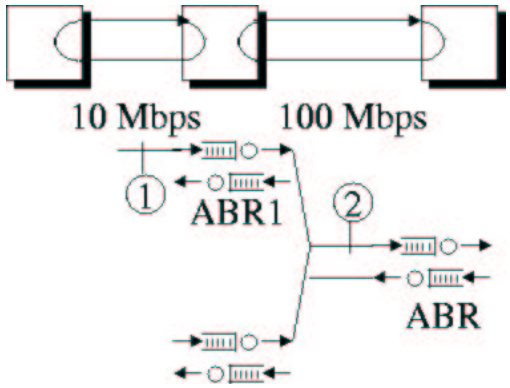


Figure 9.9: Two methods to measure the input rate at the VS/VD switch

9.3.3 Effect of Link Congestion Actions on Neighboring Links

The link congestion control actions can affect neighboring links. The following actions are possible in response to the link congestion of Link2:

1. Change ER_1. This affects the rate of the *previous loop only*. The change in rate is experienced only after a feedback delay equal to twice the propogation delay of the loop.

2. Change ACR_2. This affects the rate of the *next loop only*. The change in rate is experienced instantaneously.

3. Change ER_1 and ACR_2. This affects *both the previous and the next loop*. The next loop is affected instantaneously while the previous loop is affected after a feedback delay as in the first case.

9.3.4 Frequency of Updating the Allocated Rate

Recall that the ERICA algorithm in a non-VS/VD switch calculates the allocated rate when a BRM cell is processed in a switch. However, in a VS/VD switch, there are three options as shown in Figure 9.10:

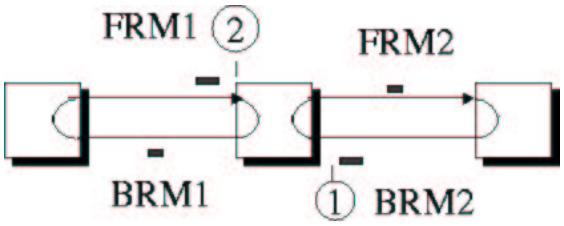


Figure 9.10: Three methods to update the allocated rate

1. Calculate allocated rate *on receiving BRM2 only.* Store the value in a table and use this table value when an FRM is turned around.

2. Calculate allocated rate *only when FRM1 is turned around.*

3. Calculate allocated rate *both when FRM1 is turned around as well as when BRM2 is received.*

In the next section, we discuss the various options and present analytical arguments to eliminate certain design combinations.

9.4 VS/VD Switch Design Options

9.4.1 VC Rate Measurement Techniques

We have presented four ways of finding the the VC's current rate in section 9.3.1, two of them used declared rates and two of them measured the actual source rate. We show that measuring source rates is better than using declared rates for two reasons.

First, the declared VC rate of a loop naively is the minimum of bottleneck rates of *downstream loops only.* It does not consider the bottleneck rates of upstream loops, and may or may not consider the bottleneck rate of the first link of the next loop. Measurement allows better estimation of load when the traffic is not regular.

Second, the actual rate of the VC may be lower than the declared ACR of the VC due to dynamic changes in bottleneck rates upstream of the current switch. The difference in ACR and VC rate will remain *at least* as long as the time required for new feedback from the bottleneck in the path to reach the source plus the time for the new VC rate to be experienced at the switch. The sum of these two delay components is called the "feedback delay." Due to feedback delay, it is possible that the declared rate is a stale value at any point of time. This is especially true in VS/VD switches where per-VC queues may control source rates to values quite different from their declared rates.

Further, the measured source rate is already available in a VS/VD switch because it is measured as part of one of the source end system rules (SES Rule 5) (see chapter 7).

9.4.2 Input Rate measurement techniques

As discussed earlier, the input rate can be measured as the sum of the input rates of VCs to the per-VC queues or the aggregate input rate to the per-class queue. These two rates can be different because the input rate to the per-VC queues is at the previous loop's rate while the input to the per-class queue is related to the next loop's rate. Figure 9.11 shows a simple case where two adjacent loops can run at very different rates (10 Mbps and 100Mbps) for one feedback delay.

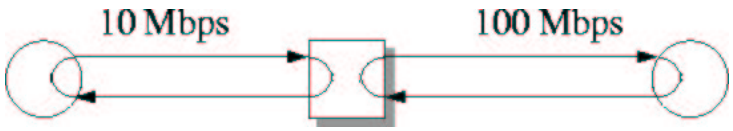


Figure 9.11: Two adjacent loops may operate at very different rates for one feedback delay

9.4.3 Combinations of VC rate and input rate measurement options

Table 9.1 summarizes the option combinations considering the fact that two adjacent loops may run at different rates. The table shows that four of these combinations may work satisfactorily. The other combinations use inconsistent information and hence may either overallocate rates leading to unconstrained queues or result in unnecessary oscillations. We can eliminate some more cases as discussed below.

The above table does not make any assumptions about the queue lengths at any of the queues (per-VC or per-class). For example, when the queue lengths are close to

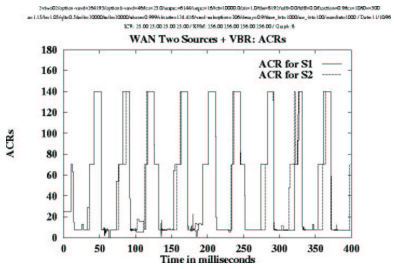
#	VC Rate Method rates (Mbps)	Σ VC	Input Rate Method	Input Rate Value	Choice (YES/NO)
1.	From FRM1	10	Σ per-VC	10	**YES**
2.	From FRM1	10	per-class	10-100	NO
3.	From FRM2	100	Σ per-VC	10	NO
4.	From FRM2	100	per-class	100	**YES**
5.	At per-VC queue	10	Σ per-VC	10	**YES**
6.	At per-VC queue	10	per-class	10-100	NO
7.	At per-class queue	100	Σ per-VC	10	NO
8.	At per-class queue	100	per-class	100	**YES**

Table 9.1: Viable combinations of VC rate and input rate measurement

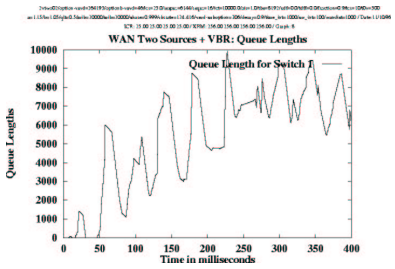
zero, the actual source rate might be much lower than the declared rate in the FRMs leading to overallocation of rates. This criterion can be used to reject more options.

The performance of one such rejected case is shown in Figure 9.12 (corresponding to row 4 in Table 9.1). The configuration used has two ABR infinite sources and one high priority VBR source contending for the bottleneck link's (LINK1) bandwidth. The VBR has an ON/OFF pattern, where it uses 80% of the link capacity when ON. The ON time and the OFF time are equal (20 ms each). The VS/VD switch overallocates rates when the VBR source is OFF. This leads to ABR queue backlogs when the VBR source comes ON in the next cycle. The queue backlogs are never cleared, and hence the queues diverge. *In this case, the fast response of VS/VD is harmful* because the rates are overallocated.

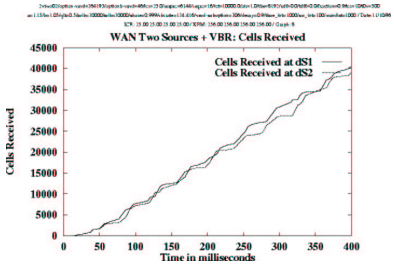
In this study, we have not evaluated row 5 of the table (measurement of VC rate at entry to the per-VC queues). Hence, out of the total of 8 combinations, we consider two viable combinations: row 1 and row 8 of the table. Note that since row



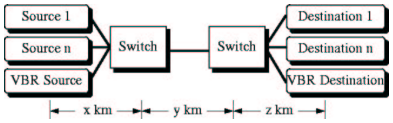
(a) ACR



(b) Queue Lengths



(c) Cells Received



(d) Configuration

Figure 9.12: 2-source+VBR configuration. Unconstrained queues due to overallocation.

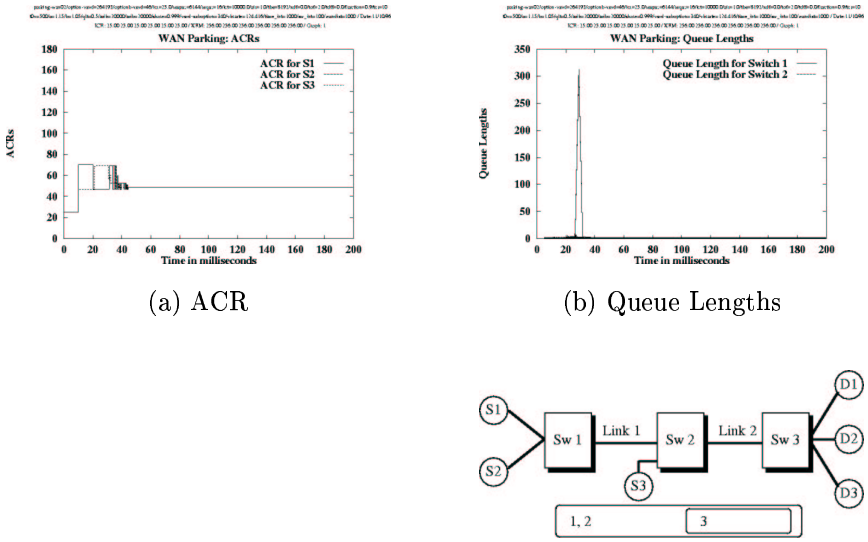
8 uses source rate measurement, we expect it to show better performance. This is substantiated by our simulation results presented later in the paper.

9.4.4 Effect of Link Congestion Control Actions

In a network with non-VS/VD switches only, the bottleneck rate needs to reach the sources before any corresponding load change is seen in the network. However, a VS/VD switch can enforce the new bottleneck rate immediately (by changing the ACR of the per-VC queue at the VS). This rate enforcement affects the utilization of links in the next loop. Hence, the VS/VD link congestion control actions can affect neighboring loops. We have enumerated three options in an earlier section.

We note that the second option ("next loop only") does not work because the congestion information is not propagated to the sources of the congestion (as required by the standard [35]). This leaves us with two alternatives. The third option ("both loops") is attractive because, when ACR_2 is updated, the switches in the next loop experience the load change faster. Switch algorithms may save a few iterations and converge faster in these cases.

Figure 9.13 shows the fast convergence in a parking lot configuration when such a VS/VD switch is used. The parking lot configuration (Figure 9.13(c)) consists of three VCs contending for the Sw2-Sw3 link bandwidth. Link lengths are 1000 km and link bandwidths are 155.52 Mbps. The target rate of the ERICA algorithm was 90% of link bandwidth i.e., 139.97 Mbps. The round trip time for the S3-D3 VC is shorter than the round trip time for the other two VCs. The optimum allocation by ERICA for each source is 1/3 of the target rate on the Sw2-Sw3 (about 46.7 Mbps).



(a) ACR

(b) Queue Lengths

(c) Configuration

Figure 9.13: Parking lot: best VS/VD option converges fast

Figure 9.13(a) shows that the optimum value is reached at 40 ms. Part (b) of the figure shows that the transient queues are small. Further, the allocation is fair.

9.4.5 Link Congestion and Allocated Rate Update Frequency: Viable Options

The allocated rate update has three options:

a) update upon BRM receipt (in VS) and enter the value in a table to be used when an FRM is turned around,

b) update upon FRM turnaround (at VD) and no action at VS,

c) update both at FRM (VD) and at BRM (VS) without use of a table.

The last option recomputes the allocated rate a larger number of times, but can potentially allocate rates better because we always use the latest information.

The allocated rate update and the effects of link congestion actions interact as shown in Figure 9.14. The figure shows a tree where the first level considers the link congestion (2 options), i.e., whether the next loop is also affected or not. The second level lists the three options for the allocated rate update frequency. The viable options are those highlighted in bold at the leaf level.

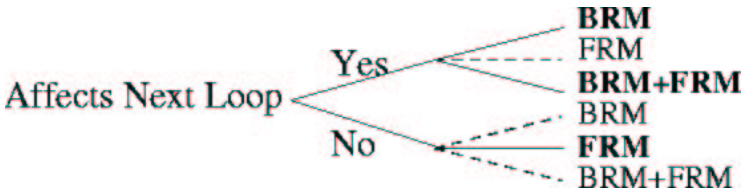


Figure 9.14: Link congestion and allocated rate update: viable options

Other options are not viable because of the following reasons. In particular, if the link congestion does not affect the next loop, the allocated rate update at the FRM turnaround is all that is required. The allocated rate at the BRM is redundant in this case. Further, if the link congestion affects the next loop, then the allocated rate update has to be done on receiving a BRM, so that ACR can be changed at the VS. This gives us two possibilities as shown in the figure (BRM only, and BRM+FRM).

Hence, we have three viable combinations of link congestion and the allocated rate update frequency. A summary of all viable options (a total of 6) is listed in Table 9.2.

The next section evaluates the performance of the viable VS/VD design options through simulation.

Option #	VC Rate Method	Input Rate Measurement point	Link Congestion Effect	Allocated Rate Updated at
A	From FRM1	per-VC	prev loop only	FRM1 only
B	At per-class Q	per-class	both loops	FRM1 only
C	From FRM1	per-VC	both loops	FRM1 only
D	At per-class Q	per-class	both loops	FRM1 and BRM2
E	From FRM1	per-VC	both loops	BRM2 only
F	At per-class Q	per-class	both loops	BRM2 only

Table 9.2: Summary of viable VS/VD design alternatives

9.5 Performance Evaluation of VS/VD Design Options

9.5.1 Metrics

We use four metrics to evaluate the performance of these alternatives:

- **Response Time:** is the time taken to reach near optimal behavior on startup.
- **Convergence Time:** is the time for rate oscillations to decrease (time to reach the steady state).
- **Throughput:** Total data transferred per unit time.
- **Maximum Queue:** The maximum queue before convergence.

The difference between response time and convergence time is illustrated in Figure 9.15. The following sections present simulation results with respect to the above metrics. Note that we have used greedy (infinite) traffic sources in our simulations. We have studied the algorithmic enhancements in non-VS/VD switches for non-greedy

sources in chapter 6. We expect the best implementation option (see below) to work well and produce consistent results when such (bursty) traffic is used.

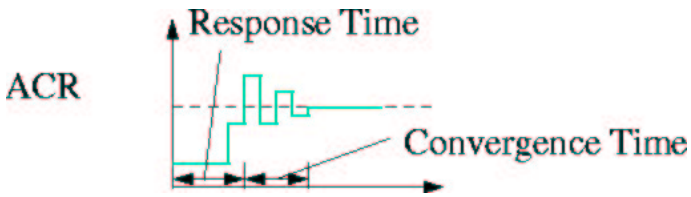


Figure 9.15: Response time vs Convergence time

Response Time

Without VS/VD all response times are close to the round-trip delay. With VS/VD, the response times are close to the feedback delay from the bottleneck. Since VS/VD reduces the response time during the first round trip, it is good for long delay paths. The quick response time (10 ms in the parking lot configuration which has a 30 ms round trip time) is shown in Figure 9.13.

Response time is also important for bursty traffic like TCP file transfer over ATM which "starts up" at the beginning of every active period (when the TCP window increases) after the corresponding idle period (see chapter 7).

Throughput

The number of cells received at the destination is a measure of the throughput achieved. These values are listed in Table 9.3. The top row is a list of the configuration codes (these codes are explained in Table 9.2. The final column lists the throughput values for the case when a non-VS/VD switch is used. The 2 source+VBR and the parking lot configurations have been introduced in earlier section.

The upstream bottleneck configuration shown in Figure 5.19 has a bottleneck at Sw1 where 15 VCs share the Sw1-Sw2 link. As a result the S15-D15 VC is not capable of utilizing its bandwidth share at the Sw2-Sw3 link. This excess bandwidth needs to be shared equally by the other two VCs. The table entry shows the number of cells received at the destination for either the S16-D16 VC or the S17-D17 VC.

In the 2 source+VBR and the upstream bottleneck configurations, the simulation was run for 400 ms (the destination receives data from time = 15 ms through 400 ms). In the parking lot configuration, the simulation was run for 200ms.

VS/VD Opt # → Config ↓	A	B	C	D	E	F	No VS/VD
2 source + VBR	31	31	32.5	34	32	33	30
Parking lot	22	22	23	20.5	23	20.5	19.5
Upstream bottleneck	61	61	61	60	61	61	62

Table 9.3: Cells received at the destination per source in Kcells

As we compare the values in each row of the table, we find that, in general, there is *little difference between the alternatives in terms of throughput.* However, there is a slight increase in throughput when VS/VD is used over the case without VS/VD switch.

Convergence Time

The convergence time is a measure of how fast the scheme finishes the transient phase and reaches steady state. It is also sometimes called "transient response." The convergence times of the various options are shown in Table 9.4. The "transient"

configuration mentioned in the table has two ABR VCs sharing a bottleneck (like the 2 source + VBR configuration, but without the VBR VC). One of the VCs comes on in the middle of the simulation and remains active for a period of 60 ms before going off.

VS/VD Opt # → Config ↓	A	B	C	D	E	F	No VS/VD
Transient	50	50	65	**20**	55	25	60
Parking lot	120	100	170	**45**	125	50	140
Upstream bottleneck	95	75	75	**20**	95	**20**	70

Table 9.4: Convergence time in ms

Observe that the convergence time of VS/VD option D (highlighted) is the best. Recall that this configuration corresponds to measuring the VC rate at the entry to the per-class queue, input rate measured at the per-class queue, link congestion affecting both the next loop and the previous loop, the allocated rate updated at both FRM1 and BRM2.

Maximum Transient Queue Length

The maximum transient queues gives a measure of how askew the allocations were when compared to the optimal allocation and how soon this was corrected. The maximum transient queues are tabulated for various configurations for each VS/VD option and for the case without VS/VD in Table 9.5.

The table shows that VS/VD option D has very small transient queues in all the configurations and the minimum queues in a majority of cases. This result, combined

VS/VD Opt # → Config ↓	A	B	C	D	E	F	No VS/VD
2 Source + VBR	**1.2**	1.4	2.7	1.8	2.7	1.8	2.7
Transient	1.4	1.1	1.4	**0.025**	1.3	1.0	6.0
Parking lot	1.9	1.9	1.4	**0.3**	3.7	0.35	2.0
Upstream bottleneck	0.025	0.08	0.3	**0.005**	1.3	**0.005**	0.19

Table 9.5: Maximum queue length in Kcells

with the fastest response and near-maximum throughput behavior confirms the choice of option D as the best VS/VD implementation.

Observe that the queues for the VS/VD implementations are in general lesser than or equal to the queues for the case without VS/VD. However, the queues reduce much more if the correct implementation (like option D) is chosen.

9.6 Conclusions

In summary:

- VS/VD is an option that can be added to switches which implement per-VC queueing. The addition can potentially yield improved performance in terms of response time, convergence time, and smaller queues. This is especially useful for switches at the edge of satellite networks or switches that are attached to links with large delay-bandwidth product. The fast response and convergence times also help support bursty traffic like data more efficiently.

- The effect of VS/VD depends upon the switch algorithm used and how it is implemented in the VS/VD switch. The convergence time and transient queues

can be very different for different VS/VD implementations of the same basic switch algorithm. In such cases the fast response of VS/VD is harmful.

- With VS/VD, ACR and actual rates are very different. The switch cannot rely on the RM cell CCR field. We recommend that the VS/VD switch and in general, switches implementing per-VC queueing measure the VC's current rate.

- The sum of the input rates to per-VC VS queues is not the same as the input rate to the link. It is best to measure the VC's rate at the output of the VS and the input rate at the entry to the per-class queue.

- On detecting link congestion, the congestion information should be forwarded to the previous loop as well as the next loop. This method reduces the convergence time by reducing the number of iterations required in the switch algorithms on the current and downstream switches.

- It is best for the the rate allocated to a VC to be calculated both when turning around FRMs at the VD as well as after receiving BRMs at the next VS.

We have shown that the VS/VD provision in the ABR traffic management framework can potentially improve performance of bursty traffic and reduce the buffer requirements in switches. The VS/VD mechanism achieves this by breaking up a large ABR loop into smaller ABR loops which are separately controlled. However, further study is required in the following areas:

- Effect of VS/VD on buffer requirements in the switch.

- Scheduling issues with VS/VD.

- Effect of different switch algorithms in different control loops, and different control loop lengths.

- Effect of non-ABR clouds and standardization issues involved.

CHAPTER 10

IMPLEMENTATION ISSUES

At the time of this writing, the Traffic Management 4.0 [35] which includes the ABR specification has been available for a year and a half. However, the first products implementing ABR are just entering the market. The reason for this long delay is in part because of the complexity of ABR implementation. We explore some of the issues in this chapter and study the implementation and performance of one of the ABR options, namely, Virtual Source/Virtual Destination, in depth. We will also mention some of the efforts currently underway to make the ABR service more attractive.

10.1 ATM Service Categories Revisited

ATM provides multiple classes of service to realize the goal of an integrated services network. The CBR and VBR services were designed primarily for voice and isochronous traffic like video. These services required the network to reserve resources. As a result, the method used to reserve resources limited the total number of CBR or VBR connections that could be setup. Data traffic did not require such resource reservations, and it could potentially use the bandwidth "left over" by CBR and VBR. Therefore, the ATM Forum decided to develop a "best-effort" service category for data traffic which uses the "left over" capacity on a physical channel. Initial

ATM data users found that their packets were being dropped indiscriminately by the network. The reason was that due to fragmentation, even a single cell loss resulted in a packet loss.

So, there was a need for a service which provides control over cell loss. ABR was initially designed to meet this need through the use of feedback control. Basically, the network explicitly distributes the "left over" capacity among the active ABR sources. During the development of the ABR service, it was realized that feedback control could also be used to provide high throughput, low delay and fairness among contending sources. The tradeoff was performance versus complexity. There was the debate between credit-based framework and the rate-based framework, and the latter was standardized because it mandated lesser *required* complexity. The standard requires the network interface card (NIC) manufacturers to implement a set of source and destination end system rules. The switches minimally need to give some kind of feedback. They can set EFCI bits on data cells and/or process RM (control) cells sent by the sources once every Nrm cell times to give feedback. Target ABR applications are file transfer, WWW, email, variable quality video and voice.

The UBR service is "unspecified" in the sense that the only standard support required from switches is the capability to accept a UBR connection request from the source. By default, there is no resource to be reserved and the connection admission control (CAC) procedure is very simple. Another implication of the service being unspecified is that nothing is guaranteed. In particular, if network gets congested, UBR cells may be dropped. The network switches may provide an enhanced UBR service by using techniques like intelligent drop policies, buffer allocation and scheduling [39]. Network monitoring traffic, email and news are examples of the UBR applications.

With an enhanced UBR service, applications like file transfer and WWW browsing and downloading become viable over this service.

The service categories of ATM can also be compared to those offered by airlines [68]. CBR is confirmed reservations with no recourse if you do not show up. VBR is like confirmed reservation but you do not pay if you do not show up. ABR is standby. You get to go if seats are available. Having standby service is good for the airline. They can fill their seats that would otherwise would have gone empty. The service is also good for passengers. They can travel cheaply particularly if they don't have to be at their destinations at a certain time. UBR service is not currently offered by the airlines. Passengers travelling on UBR class may be allowed to board a plane but may be strangled at the subsequent airports forever if seats are not available. ABR users would generally be asked to stay home as much as possible if their routes are congested.

10.2 Issues in ABR Implementation Complexity

From an architecture viewpoint, currently, ABR is a complex service to implement. The important architectural tradeoffs we will encounter involve requirements in terms of processing speed, latency, memory, and compactness. We will encounter processing speed mismatches for RM cells versus data cells. Further, the RM cell might need to be processed in both the forward and reverse directions. The latency issue arises when RM cells are processed separately, and/or in software, and/or block on a slow shared DRAM for information access. Memory requirements and access speed requirements vary depending upon the RM cell processing strategy. Compactness and overall cost depends upon the particular implementation (for example: ASIC or FPGA). In this

section, we outline some of the implementation problems for both switches and NICs, and suggest solutions.

10.2.1 Switch issues

1. ABR requires the switch to process RM cells. The processing of RM cells takes a longer time than processing data cells. As a result, the processing of RM cells may disrupt the switch pipeline mechanism. Note that a pipeline mechanism processes a job in several stages of a "pipeline" and assumes that the processing time at each stage is simple and involves the same (small) amount of time. Any task with disproportionate processing requirements disrupts the pipeline. One solution is to extract such tasks from the stream before they enter the pipeline, process them separately, and reinsert them into the stream. In this case, we require a special hardware/software design to extract, process and reinsert RM cells to/from the ABR VC. Note that this solution might extract an RM cell from one point in the stream and reinsert it at a different point. However, the traffic management 4.0 standard allows RM cells to be extracted, processed separately, and reinserted, as long as the RM cell sequence within each VC is maintained. Note that the correlation of the declared parameters with the actual stream is lost under such conditions. For example, the CCR field may not be indicative of the rate of the VC (as measured) when the RM cell is processed. Software processing of RM cells is possible if the Nrm (RM cell frequeny parameter) is negotiated appropriately (eg: use a value like 192, instead of the default value, 32).

2. Some switch schemes have a processing requirement for both the forward and backward going RM cell. This requires extra processing at the switch. Another related problem which arises in this case is that the switch scheme requires exchange of information from one port to another (the ingress chip to the egress chip). The assumption made by the scheme is that the switch has a shared memory which is used for tables facilitating the information exchange. This assumption is based on the fact that early switches had the VC table (which maps a cell of a VC from one port to another) was in such a shared memory. The problem with the shared memory is that in the worst case it needs to support accesses from all ports in a single cell time. Modern switches have evolved to use cheaper (and slower memory) to build a distributed VC table – based on the assumption that VC label allocations are relatively static (written only during connection setup, read by the local port only), and local between pairs of ports (except point-to-multipoint VCs which could involve multiple ports). One disadvantage of the distributed memory implementation is that sharing information between ports via memory is not possible. A solution to this problem is to have a cheap low speed shared memory (DRAM) for storing shared tables which are accessed when RM cells are processed. We take advantage of the fact that RM cells on every VC arrive at a frequency of at most one in $Nrm = 32$ cells. Even in the case when RM cells of multiple VCs arrive together, they need to be processed at a rate much smaller than the link rate. As mentioned before, RM cells can be staggered with respect to the data stream as long as the sequence integrity on a VC is maintained.

The ERICA scheme also works best when calculations are done at the receipt of the FRM and the BRM cell, and information is exchanged. However, it is possible to implement ERICA such that feedback can be given when the RM cell is seen in the forward direction. In this implementation, certain fields of the ERICA table (eg, the CCR field) are not required. Further, the per-VC state can be stored in memory local to the port. Also, the computations usually done when an RM cell is received can be avoided by precalculating the feedback at the end of an averaging interval for the set of active sources. In general, the averaging interval computation can be done in software as a background process. A lazy evaluation technique for the same is also possible.

3. Many switch implementations provide per-VC queueing and scheduling in order to ensure isolation of traffic and provide fairness among VCs. The ERICA algorithm does not require per-VC queueing and scheduling. But, it does assume that misbehaving sources (which do not send data according to their allocations). In a corporate network, the source end-system cards or NICs can be chosen such that they schedule the traffic depending upon the current rate. If the NIC technology cannot handle the scheduling of cells when ACRs vary rapidly, the VC output rates at the NICs may be close to, but not conform to ACRs. Under such conditions, the policing function needs to be done at the edge switch. This switch does require per-VC queues (but a smaller number because it is an edge switch), large buffers, and it needs to monitor and enforce the ACRs of VCs. The non-edge switches can provide simple FIFO queueing, and relatively smaller buffers, simple drop policies, and tradeoff the complexity of the switch feedback scheme.

4. Large legacy switches have a problem in that they are very expensive to replace, but provide few hooks for adding new functionality. Counters, registers and pin-outs are always at a premium on a chip, and are rarely left unused. For example, one might decide to use a switch hook (a pin in the chip) to implement an improved switch scheme. However, if the scheme requires measurement of different quantities, it cannot be done due to the lack of hooks. For example, the ERICA switch scheme requires the measurement of quantities such as the input rate and the number of active ABR sources. But, the only available metric in the switch might be just the queue length, which does not allow the implementation of the algorithm to be retrofitted on the switch. Simple algorithms which use just the queue metric need to be used for such cases.

5. Some ABR features such as the Virtual Source/Virtual Destination feature require the implementation of per-VC queueing. Recall that the requirement of per-VC queueing was one of the key reasons why the credit-based framework was rejected. It might seem contradictory to see per-VC queueing implemented by all major vendors. However, note that the current switches typically support upto 128K VCs per port. When the number of VCs grows further (millions of VCs), the accounting information required and the scheduling overhead is expected to become prohibitively expensive. In such cases, VCs will be aggregated into classes and supported by a few thousand class queues.

   VS/VD, on the other hand, requires the implementation of the source end system rules, scheduling of VC cells at a variable ACR, and the maintainence of a large amount of state per-VC. This has resulted in the VS/VD option to be implemented only in very large switches (like satellite switches) where the

advantages of the mechanism justify the cost of implementation. The implementation issues of VS/VD are further discussed in section 9.

6. Another issue of importance to long-distance ATM service providers is how to price the ABR service for ISPs. This requires switch support in terms of management software for usage-based billing. The definition of "usage" is "the number of cells delivered to the destination end-system." This definition implies that the measurements of usage have to be made in the egress switches. This adds cost and complexity to the implementation of edge switches. Currently, no cost estimates exist for ABR service.

7. There is a cost-performance tradeoff in implementing the various options of the ABR service. LAN switches are typically lower end, and the EFCI feedback mechanism provides sufficient performance, since the round trip times are small. It is anticipated that ABR will be the service of choice for WAN and satellite networks. This is because the ABR service (ER-based implementations) can provide throughput and delay performance, and is more scalable in terms of buffer requirements than the UBR service. WAN switches are expected to use ER-based ABR implementations. Complex alternatives like per-VC queueing and scheduling are required for ABR only at the edge switches. Interior network switches typically would use ER-based feedback, simple FIFO queueing, allocate small amount of buffers, and have simple buffer management and cell drop policies. The VS/VD alternative is required at a few edge switches of a very large delay-bandwidth product network. This option allows network managers to isolate the effects of the large network on downstream small networks (see

section 9. Further, WAN switches would typically allocate some bandwidth for aggregate ABR traffic to avoid zero capacity problems during congestion (when the feedback loops are disrupted), and in general to reduce the variation in available capacity for the service (which allows the switch algorithm to allocate rates more aggressively without worrying about effects of errors due to variation).

10.2.2 End-system issues

1. The end-system (which is the network interface card (NIC) for end-to-end ATM, or an edge router in a backbone network) needs to implement the end-system rules for ABR as specified in the standard. A mechanism needed for this implementation is one which schedules cells of different VCs based upon a dynamically changing set of ACRs. In practice, only a few ACR levels may be possible which can lead to link underutilization. SES Rule 9 allows the source to reschedule a cell on a VC based upon a new rate allocation. The implementation of this mechanism per-VC can be difficult.

2. In the ABR service, the network allocated rate may not match the input rate of a VC at a NIC. Mechanisms to control the actual sources of traffic are necessary to avoid cell loss at the NIC. Further, the NIC needs to have a large number of buffers and typically needs to manage per-VC queues. Current implementations are possible since the number of VCs per NIC is small. When ATM is deployed in the backbone alone, then the cost increases in the edge routers/switches.

3. Typically, the number of end systems is one order of magnitude greater than the number of switches. This has a multiplier effect on the cost of an ATM network supporting ABR. The service can be made attractive only if:

   a) cheap ABR end-system implementations are available;

   b) there are mechanisms which carry the benifits of ABR (cell loss control, thoughput, controlled delay, fairness) to the applications (an application would not choose ABR if its performance degrades due to cell loss at the ABR end-system;

   c) an application programming interface (API) is available for end-to-end ATM implementations which maps applications to ABR;

   d) a larger class of applications (like variable quality voice, audio, video) can be scalably supported using the ABR service (and allow higher/costlier classes of service for applications willing to pay the price for the higher quality of service).

CHAPTER 11

SUMMARY AND FUTURE WORK

11.1 Summary of Contributions

This dissertation has examined the design of several traffic management mechanisms and methodologies for the ABR service in ATM networks. The ideas presented in this dissertation have significantly impacted the shape of the ATM Traffic Management 4.0 standard. This section summarizes the contributions of this work.

In **Chapter 1** we gave a specification of the control problem in ABR traffic management. We presented an open-loop equation in this chapter and presented the requirements for the closed loop solution in **Chapter 3**. The goals we seek to address include: efficiency (high throughput and low delay), fairness, steady state and transient performance, buffer requirements, robustness, implementation complexity and scalability. **Chapter 2** gives a tutorial introduction to the source, destination and switch rules as defined by the ATM Traffic Management 4.0 standard.

A large body of related work (ABR switch schemes) are surveyed in **Chapter 4. Chapters 5 and 6** describe the OSU, ERICA and ERICA+ schemes and related performance analyses.

The OSU scheme was one of the first explicit rate schemes designed for ABR. It exposed some of the pitfalls in using window-based control techniques in rate-based control. The key contributions of the algorithm are:

- Choice of congestion indicator (input rate i.e. aggregate demand, instead of queue length)
- Application of the congestion avoidance concept in rate-based control (use of the target utilization parameter).
- Use of the "overload factor" and "equal fairshare" metrics instead of simply the queue length.
- Small number of parameters
- Measurement of the number of active sources. In general, the scheme uses measurement instead of beleiving the declared values of metrics.
- O(1) time complexity.
- A proof that the fairness algorithm does achieve fairness.

The drawbacks of the scheme are its slow convergence in complex configurations, and the fact that it is incompatible with the final version of the standard (since it was developed at a time when the standards themselves were not finalized).

Three different options that further improve the performance over the basic scheme are also described. These allow the fairness to be achieved quickly, oscillations to be minimized, and feedback delay to be reduced.

The OSU scheme drawbacks were addressed in the ERICA schemes. The ERICA set of schemes use an optimistic approach to provide good steady state as well as

good transient performance. Since real networks are in a transient state most of the time (sources are rarely infinite and ABR capacity varies), the transient performance of a scheme deployed under these conditions is of importance.

The difference in approach between the OSU scheme and ERICA is that while the former attempts to achieve efficiency and fairness one after another, the latter attempts to move towards efficiency and fairness at the same time. It uses an aggressive core algorithm - the maximum of an "efficiency term" (based on the overload factor and the source's current rate) and a "fairness term" (based on the available capacity and the number of active sources). The ERICA schemes still rely on measurement of load, capacity and the number of active sources to calculate the rates. The ERICA+ scheme attempts to achieve an operating point of 100% utilization and a target queueing delay. The use of the queueing delay metric allows the scheme to be robust to errors in measurement and feedback delays (which manifest as queues at the switch). Simulation results with different configurations and traffic patterns have also been presented.

Chapter 7 examines the design of source rules in the ATM Traffic Management framework, i.e., how "open-loop" control complements the "closed-loop" feedback system. This dissertation work has helped develop a number of the rules in the international standard. However, two of these issues are investigated in depth in this chapter.

The first issue is the design of "Use-it-or-Lose-it" policies. These policies take away a source's assigned rate if the source does not use it. The choice of the policy has a significant impact on ABR service capabilities, affecting the performance of bursty (on-off) sources and sources bottlenecked below their network-assigned rate. We

present a survey of the proposed approaches including our approach. The approaches can broadly be classified as source-based approaches (where the source end-system implements the policy) and switch-based policies (where the switch may implement proprietary measures to address the problem). After a long debate, the ATM Forum decided not to standardize an elaborate source-based UILI policy. A simple timeout is mandated for the source, where sources keep their rate allocations until a timeout (parameter ATDF, of the order of 500 ms) expires. We present a detailed study of the various alternatives in this chapter.

The second issue is the efficient support of low-rate sources. We study three mechanisms - tuning the Trm parameter setting, the TCR parameter which controls the rate of out-of-rate RM cells, and a source rescheduling policy which may trigger when the source receives a rate increase indication. The tradeoffs in these mechanisms are examined in this chapter.

Chapter 8 deals with issues in supporting internet applications like file transfer and world wide web (which run over the TCP/IP protocol) over ATM ABR, with different models of higher priority VBR background traffic in the background. We show that a well-designed ABR system can scalably support persistant TCP applications like ftp as well as bursty TCP applications like WWW clients and servers. We study the TCP dynamics and show that when ftp applications using TCP run over ABR, the switch algorithm can control the TCP sources given sufficient amount of buffering. Once the control has been established, given no changes in traffic behavior, TCP can achieve maximum throughput and zero cell loss. The buffer requirements do not depend upon the number of TCP sources - only on parameters like the switch algorithm parameters and round trip time. We verify that this requirement holds despite highly

variant background traffic conditions, and for LAN, WAN and satellite configurations. To introduce highly variant conditions, we use conventional ON-OFF models and also propose new models of MPEG-2 transport streams (resulting in long-range dependent traffic) multiplexed over VBR. We observe that the ABR control system only pushes the queues to the edge of the network - where an edge router has to again handle the issue of large TCP queues.

We note that the system can theoritically be loaded with unbounded queues when bursty traffic like WWW is used. However, under practical conditions, the average load when a large number of WWW exist increases more smoothly than expected. Since the ABR switch scheme reacts to load and can tolerate variation in load and capacity, we see that queues are controlled and high throughput is attained even under such conditions.

In **Chapter 9** we look at the switch design issues for a specific ABR framework option called the "Virtual Source/Virtual Destination" option. In this option, the switch splits the network into two segments and shortens the feedback loop for both segments. We show that this option has the potential to increase the performance of the network, but the implementation can be complex and has to be carefully done. In our study of multiple implementation options of this feature, we found that only a very few performed well, and we identify the properties of the best option. This chapter is followed up by **Chapter 10** where we briefly address certain implementation issues.

11.2 Future Work

At the time of this writing, the ABR service is being actively implemented and is currently facing interesting cost-performance tradeoff questions. Field trials and

interoperability tests are required to ensure that the implementations conform to the specifications and deliver the promised performance. Another step to make the service more attractive in the cost-performance tradeoff is to demonstrate that a large class of applications can be made to run over the service. Currently, ABR is promising for two key internet applications: file transfer and the world wide web. It is interesting to see if ABR can support variable quality voice and video. ABR multicast is also another pre-requisite for supporting a wider class of applications

Another issue with ATM backbone scenarios is that ABR provides control only upto the edge of the ATM network. It is possible that the edge router can use the ABR rate feedback information to pace TCP traffic. This will carry the benifits of ABR to applications (i.e., end-to-end).

The proliferation of high-speed networking will increase the demand for high quality of service (QoS) on access network technologies like wireless and DSL (ADSL, 56 kbps modems etc). Such technologies are characterized by low bit rates and high error rates. The implication of the mapping of ATM on such technologies is that traffic management has to deal with the effect of uniform errors (due to the underlying technology) as well as burst errors (due to congestion). Another scenario is that of satellite networks where the delay is large, the bit rates are smaller, and the satellite technology imposes rigid design constraints. Special schemes are required to handle such scenarios correctly.

APPENDIX A

SOURCE, DESTINATION AND SWITCH RULES

This appendix provides the precise source and destination behavior verbatim from the ATM Forum's Traffic Management 4.0 specification [35]. All table, section, and other references in this appendix refer to those in the TM specification.

5.10.4 Source Behavior

The following items define the source behavior for CLP=0 and CLP=1 cell streams of a connection. By convention, the CLP=0 stream is referred to as in-rate, and the CLP=1 stream is referred to as out-of-rate. Data cells shall not be sent with CLP=1.

1. The value of ACR shall never exceed PCR, nor shall it ever be less than MCR. The source shall never send in-rate cells at a rate exceeding ACR. The source may always send in-rate cells at a rate less than or equal to ACR.

2. Before a source sends the first cell after connection setup, it shall set ACR to at most ICR. The first in-rate cell shall be a forward RM-cell.

3. After the first in-rate forward RM-cell, in-rate cells shall be sent in the following order:

   a) The next in-rate cell shall be a forward RM cell if and only if, since the last in-rate forward RM-cell was sent, either:

i) at least Mrm in-rate cells have been sent and at least Trm time has elapsed, or

ii) Nrm -1 in-rate cells have been sent.

b) The next in-rate cell shall be a backward RM-cell if condition (a) above is not met, if a backward RM cell is waiting for transmission, and if either:

i) no in-rate backward RM-cell has been sent since the last in-rate forward RM-cell, or

ii) no data cell is waiting for transmission.

c) The next in-rate cell sent shall be a data cell if neither condition (a) nor condition (b) is met, and if a data cell is waiting for transmission.

4. Cells sent in accordance with source behaviors #1,#2, and #3 shall have CLP=0.

5. Before sending a forward in-rate RM cell, if ACR > ICR and the time T that has elapsed since the last in-rate forward RM-cell was sent is greater than ADTF, then ACR shall be reduced to ICR.

6. Before sending an in-rate forward RM cell, and following behavior #5 above, if at least CRM in-rate forward RM-cells have been sent since the last backward RM-cell with BN=0 was received, then ACR shall be reduced by at least ACR $\times$ CDF, unless that reduction would result in a rate below MCR, in which case ACR shall be set to MCR.

7. After following behaviors #5 and #6 above, the ACR value shall be placed in the CCR field of the outgoing forward RM-cell, but only in-rate cells sent after the outgoing forward RM-cell need to follow the new rate.

8. When a backward RM-cell (in-rate or out-of-rate) is received with CI=1, then ACR shall be reduced by at least ACR $\times$ RDF, unless that reduction would result in a rate below MCR, in which case ACR shall be set to MCR. If the backward RM-cell has both CI=0 and NI=0, then the ACR may be increased by no more than RIF $\times$ PCR, to a rate not greater than PCR. If the backward RM-cell has NI=1, the ACR shall not be increased.

9. When a backward RM-cell (in-rate or out-of-rate) is received, and after ACR is adjusted according to source behavior #8, ACR is set to at most the minimum of ACR as computed in source behavior #8, and the ER field, but no lower than MCR.

10. When generating a forward RM-cell, the source shall assign values to the various RM-cell fields as specified for source-generated cells in Table 5-4.

11. Forward RM-cells may be sent out-of-rate (i.e., not conforming to the current ACR). Out-of-rate forward RM-cells shall not be sent at a rate greater than TCR.

12. A source shall reset EFCI on every data cell it sends.

13. The source may implement a use-it-or-lose-it policy to reduce its ACR to a value which approximated the actual cell transmission rate. Use-it-or-lose-it policies are discussed in Appendix I.8.

Notes:

1. In-rate forward and backward RM-cells are included in the source rate allocated to a connection.

2. The source is responsible for handling congestion within its scheduler in a fair manner. This congestion occurs when the sum of the rates to be scheduled exceeds the output rate of the scheduler. The method for handling local congestion is implementation specific.

5.10.5 Destination Behavior

The following items define the destination behavior for CLP=0 and CLP=1 cell streams of a connection. By convention, the CLP=0 stream is referred to as in-rate, and the CLP=1 stream is referred to as out-of-rate.

1. When a data cell is received, its EFCI indicator is saved as the EFCI state of the connection.

2. On receiving a forward RM-cell, the destination shall turn around the cell to return to the source. The DIR bit in the RM-cell shall be changed from "forward" to "backward," BN shall be set to zero, and CCR, MCR, ER, CI, and NI fields in the RM-cell shall be unchanged except:

   a) If the saved EFCI state is set, then the destination shall set CI=1 in the RM cell, and the saved EFCI state shall be reset. It is preferred that this step is performed as close to the transmission time as possible;

   b) The destination (having internal congestion) may reduce ER to whatever rate it can support and/or set CI=1 or NI=1. A destination shall either

set the QL and SN fields to zero, preserve these fields, or set them in accordance with ITU-T Recommendation I.371-draft. The octets defined in Table 5-4 as reserved may be set to 6A (hexadecimal) or left unchanged. The bits defined as reserved in Table 5-4 for octet 7 may be set to zero or left unchanged. The remaining fields shall be set in accordance with Section 5.10.3.1 (Note that this does not preclude looping fields back from the received RM cell).

3. If a forward RM-cell is received by the destination while another turned-around RM-cell (on the same connection) is scheduled for in-rate transmission:

   a) It is recommended that the contents of the old cell are overwritten by the contents of the new cell;

   b) It is recommended that the old cell (after possibly having been overwritten) shall be sent out-of-rate; alternatively the old cell may be discarded or remain scheduled for in-rate transmission;

   c) It is required that the new cell be scheduled for in-rate transmission.

4. Regardless of the alternatives chosen in destination behavior #3, the contents of the older cell shall not be transmitted after the contents of a newer cell have been transmitted.

5. A destination may generate a backward RM-cell without having received a forward RM-cell. The rate of the backward RM-cells (including both in-rate and out-of-rate) shall be limited to 10 cells/second, per connection. When a destination generated an RM-cell, it shall set either CI=1 or NI=1, shall set set

BN=1, and shall set the direction to backward. The destination shall assign values to the various RM-cell fields as specified for destination generated cells in Table 5-4.

6. When a forward RM-cell with CLP=1 is turned around it may be sent in-rate (with CLP=0) or out-of-rate (with CLP=1)

Notes-

1. "Turn around" designates a destination process of transmitting a backward RM-cell in response to having received a forward RM-cell.

2. It is recommended to turn around as many RM-cells as possible to minimize turn-around delay, first by using in-rate opportunities and then by using out-of-rate opportunities as available. Issues regarding turning RM-cells around are discussed in Appendix I.7.

5.10.6 Switch Behavior

The following items define the switch behavior for CLP=0 and CLP=1 cell streams of a connection. By convention, the CLP=0 stream is referred to as in-rate, and the CLP=1 stream is referred to as out-of-rate. Data cells shall not be sent with CLP=1.

1. A switch shall implement at least one of the following methods to control congestion at queueing points:

   a) *EFCI marking:* The switch may set the EFCI state in the data cell headers;

   b) *Relative Rate Marking:* The switch may set CI=1 or NI=1 in forward and/or backward RM-cells; item[c)] *Explicit Rate Marking:* The switch may reduce the ER field of forward and/or backward RM-cells (Explicit Rate Marking);

d) *VS/VD Control:* The switch may segment the ABR control loop using a virtual source and destination.

2. A switch may generate a backward RM-cell. The rate of these backward RM-cells (including both in-rate and out-of-rate) shall be limited to 10 cells/second, per connections. When a switch generates an RM-cell it shall set either CI=1 or NI=1, shall set BN=1, and shall set the direction to backward. The switch shall assign values to the various RM-cell fields as specified for switch-generated cells in Table 5-4.

3. RM-cells may be transmitted out of sequence with respect to data cells. Sequence integrity within the RM-cell stream must be maintained.

4. For RM-cells that transit a switch (i.e., are received and then forwarded), the values of the various fields before the CRC-10 shall be unchanged except:

   a) CI,NI and ER may be modified as noted in #1 above

   a) RA, QL and SN shall be set in accordance with ITU-T Recommendation I.371-draft

   MCR may be corrected to the connection's MCR if the incoming MCR value is incorrect.

5. The switch may implement a use-it-or-lose it policy to reduce an ACR to a value which approximates the actual cell transmission rate from the source. Use-it-or-lose-it policies are discussed in Appendix I.8.

Notes-

1. A switch queueing point is a point of resource contention where cells may be potentially delayed or lost. A switch may contain multiple queueing points.

2. Some example switch mechanisms are presented in Appendix I.5.

3. The implications of combinations of the above methods is beyond the scope of this specification.

5.10.7 Virtual Source and Virtual Destination Behavior

VS/VD behavior divides an ABR connection into two or more separately controlled ABR segments. The coupling between adjacent ABR control segments associated with an ABR connection is implementation specific.

The following applies to VS/VD behavior:

1. Each ABR control segment, except the first, is sources by a virtual source. A virtual source assumes the behavior of an ABR source end point. Backward RM-cells received by a virtual source are removed from the connection.

2. Each ABR control segment, except the last, is terminated by a virtual destination. A virtual destination assumes the behavior of an ABR destination end point. Forward RM-cells received by a virtual destination shall be turned around as defined in destination behavior #2, and shall not be forwarded to the next segment of the connection.

3. The coupling between two adjacent ABR control segments associated with an ABR connection is implementation specific.

4. MCR shall be conveyed across VS/VD boundaries.

5. Setting of other parameters at VS/VD is network specific

APPENDIX B

THE OSU SCHEME: PSEUDO CODE

B.1 The Source Algorithm

There are four events that can happen at the source adapter or Network Interface Card (NIC). These events and the action to be taken on these events are described below.

1. Initialization:

   TCR ←Initial Cell Rate;

   Averaging_Interval ←Some initial value;

   IF (BECN_Option) THEN Time_Already_Acted ←0;

2. A data cell or cell burst is received from the host.

   Enqueue the cell(s) in the output queue.

3. The inter-cell transmission timer expires.

   IF Output_Queue NOT Empty THEN dequeue the first cell and transmit;

   Increment Transmitted_Cell_Count;

   Restart Inter_Cell_Transmission_Timer;

4. The averaging interval timer expires.

   Offered_Cell_Rate ←Transmitted_Cell_Count/Averaging_Interval;

   Transmitted_Cell_Count ←0;

   Create a control cell;

   OCR_In_Cell ←Offered_Cell_Rate ;

   TCR_In_Cell ←max{TCR, OCR} ;

   Load_Adjustment_Factor ←0;

   IF (BECN_Option) THEN Time_Stamp_in_Cell ←Current Time;

   Transmit the control cell;

   Restart Averaging_Interval_Timer;

5. A control cell returned from the destination is received.

   IF ((BECN_Option AND Time_Already_Acted < Time_Stamp_In_Cell) OR

   (NOT BECN_Option))

   THEN BEGIN

   New_TCR ←TCR_In_Cell/Load_Adjustment_Factor_In_Cell;

   IF Load_Adjustment_Factor_In_Cell ≥ 1

   THEN IF New_TCR < TCR

   THEN BEGIN

   TCR ←New_TCR ;

   IF(BECN_Option)

   THEN Time_Already_Acted ←Time_Stamp_In_Cell;

   END

   ELSE IF Load_Adjustment_Factor_In_Cell < 1

THEN IF New_TCR > TCR THEN TCR ←New_TCR ;

Inter_Cell_Transmission_Time ←1/TCR;

END; (* of FECN Cell processing *)

Averaging_Interval ←Averaging_Interval_In_Cell;

6. A BECN control cell is received from some switch.

IF BECN_Option

THEN IF Time_Already_Acted <

Time_Stamp_In_Cell

THEN IF Load_Adjustment_Factor_In_Cell ≥ 1

THEN BEGIN

New_TCR ←

TCR_In_Cell/Load_Adjustment_Factor_In_Cell;

IF New_TCR < TCR

THEN BEGIN

TCR ←New_TCR;

Inter_Cell_Transmission_Time ←1/TCR;

Time_Already_Acted ←Time_Stamp_In_Cell;

END;

END;

B.2 The Switch Algorithm

The events at the switch and the actions to be taken on these events are as follows:

1. Initialization:

   Target_Cell_Rate ←Link_Bandwidth × Target_Utilization / Cell_Size ;

   Target_Cell_Count ←Target_Cell_Rate×Averaging_Interval;

   Received_Cell_Count ←0;

   Clear VC_Seen_Bit for all VCs;

   IF (Basic_Fairness_Option OR Aggressive_Fairness_Option)

   THEN BEGIN

      Upper_Load_Bound ←1 + Half_Width_Of_TUB;

      Lower_Load_Bound ←1 - Half_Width_Of_TUB;

   END;

2. A data cell is received.

      Increment Received_Cell_Count;

      Mark VC_Seen_Bit for the VC in the Cell;

3. The averaging interval timer expires.

      Num_Active_VCs ←max{$\sum$ VC_Seen_Bit, 1};

      Fair_Share_Rate ←Target_Cell_Rate/Num_Active_VCs;

      Load_Level ←Received_Cell_Count/Target_Cell_Count;

      Reset all VC_Seen_Bits;

      Received_Cell_Count ←0;

      Restart Averaging_Interval_Timer;

4. A control cell is received.

IF (Basic_Fairness_Option)

THEN IF (Load_Level $\geq$ Lower_Load_Bound)

and (Load_Level $\leq$ Upper_Load_Bound)

THEN BEGIN

IF OCR_In_CELL > Fair_Share_Rate

THEN Load_Adjustment_Decision $\leftarrow$Load_Level/Lower_Load_Bound

ELSE Load_Adjustment_Decision $\leftarrow$Load_Level/Upper_Load_Bound

END (*IF *)

ELSE Load_Adjustment_Decision $\leftarrow$Load_Level;

IF (Aggressive_Fairness_Option)

THEN BEGIN

Load_Adjustment_Decision $\leftarrow$1;

IF (Load_Level < Lower_Load_Bound)

THEN IF ((OCR_In_Cell < Fair_Share_Rate$\times$Load_Level) OR

(Num_VC_Active =1))

THEN Load_Adjustment_Decision $\leftarrow$Load_Level

ELSE IF (OCR_In_Cell < Target_Cell_Rate$\times$Load_Level)

THEN Load_Adjustment_Decision $\leftarrow$Load_Level + (1-

Load_Level)$\times$(OCR_In_Cell/(Load_level$\times$

Fair_Share)-1)/(Num_VC_Active-1)

ELSE Load_Adjustment_Decision $\leftarrow$1

ELSE IF Load_Level $\geq$ Upper_Load_Bound

```
                    THEN IF (OCR_In_Cell ≤ Fair_Share_Rate AND
                        Num_Active_VCs ≠ 1)
                        THEN Load_Adjustment_Decision ←1
                        ELSE IF (OCR_In_Cell <
                          Fair_Share_Rate × Load_Level)
                            THEN Load_Adjustment_Decision ←max{1,
                                OCR_In_Cell/Fair_Share_Rate}
                            ELSE IF (OCR_In_Cell ≤ Target_Cell_Rate)
                                THEN Load_Adjustment_Decision ←
                                  Load_Level
                                ELSE Load_Adjustment_Decision ←
                                    OCR_In_Cell ×
                                      Load_Level/Target_Cell_Rate;
    END (* of Aggressive Fairness Option *)

IF (Precise_Fairshare_Computation_Option)
BEGIN
    OCR_Of_VC_In_Table ←OCR_In_Cell;
    Fair_Share_Rate ←Target_Cell_Rate/Num_VC_Active;
    REPEAT
        Num_VC_Underloading ←0 ;
        Sum_OCR_Underloading ←0 ;
        FOR each VC seen in the last interval DO
        IF (OCR_In_Cell < Fair_Share_Rate)
```

```
            THEN BEGIN
                Increment Num_VC_Underloading ;
                Sum_OCR_Underloading ←
                 Sum_OCR_Underloading + OCR_Of_VC
            END (* IF *)
            Fair_Share_Rate ←(Target_Cell_Rate - SUM_OCR_Underloading)
                /max{1, (Num_VC_Active - Num_VC_Underloading )}
        UNTIL Fair_Share_Rate does not change (* Maximum of 2 iterations *);
Load_Adjustment_Decision ←OCR_In_Cell/Fair_Share_Rate;
END; (* Precise Fairness Computation Option *)

IF (Load_Adjustment_Decision > Load_Adjustment_Factor_In_Cell)
THEN BEGIN
    Load_Adjustment_Factor_In_Cell ←Load_Adjustment_Decision;
   IF BECN_Option and Load_Adjustment_Decision > 1
    THEN SEND_A_COPY_OF_CONTROL_CELL_BACK_TO_SOURCE ;
END (* IF *)
```

APPENDIX C

ERICA SWITCH ALGORITHM: DETAILED DESCRIPTION

C.1 Variables and Flow charts

Notes:

- All rates are in the units of cells/s
- The following pseudo-code assumes a simple fixed-time averaging interval. Extension to a cells and time averaging interval is trivial.

We use a combination of flowcharts and pseudo-code to describe the ERICA algorithm. The following names are used to identify the flow charts:

Flow Chart 1: Flow Chart of the Basic ERICA Algorithm. Figure C.1.

Flow Chart 2: Flow Chart for Achieving Max-Min Fairness. Figure C.2.

Flow Chart 3: Flow Chart for Bi-Directional Counting. Figure C.3.

Flow Chart 4: Flow Chart of averaging number of active sources (part 1 of 2). Figure C.4.

Flow Chart 5: Flow Chart of averaging number of active sources (part 2 of 2). Figure C.5.

Flow Chart 6: Flow Chart of averaging load factor (method 1). Figure C.6.

Flow Chart 7: Flow Chart of averaging load factor (method 2). Figure C.7.

C.2 Pseudocode

Initialization:

(* ABR Capacity and Target Utilization *)

IF (Queue_Control_Option) THEN

Target_Utilization ←1

END (* IF *)

ABR_Capacity_In_cps ←Target_Utilization × Link_Bandwidth −

VBR_and_CBR_Capacity

(* Count of Number of VCs, Cells *)

FOR ALL VCs DO

Contribution[VC] ←0

Seen_VC_In_This_Interval[VC] ←0

Seen_BRM_Cell_In_This_Interval[VC] ←0

END (* FOR *)

ABR_Cell_Count ←ABR_Capacity_In_cps × Averaging_Interval

Number_Active_VCs_In_This_Interval ←Total Number of Setup VCs

```
Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_This_Interval

(* Fairshare and Load Factor variables *)
Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
Max_Alloc_Previous ←0
Max_Alloc_Current ←Fair_Share
Load_Factor ←ABR_Capacity_In_cps/ FairShare

(* Per VC CCR Option Variables *)
IF (Per_VC_CCR_Option) THEN
   FOR ALL VCs DO
      Number_Of_Cells[VC] ←0
   END (* FOR *)
END (* IF *)
```

A cell of "VC" is received in the forward direction:

```
IF (Averaging_VCs_Option) THEN
   IF (Contribution[VC] < 1) THEN (* VC inactive in current interval *)
      Number_Active_VCs_In_This_Interval ←
            Number_Active_VCs_In_This_Interval − Contribution[VC] + 1
      IF ((Immediate_Fairshare_Update_Option) AND
      (Contribution[VC] < Decay_Factor)) THEN
         Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_Last_Interval
```

```
        – (Contribution[VC] / Decay_Factor) + 1
      Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
    END (* IF *)
    Contribution[VC] ←1
  END (* IF *)
ELSE
  IF (NOT(Seen_VC_In_This_Interval[VC])) THEN
    Seen_VC_In_This_Interval[VC] ←1
  END (* IF *)
  IF ((Immediate_Fair_Share_Option) AND (NOT(Seen_VC_In_Last_Interval[VC])))
THEN
    Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_Last_Interval +
1
    Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
    Seen_VC_In_Last_Interval[VC] ←1
  END (* IF *)
END (* IF *)
ABR_Cell_Count ←ABR_Cell_Count + 1
IF (Per_VC_CCR_Option) THEN
  Number_Of_Cells[VC] ←Number_Of_Cells[VC] + 1
END (* IF *)
```

Averaging interval timer expires:

```
IF (NOT(Averaging_VCs_Option)) THEN
    Number_Active_VCs_In_Last_Interval ←
    Max (∑ Seen_VC_In_This_Interval, 1)
    Number_Active_VCs_In_This_Interval ←0
    FOR ALL VCs DO
      Seen_VC_In_Last_Interval[VC] ←Seen_VC_In_This_Interval[VC]
    END (* FOR *)
ELSE
    Number_Active_VCs_In_Last_Interval ←
     Max(Number_Active_VCs_In_This_Interval , 1)
    Number_Active_VCs_In_This_Interval ←0
    FOR ALL VCs DO
      Contribution[VC] ←Contribution[VC] × Decay_Factor
      Number_Active_VCs_In_This_Interval ←Number_Active_VCs_In_This_Interval +
Contribution[VC]
    END (* FOR *)
END (* IF *)

IF (Exponential_Averaging_Of_Load_Method_2_Option) THEN
    ABR_Capacity_In_Cells ←
    Max(Target_Utilization × Link_Bandwidth × Averaging_Interval)
      − VBR_and_CBR_Cell_Count, 0)
    Avg_ABR_Capacity_In_Cells ←
```

```
    (1−α) × Avg_ABR_Capacity_In_Cells +
    α× ABR_Capacity_In_Cells
  Avg_Averaging_Interval ←
    (1−α)× Avg_Averaging_Interval + α×Averaging_Interval
  Avg_ABR_Cell_Count ←(1−α)×Avg_ABR_Cell_Count + α×ABR_Cell_Count
  ABR_Input_Rate ←Avg_ABR_Cell_Count / Avg_Averaging_Interval
  ABR_Capacity_In_cps ←Avg_ABR_Capacity_In_Cells / Avg_Averaging_Interval
ELSE
  VBR_and_CBR_Cell_Rate ←VBR_and_CBR_Cell_Count / Averaging_Interval
  ABR_Capacity_In_cps ←
    Max(Target_Utilization×Link_Bandwidth − VBR_and_CBR_Cell_Rate, 0)
  ABR_Input_Rate ←ABR_Cell_Count / Averaging_Interval
END (* IF *)

IF (Queue_Control_Option) THEN
Target_Queue_Length ←Target_Time_To_Empty_Queue × ABR_Capacity_In_cps
Queue_Control_Factor ←Fn(Current_Queue_Length)
ABR_Capacity_In_cps ←Queue_Control_Factor × ABR_Capacity_In_cps
END (* IF *)

IF (Exponential_Averaging_Of_Load_Method_1_Option) THEN
  IF (ABR_Capacity_In_cps ≤ 0) THEN
    Load_Factor ←Infinity
  ELSE
```

```
      IF (Load_Factor = Infinity) THEN
        Load_Factor ←ABR_Input_Rate / ABR_Capacity_In_cps
      ELSE
        Load_Factor ←(1−α) × Load_Factor +
          α × ABR_Input_Rate / ABR_Capacity_In_cps
      END (* IF *)
    END (* IF *)
ELSE IF (Exponential_Averaging_Of_Load_Method_2_Option) THEN
    IF (ABR_Capacity_In_cps ≤ 0) THEN
      Load_Factor ←Infinity
    ELSE
      Load_Factor ←ABR_Input_Rate / ABR_Capacity_In_cps
    END (* IF *)
ELSE (* No exponential averaging *)
    IF (ABR_Capacity_In_cps ≤ 0) THEN
      Load_Factor ←Infinity
    ELSE
      Load_Factor ←ABR_Input_Rate / ABR_Capacity_In_cps
    END (* IF *)
END (* IF *)
Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
Max_Alloc_Previous ←Max_Alloc_Current
Max_Alloc_Current ←Fair_Share
FOR ALL VCs DO
```

```
    Seen_VC_In_This_Interval[VC] ←0
    Seen_BRM_Cell_In_This_Interval[VC] ←0
END (* FOR *)
ABR_Cell_Count ←0
IF (Per_VC_CCR_Option) THEN
    FOR ALL VCs DO
      CCR[VC] ←Number_Of_Cells[VC]/Averaging_Interval
      Number_Of_Cells[VC] ←0
    END (* FOR *)
END (* IF *)
VBR_and_CBR_Cell_Count ←0
Restart Averaging_Interval Timer
```

A Forward RM (FRM) cell of "VC" is received:

```
    IF (NOT(Per_VC_CCR_Option)) THEN
    CCR[VC] ←CCR_In_FRM_Cell
END (* IF *)
```

A Backward RM (BRM) cell of "VC" is received:

```
    IF (Averaging_VCs_Option) THEN
    IF (Contribution[VC] < 1) THEN (* VC inactive in current interval *)
      Number_Active_VCs_In_This_Interval ←
        Number_Active_VCs_In_This_Interval − Contribution[VC] + 1
      IF ((Immediate_Fairshare_Update_Option) AND
       (Contribution[VC] < Decay_Factor)) THEN
        Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_Last_Interval
```

```
            – (Contribution[VC] / Decay_Factor) + 1
          Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
      END (* IF (Immediate ...) *)
      Contribution[VC] ←1
   END (* IF (Contribution ... ) *)
ELSE (* NOT (Averaging_VCs_Option) *)
   IF (NOT(Seen_VC_In_This_Interval[VC])) THEN
      Seen_VC_In_This_Interval[VC] ←1
   END (* IF *)
   IF ((Immediate_Fair_Share_Option) AND (NOT(Seen_VC_In_Last_Interval[VC])))
THEN
      Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_Last_Interval +
1
      Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
      Seen_VC_In_Last_Interval[VC] ←1
   END (* IF ((Immediate ..)) *)
END (* IF-THEN-ELSE (Averaging_VCs_Option) *)

IF (Seen_BRM_Cell_In_This_Interval[VC]) THEN
   ER_Calculated ←Last_Allocated_ER[VC]
ELSE
   VC_Share[VC] ←CCR[VC] / Load_Factor
   (* Max-Min Fairness Algorithm *)
   IF (Load_Factor > 1 + δ) THEN
```

ER_Calculated ←Max (Fair_Share, VC_Share)

ELSE

ER_Calculated ←Max (Fair_Share, VC_Share, Max_Alloc_Previous)

END (* IF *)

Max_Alloc_Current ←Max (Max_Alloc_Current, ER_Calculated)

(* Avoid Unnecessary Transient Overloads *)

IF ((CCR[VC] < Fair_Share) AND (ER_Calculated ≥ Fair_Share)) THEN

ER_Calculated ←Fair_Share

(* Optionally Disable Feedback To This VC For An Averaging Interval *)

END (* IF *)

ER_Calculated ←Min(ER_Calculated, ABR_Capacity_In_cps)

(* Ensure One Feedback Per Switch Averaging Interval *)

Last_Allocated_ER[VC] ←ER_Calculated

Seen_BRM_Cell_In_This_Interval[VC] ←1

END (* IF *)

(* Give Feedback In BRM Cell *)

ER_In_BRM_Cell ←Min (ER_in_BRM_Cell, ER_Calculated)

At each cell slot time schedule cell from a service class using a scheduling policy

Name	Explanation	Flow Chart (FC) or Figure
ABR_Cell_Count	Number of ABR input cells in the current interval	FCs 1 and 7 (step 2)
Contribution[VC]	Contribution of the VC towards the count of the number of active sources	FCs 4 and 5
Seen_VC_In_ This_Interval[VC]	A bit which is set when a VC is seen in the current (last) interval	FCs 1,3 and 5
Number_Of_Cells[VC]	Used in Per VC CCR option to count number of cells from each VC in the current interval	
Max_Alloc_Previous	Max rate allocation in previous interval	FC 2
Max_Alloc_Current	Max rate allocation in current interval	FC 2
Seen_BRM_Cell_In_ This_Interval[VC]	A BRM from the source has been seen (and feedback given) in this interval. Do not give new feedback	Figure 6.2
Last_Allocated_ER	Unique ER feedback to the source in the current interval	Figure 6.2
Decay_Factor	Factor Used in Averaging the Number of Active Sources $0 < \text{Decay\_Factor} \leq 1$	FCs 4 and 5

Table C.1: Explanation of some of the ERICA Pseudocode variables

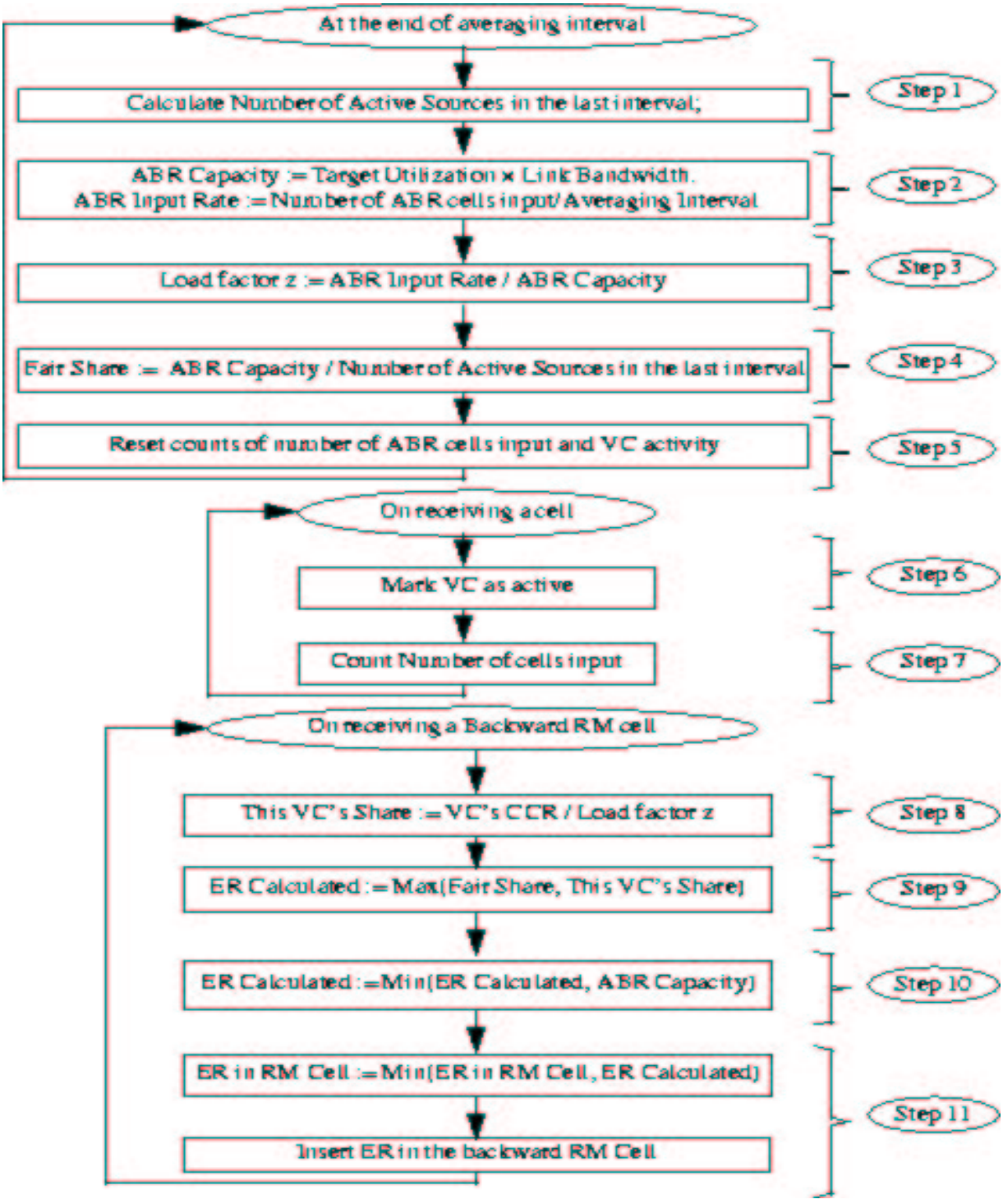


Figure C.1: Flow Chart of the Basic ERICA Algorithm

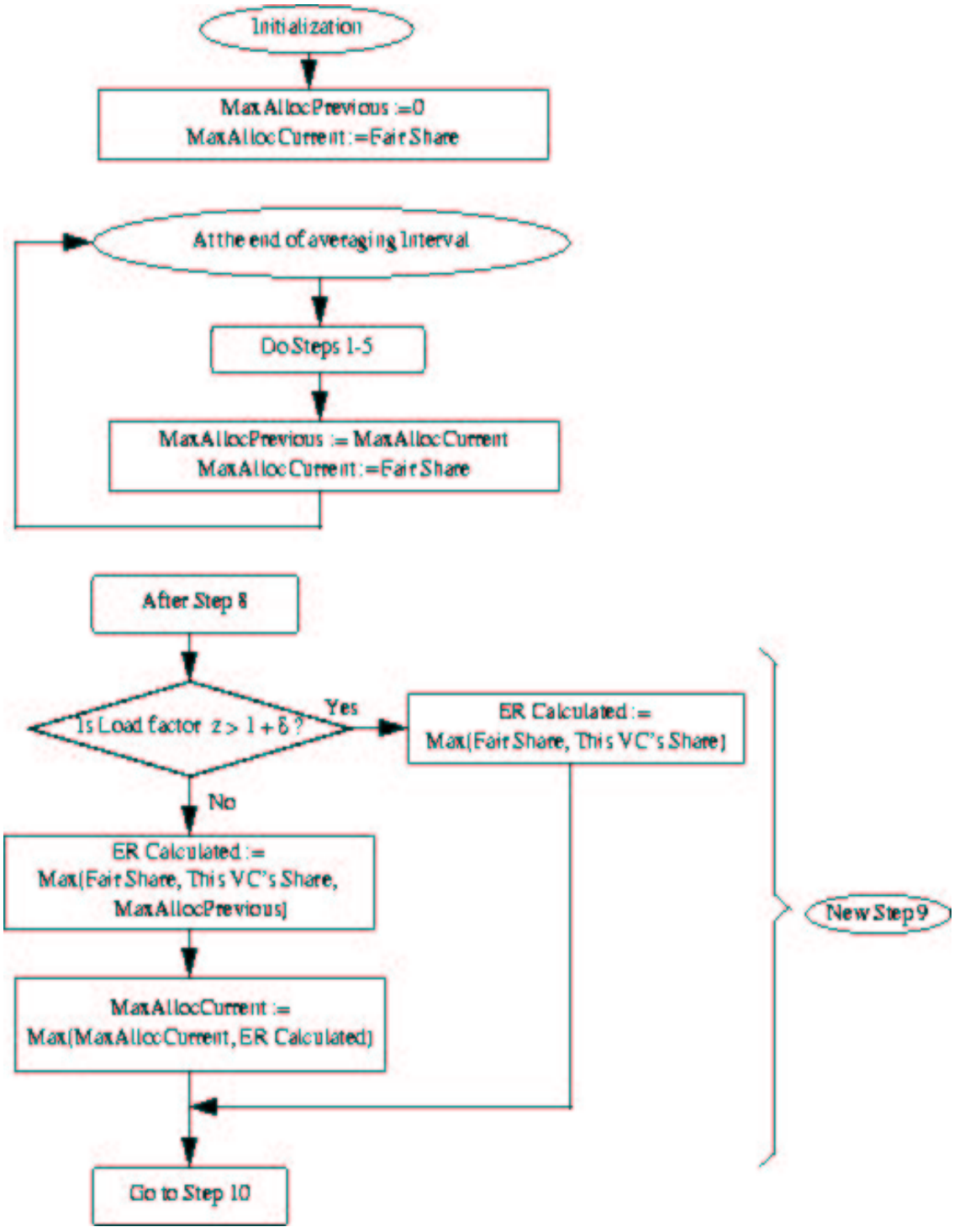


Figure C.2: Flow Chart for Achieving Max-Min Fairness

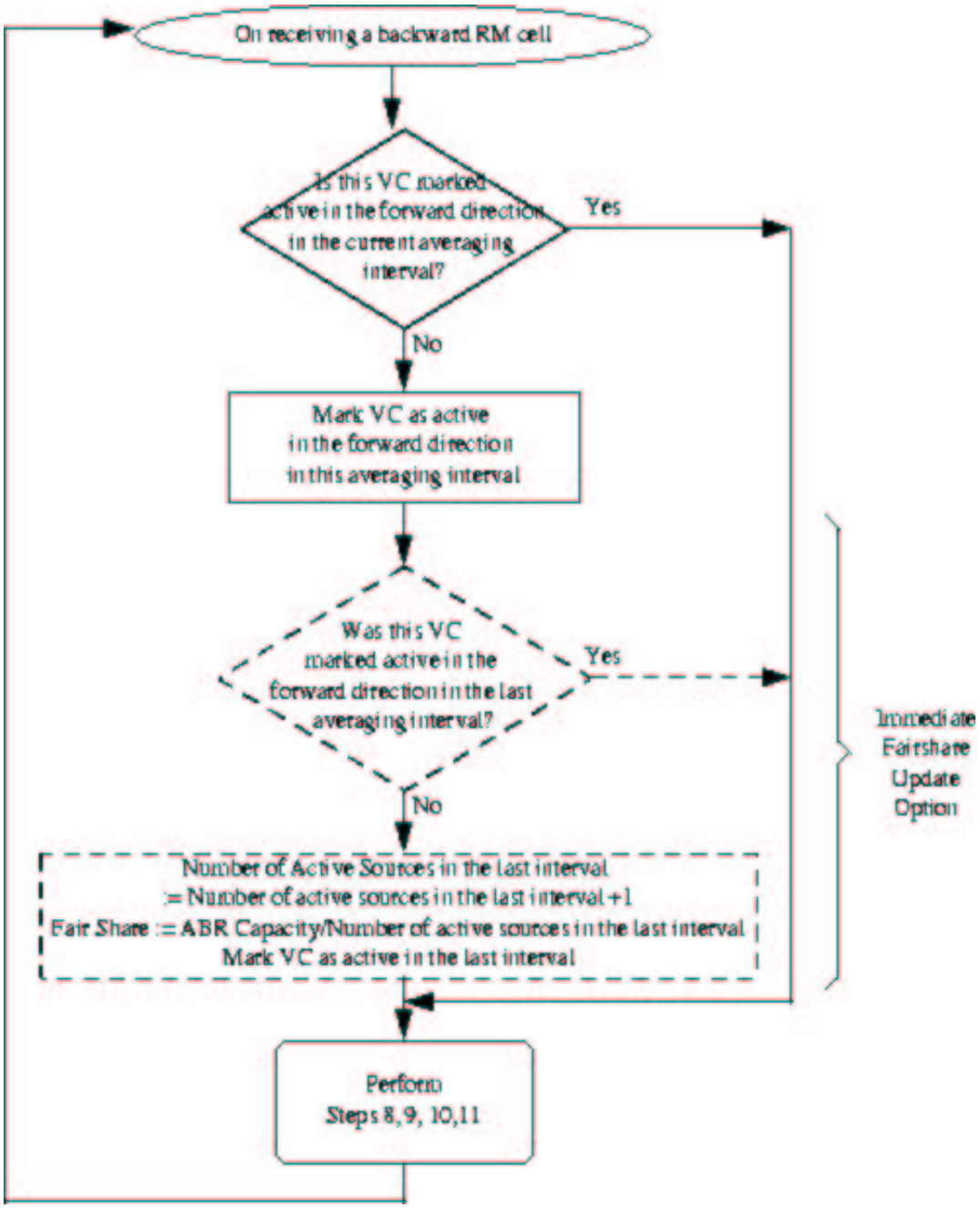


Figure C.3: Flow Chart of Bi-Directional Counting

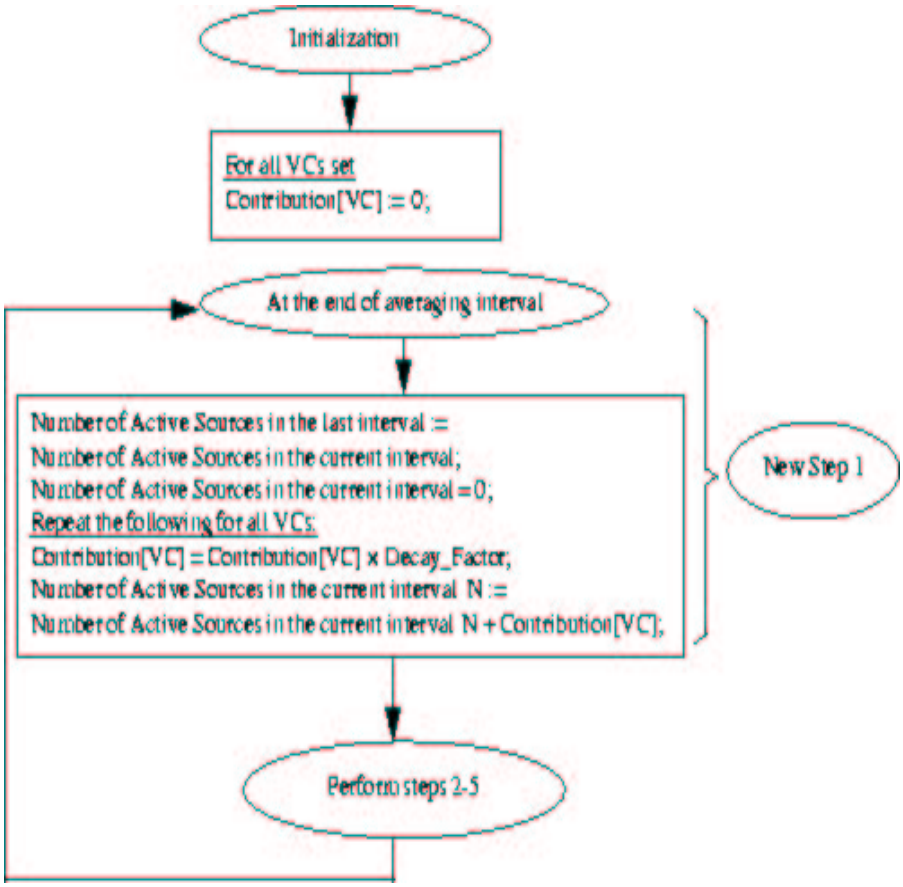


Figure C.4: Flow Chart of averaging number of active sources (part 1 of 2)

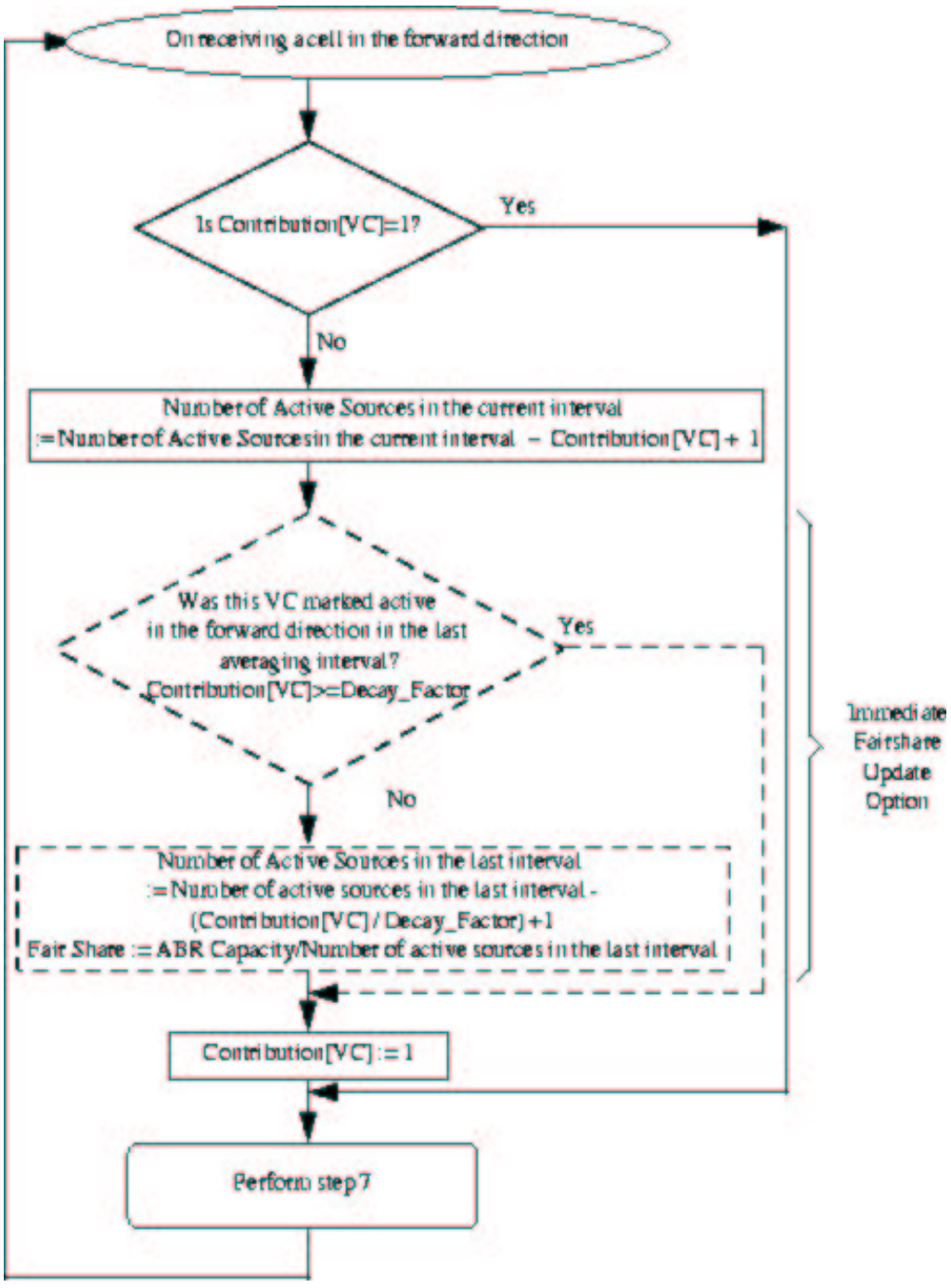


Figure C.5: Flow Chart of averaging number of active sources (part 2 of 2)

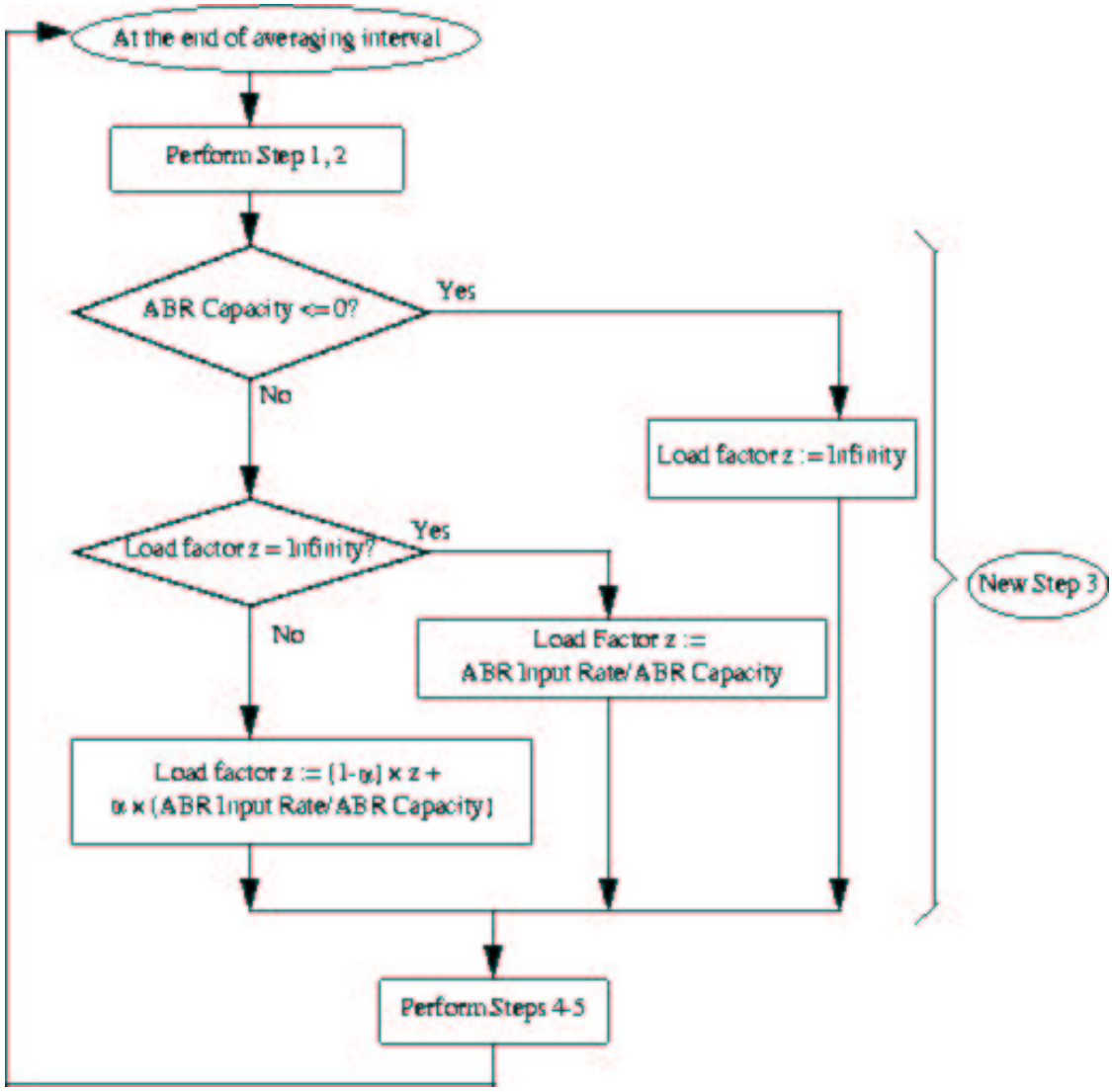


Figure C.6: Flow chart of averaging of load factor (method 1)

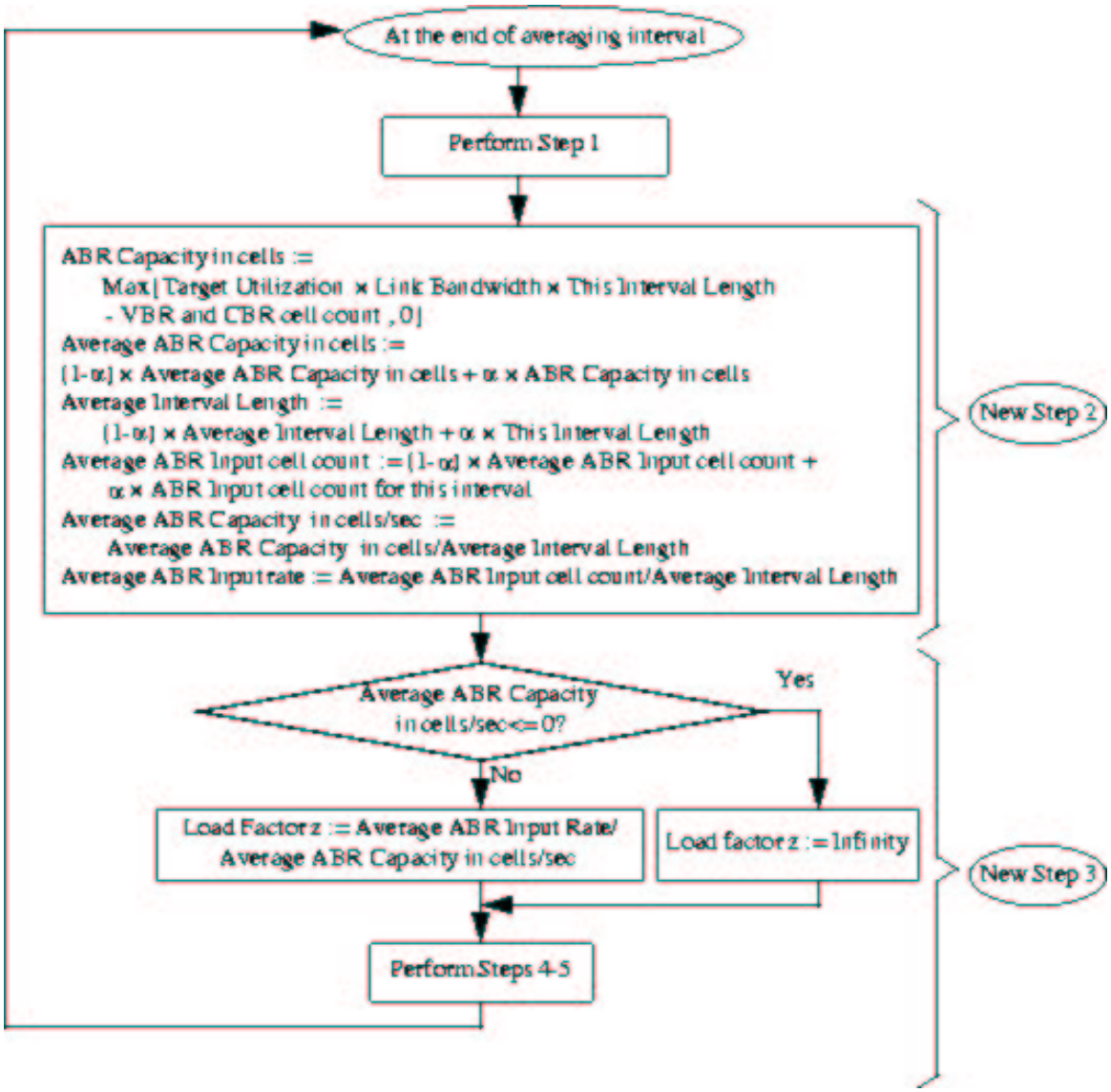


Figure C.7: Flow chart of averaging of load factor (method 2)

C.3 Pseudocode for VS/VD Design Options

The pseudo code describes the combination of the following options:

a) VC's rate from FRM1

b) VC's rate from FRM2

c) VC's rate from measured VC's source rate to per class queues.

A) Measure input rate at entry to per VC queues

B) Measure input rate at entry to per class queues

I) Allocated rate update at FRM turnaround only. Link congestion affects previous loop only.

II) Allocated rate update at BRM receive only. Link congestion affects previous loop and next loop.

III) Allocated rate update at FRM turnaround and BRM receive.Link congestion affects previous loop and next loop.

Observe that the only acceptable combinations are:

a), A), I)

a), A), II)

a), A), III)

c), B), I)

c), B), II)

c), B), III)

C.4 Pseudocode

FRM/BRM/Data receive

(* — switch receives a cell from 'VC' — *)

IF(Opt A) Measure_Input_Rate() ENDIF (* Opt A *)

IF cell is an ABR cell

(* — VD Code: FRM receive — *)

IF cell is an FRM cell

(* — Opt a: CCR update from FRM1 (for ERICA table)— *)

IF (Opt a) CCR ←CCR from FRM ENDIF (* Opt a *)

(* — Link (switch) bottleneck rate — *)

IF (Opt I or III) newER ←Calculate_Allocated_Rate() (* VAL update *)

ELSE IF (Opt II)

newER ←Latest_Allocated_Rate[VC] (* VAL table lookup *)

ENDIF

(* — Bottleneck rate of next loop —- *)

newER ←Minm(newER, ACR of the VC on next loop)

(* — Source bottleneck rate (cell-¿ER) — *)

newER ←Minm(newER,cell-¿ER)

ER_TA for the VC ←newER

(* — Link congestion propogation to the next loop: set ACR — *)

IF (Opt II or III used)

ACR (of the VC on next loop) ←Min(ACR , newER

(* – CCR update from FRM2 (or ACR) — *)

IF (Opt b) used) CCR ←ACR for the VC on next loop ENDIF

ENDIF

turn_around ←1 (* BRM will be generated *)

Turnaround BRM as in DES rules 2-6 (See appendix A)

free the FRM cell

(* —- VS code: BRM receive —- *)

ELSE IF cell is an BRM cell

ACR of the VC ←According to SES rules 8-9

IF (Opt II or III) (* Opt II or III *)

newER ←ERICA algorithm

ACR for the VC ←Minm(ACR, newER)(* Link congestion *)

(* Opt b: CCR update from FRM2 (ACR) *)

IF (Opt b used) CCR ←ACR for the VC

ENDIF

IF (rescheduling option) reschedule the cell sending of this VC ENDIF

free the BRM cell

(* —- VS code: data receive —- *)

ELSE IF cell is a data cell

enqueue the cell to per VC queue in VS of the switch

schedule the sending of this VC queue if necessary

ENDIF (* Cell type: FRM/BRM/data *)

ENDIF (* ABR cell *)

(* — More VS code: FRM/BRM/Data send — *)

(* There is a cell in the VS queue of a source and it

is the scheduling slot for the source ... *)

Follow SES Rules 1-4 (see appendix A).

(* Before Rule 5 *)

(* T = inter FRM time used in Rule 5 test *)

SR ←Nrm/T

IF (Opt c) CCR ←SR ENDIF (* Opt c *)

SES Rules 6 etc (see appendix A).

(* — Before enqueuing the cell (data/FRM/BRM) to per class queue — *)

(* Input rate to per class queue *)

IF (Opt B) Measure_Input_Rate() ENDIF

Enqueue the cell to the ABR per-class queue.

(* END of VS/VD options *)

APPENDIX D

GLOSSARY OF COMMONLY USED ACRONYMNS

ABR - Available Bit Rate

ACR - Allowed Cell Rate

ADTF - ACR Decrease Time Factor

AI - Averaging Interval

APRC Scheme - Adaptive Proportional Rate Control scheme

ATM - Asynchronous Transfer Mode

BECN - Backward Explicit Congestion Notification

BN bit - backward notification bit (RM cell)

BRM - Backward RM cell

CAC - Connection Admission Control

CAPC2 Scheme- Congestion Avoidance using Proportional Control scheme, version 2

CBR - Constant Bit Rate

CCR - Current Cell Rate

CDF - Cutoff Decrease Factor

CI bit - Congestion Indication bit

CLP - Cell Loss Priority

CRM - Missing RM-cell Count

DIR bit - direction bit (RM cell)

DMRCA Scheme - Dynamic Max Rate Control Algorithm

EFCI - Explicit Forward Congestion Indicator

EOM cell - End of Message cell

EPRCA Scheme - Enhanced Proportional Rate Control Algorithm

ER - Explicit Rate

ERICA(+) - Explicit Rate Indication for Congestion Avoidance Schemes

FCVC - Flow Controlled Virtual Circuits (Credit Based Scheme)

FD - Feedback Delay

FIFO - First In First Out

FMMRA Scheme - Fast Max-Min Rate Allocation scheme

FRM - Forward RM cell

FRTT - Fixed Round-Trip Time

HKUST Scheme - Hong Kong University of Science and Technology scheme

ICR - Initial Cell Rate

IRCT - Inter-RM Cell Time

ITU-T - International Telecommunications Union, Telcommunications Sector

LAN - Local area network

LANE - LAN Emulation

LRD traffic - Long range dependendent traffic

MCR - Minimum Cell Rate

MIT Scheme - Massachussetts Institute of Technology Scheme

MPEG-2 standard - Motion Pictures Experts Group Standard for compression, version 2

MPOA - Multiprotocol over ATM

MSS - Maximum Segment Size

Mrm - Controls bandwidth allocation between FRM, BRM and data cells

NI bit - No Increase bit

Nrm - Number of cells between FRM cells

OCR - Offered Average Cell Rate

OSU Scheme - Ohio State University Scheme

PCR - Peak Cell Rate

PRCA - Proportional Rate Control Algorithm

PTI - Payload Type Indicator

QoS - Quality of Service

RDF - Rate Decrease Factor

RIF - Rate Increase Factor

RM cells - Resource Management cells

RTT - Round Trip Time

SFD - Shortest Feedback Delay

SPTS - Single Program Transport Stream (MPEG-2)

TBE - Transient Buffer Exposure

TCP/IP - Transmission Control Protocol/Internet Protocol (layer 4/3 of the Internet)

TCR - a) Transmitted Cell Rate (OSU scheme) b) Tagged Cell Rate (SES parameter)

TM4.0 - ATM Traffic Management Specification, version 4.0

TUB - Target Utilization Band

Trm - Upper Bound on Inter-FRM Time

U - Target Utilization parameter in OSU, ERICA schemes

UCSC Scheme - University of Santa Cruz Scheme

UILI policies - Use-it-or-Lose-it policies

VBR - Variable Bit Rate (comes in the -rt (real-time) and -nrt (non-real time) flavors)

VC - Virtual Circuit

VS/VD - Virtual Source/Virtual Destination Option

WAN - Wide area network

BIBLIOGRAPHY

[1] Santosh P. Abraham and Anurag Kumar. Max-Min Fair Rate Control of ABR Connections with Nonzero MCRs. *IISc Technical Report*, 1997.

[2] Yehuda Afek, Yishay Mansour, and Zvi Ostfeld. Convergence Complexity of Optimistic Rate Based Flow Control Algorithms. In *28th Annual Symposium on Theory of Computing (STOC)*, pages 89–98, 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] Anthony Alles. ATM Internetworking. White paper, Cisco Systems, http://www.cisco.com, May 1995.

[5] G.J. Armitage and K.M. Adams. ATM Adaptation Layer Packet Reassembly during Cell Loss. *IEEE Network Magazine*, September 1993.

[6] 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.

[7] A.W.Barnhart. Changes Required to the Specification of Source Behavior. ATM Forum 95-0193, February 1995.

[8] A.W.Barnhart. Evaluation and Proposed Solutions for Source Behavior # 5. ATM Forum 95-1614, December 1995.

[9] A. W. Barnhart. Use of the Extended PRCA with Various Switch Mechanisms. ATM Forum 94-0898, 1994.

[10] A. W. Barnhart. Example Switch Algorithm for TM Spec. ATM Forum 95-0195, February 1995.

[11] J. Bennett, K. Fendick, K.K. Ramakrishnan, and F. Bonomi. RPC Behavior as it Relates to Source Behavior 5. ATM Forum 95-0568R1, May 1995.

[12] J. Bennett and G. Tom Des Jardins. Comments on the July PRCA Rate Control Baseline. ATM Forum 94-0682, July 1994.

[13] J. Beran, R. Sherman, M. Taqqu, and W. Willinger. Long-Range Dependence in Variable-Bit-Rate Video Traffic. *IEEE Transactions on Communications*, 43(2/3/4), February/March/April 1995.

[14] U. Black. *ATM: Foundation for Broadband Networks*. Prentice Hall, New York, 1995.

[15] P. E. Boyer and D. P. Tranchier. A reservation principle with applications to the atm traffic control. *Computer Networks and ISDN Systems*, 1992.

[16] D. Cavendish, S. Mascolo, and M. Gerla. SP-EPRCA: an ATM Rate Based Congestion Control Scheme basedon a Smith Predictor. Technical report, UCLA, 1997.

[17] Y. Chang, N. Golmie, L. Benmohamed, and D. Siu. Simulation study of the new rate-based eprca traffic management mechanism. *ATM Forum 94-0809*, 1994.

[18] A. Charny, G. Leeb, and M. Clarke. Some Observations on Source Behavior 5 of the Traffic Management Specification. ATM Forum 95-0976R1, August 1995.

[19] Anna Charny. An Algorithm for Rate Allocation in a Cell-Switching Network with Feedback. Master's thesis, Massachusetts Institute of Technology, May 1994.

[20] Anna Charny, David D. Clark, and Raj Jain. Congestion control with explicit rate indication. In *Proceedings of the IEEE International Communications Conference (ICC)*, June 1995.

[21] D. Chiu and R. Jain. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN Systems*, 1989.

[22] Fabio M. Chiussi, Ye Xia, and Vijay P. Kumar. Dynamic max rate control algorithm for available bit rate service in atm networks. In *Proceedings of the IEEE GLOBECOM*, volume 3, pages 2108–2117, November 1996.

[23] D.P.Heyman and T.V. Lakshman. What are the implications of Long-Range Dependence for VBR-Video Traffic Engineering ? *ACM/IEEE Transactions on Networking*, 4(3):101–113, June 1996.

[24] Harry J.R. Dutton and Peter Lenhard. *Asynchronous Transfer Mode (ATM) Technical Overview*. Prentice Hall, New York, 2nd edition, 1995.

[25] H. Eriksson. MBONE: the multicast backbone. *Communications of the ACM*, 37(8):54–60, August 1994.

[26] J. Scott et al. Link by Link, Per VC Credit Based Flow Control. ATM Forum 94-0168, 1994.

[27] L. Roberts et al. New pseudocode for explicit rate plus efci support. *ATM Forum 94-0974*, 1994.

[28] M. Hluchyj et al. Closed-loop rate-based traffic management. *ATM Forum 94-0438R2*, 1994.

[29] M. Hluchyj et al. Closed-Loop Rate-Based Traffic Management. ATM Forum 94-0211R3, April 1994.

[30] S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal, and F. Lu. On source rules for abr service on atm networks with satellite links. In *Proceedings of First International Workshop on Satellite-based Information Services (WOSBIS)*, November 1996.

[31] Chien Fang and Arthur Lin. A Simulation Study of ABR Robustness with Binary-Mode Switches: Part II. ATM Forum 95-1328R1, October 1995.

[32] Chien Fang and Arthur Lin. On TCP Performance of UBR with EPD and UBR-EPD with a Fair Buffer Allocation Scheme. ATM Forum 95-1645, December 1995.

[33] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. Request For Comments, RFC 2068, January 1997.

[34] ATM Forum. http://www.atmforum.com.

[35] ATM Forum. The ATM Forum Traffic Management Specification Version 4.0. ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps, April 1996.

[36] M. Garrett and W. Willinger. Analysis, modeling, and generation of self-similar vbr video traffic. In *Proceedings of the ACM SIGCOMM*, August 1994.

[37] Matthew S. Goldman. Variable Bit Rate MPEG-2 over ATM: Definitions and Recommendations. ATM Forum 96-1433, October 1996.

[38] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, and Seong-Cheol Kim. Performance of TCP over UBR+. ATM Forum 96-1269, October 1996.

[39] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, Bobby Vandalore, Xiangrong Cai, and Seong-Cheol Kim. Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks. ATM Forum 97-0423, April 1997.

[40] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, Bobby Vandalore, Xiangrong Cai, and Seong-Cheol Kim. Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks. ATM Forum 97-0423, April 1997.

[41] M. Grossglauser, S.Keshav, and D.Tse. RCBR: a simple and efficient service for multiple time-scale traffic. In *Proceedings of the ACM SIGCOMM*, August 1995.

[42] S. Hrastar, H. Uzunalioglu, and W. Yen. Synchronization and de-jitter of mpeg-2 transport streams encapsulated in aal5/atm. In *Proceedings of the IEEE International Communications Conference (ICC)*, volume 3, pages 1411–1415, June 1996.

[43] D. Hughes and P. Daley. More abr simulation results. *ATM Forum 94-0777*, 1994.

[44] D. Hunt, Shirish Sathaye, and K. Brinkerhoff. The realities of flow control for abr service. *ATM Forum 94-0871*, 1994.

[45] Van Jacobson. Congestion avoidance and control. In *Proceedings of the ACM SIGCOMM*, pages 314–329, August 1988.

[46] J. Jaffe. Bottleneck Flow Control. *IEEE Transactions on Communications*, COM-29(7):954–962, 1980.

[47] R. Jain. A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks. *Computer Communications Review*, 19.

[48] R. Jain. A timeout-based congestion control scheme for window flow-controlled networks. *IEEE Journal on Selected Areas in Communications*, 1986.

[49] R. Jain. A comparison of hashing schemes for address lookup in computer networks. *IEEE Transactions on Communications*, 1992.

[50] R. Jain. The eprca+ scheme. *ATM Forum 94-0988*, 1994.

[51] R. Jain. The osu scheme for congestion avoidance using explicit rate indication. *ATM Forum 94-0883*, 1994.

[52] R. Jain. Atm networking: Issues and challenges ahead. *Engineers Conference, InterOp+Network World*, 1995.

[53] R. Jain. Congestion control and traffic management in atm networks: Recent advances and a survey. *Computer Networks and ISDN Systems*, 1995.

[54] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared systems. *DEC TR-301*, 1984.

[55] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared systems. *DEC TR-301*, 1984.

[56] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and F. Lu. A Fix for Source End System Rule 5. ATM Forum 95-1660, December 1995.

[57] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and F. Lu. Erica+: Extensions to the erica switch algorithm. *ATM Forum 95-1145R1*, 1995.

[58] R. Jain, S. Kalyanaraman, and R. Viswanathan. Method and apparatus for congestion management in computer networks using explicit rate indication. *U. S. Patent application filed (S/N 307, 375),*, 1994.

[59] R. Jain, S. Kalyanaraman, and R. Viswanathan. The transient performance: Eprca vs eprca++. *ATM Forum 94-1173*, 1994.

[60] R. Jain, S. Kalyanaraman, R. Viswanathan, and R. Goyal. A sample switch algorithm. *ATM Forum 95-0178R1*, 1995.

[61] R. Jain, K. Ramakrishnan, and D Chiu. Congestion avoidance scheme for computer networks. *U.S. Patent #5377322,*, 1994.

[62] R. Jain and K. K. Ramakrishnan. Congestion avoidance in computer networks with a connectionless network layer: Concepts, goals, and methodology. *Proc. IEEE Computer Networking Symposium*, 1988.

[63] R. Jain, K. K. Ramakrishnan, and D. M. Chiu. Congestion Avoidance in Computer Networks with a Connectionless Network Layer. Technical Report DEC-TR-506, Digital Equipment Corporation, August 1987.

[64] R. Jain and S. Routhier. Packet Trains - Measurement and a new model for computer network trafic. *IEEE Journal of Selected Areas in Communications,*, 1986.

[65] Raj Jain. Congestion Control in Computer Networks: Issues and Trends. *IEEE Network Magazine*, pages 24–30, May 1990.

[66] Raj Jain. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, 1991.

[67] Raj Jain. Myths about Congestion Management in High-speed Networks. *Internetworking: Research and Experience*, 3:101–113, 1992.

[68] Raj Jain. ABR Service on ATM Networks: What is it? Network World, 1995.

[69] Raj Jain. Congestion Control and Traffic Management in ATM Networks: Recent advances and a survey. *Computer Networks and ISDN Systems Journal*, October 1996.

[70] Raj Jain, Sonia Fahmy, Shivkumar Kalyanaraman, Rohit Goyal, and Fang Lu. More Straw-Vote Comments: TBE vs Queue sizes. ATM Forum 95-1661, December 1995.

[71] Raj Jain, Shiv Kalyanaraman, Rohit GOyal, and Sonia Fahmy. Source Behavior for ATM ABR Traffic Management: An Explanation. *IEEE Communications Magazine*, 34(11), November 1996.

[72] Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy, and Fang Lu. Bursty ABR Sources. ATM Forum 95-1345, October 1995.

[73] Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy, and Fang Lu. Out-of-Rate RM Cell Issues and Effect of Trm, TOF, and TCR. ATM Forum 95-973R1, August 1995.

[74] Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy, and Fang Lu. Straw-Vote comments on TM 4.0 R8. ATM Forum 95-1343, October 1995.

[75] Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Ram Viswanathan, and Sonia Fahmy. Erica: Explicit rate indication for congestion avoidance in atm networks. U.S. Patent Application (S/N 08/683,871), July 1996.

[76] Raj Jain, Shivkumar Kalyanaraman, and Ram Viswanathan. The osu scheme for congestion avoidance in atm networks: Lessons learnt and extensions. *Performance Evaluation Journal*, October 1997. to appear.

[77] Raj Jain and Shivkumar Kalyanaraman Ram Viswanathan. 'method and apparatus for congestion management in computer networks using explicit rate indication. U. S. Patent application (S/N 307,375), SepJuly 1994.

[78] H. Tzeng K. Siu. Intelligent congestion control for abr service in atm networks. *Computer Communication Review*, 24(5):81–106, October 1995.

[79] Lampros Kalampoukas, Anujan Varma, and K.K. Ramakrishnan. An efficient rate allocation algorithm for atm networks providing max-min fairness. In *6th IFIP International Conference on High Performance Networking (HPN)*, September 1995.

[80] Shivkumar Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, and Jianping Jiang. Performance of TCP over ABR on ATM backbone and with various VBR traffic patterns. In *Proceedings of the IEEE International Communications Conference (ICC)*, June 1997.

[81] Shivkumar Kalyanaraman, Raj Jain, Rohit Goyal, and Sonia Fahmy. A Survey of the Use-It-Or-Lose-It Policies for the ABR Service in ATM Networks. Technical Report OSU-CISRC-1/97-TR02, Dept of CIS, The Ohio State University, 1997.

[82] D. Kataria. Comments on rate-based proposal. *ATM Forum 94-0384*, 1994.

[83] J.B. Kenney. Problems and Suggested Solutions in Core Behavior. ATM Forum 95-0564R1, May 1995.

[84] Bo-Kyoung Kim, Byung G. Kim, and Ilyoung Chong. Dynamic Averaging Interval Algorithm for ERICA ABR Control Scheme. ATM Forum 96-0062, February 1996.

[85] H. T. Kung. Adaptive Credit Allocation for Flow-Controlled VCs. ATM Forum 94-0282, 1994.

[86] H. T. Kung. Flow Controlled Virtual Connections Proposal for ATM Traffic Management. ATM Forum 94-0632R2, September 1994.

[87] T.V. Lakshman, P.P. Mishra, and K.K. Ramakrishnan. Transporting compressed video over atm networks with explicit rate feedback control. In *Proceedings of the IEEE INFOCOM*, April 1997.

[88] L.G.Roberts. Operation of Source Behavior # 5. ATM Forum 95-1641, December 1995.

[89] Hongqing Li, Kai-Yeung Siu, Hong-Ti Tzeng, Chinatsu Ikeda, and Hiroshi Suzuki. Tcp over abr and ubr services in atm. In *Proceedings of IPCCC'96*, March 1996.

[90] S. Liu, M. Procanik, T. Chen, V.K. Samalam, and J. Ormond. An analysis of source rule # 5. ATM Forum 95-1545, December 1995.

[91] B. Lyles and A. Lin. Definition and preliminary simulation f a rate-based congestion control mechanism with explicit feedback of bottleneck rates. *ATM Forum 94-0708*, 1994.

[92] P. Newman. Traffic Management for ATM Local Area Networks. *IEEE Communications Magazine*, 1994.

[93] P. Newman and G. Marshall. Becn congestion control. *ATM Forum 94-789R1*, 1993.

[94] P. Newman and G. Marshall. Update on becn congestion control. *ATM Forum 94-855R1*, 1993.

[95] Craig Partridge. *Gigabit Networking*. Addison-Wesley, Reading, MA, 1993.

[96] Vern Paxson. Fast Approximation of Self-Similar Network Traffic. Technical Report LBL-36750, Lawrence Berkeley Labs, April 1995.

[97] K. Ramakrishnan and R. Jain. A binary feedback scheme for congestion avoidance in computer networks with connectionless network layer. *ACM Transactions on Computers*, 1990.

[98] K. K. Ramakrishnan, D. M. Chiu, and R. Jain. Congestion Avoidance in Computer Networks with a Connectionless Network Layer. Part IV: A Selective Binary Feedback Scheme for General Topologies. Technical report, Digital Equipment Corporation, 1987.

[99] K. K. Ramakrishnan and P. Newman. Credit where credit is due. *ATM Forum 94-0916*, 1994.

[100] K. K. Ramakrishnan and "Issues with Backward Explicit Congestion Notification based Congestion Control. Issues with backward explicit congestion notification based congestion control. *ATM Forum 94-0231*, 1993.

[101] K. K. Ramakrishnan and J. Zavgren. Preliminary simulation results of hop-by-hop/vc flow control and early packet discard. *ATM Forum 94-0231*, 1994.

[102] K.K. Ramakrishnan, P. P. Mishra, and K. W. Fendick. Examination of Alternative Mechanisms for Use-it-or-Lose-it. ATM Forum 95-1599, December 1995.

[103] L. Roberts. The benefits of rate-based flow control for abr service. *ATM Forum 94-0796*, 1994.

[104] L. Roberts. Enhanced prca (proportional rate-control algorithm). *ATM Forum 94-0735R1*, 1994.

[105] L. Roberts. Rate-based algorithm for point to multipoint abr service. *ATM Forum 94-0772R1*, 1994.

[106] Larry Roberts. Enhanced PRCA (Proportional Rate-Control Algorithm). ATM Forum 94-0735R1, August 1994.

[107] A. Romanov. A performance enhancement for packetized abr and vbr+ data. *ATM Forum 94-0295*, 1994.

[108] Allyn Romanov and Sally Floyd. Dynamics of TCP Traffic over ATM Networks. *IEEE Journal on Selected Areas in Communications*, May 1995.

[109] W. Stallings. Isdn and broadband isdn with frame relay and atm. *ATM Forum 94-0888*, 1995.

[110] Lucent Technologies. Atlanta chip set, microelectronics group news announcement, http://www.lucent.com/micro/news/032497.html.

[111] Christos Tryfonas. MPEG-2 Transport over ATM Networks. Master's thesis, University of California at Santa Cruz, September 1996.

[112] H. Tzeng and K. Siu. A class of proportional rate control schemes and simulation results. *ATM Forum 94-0888*, 1994.

[113] H. Tzeng and K. Siu. Enhanced credit-based congestion notification (eccn) flow control for atm networks. *ATM Forum 94-0450*, 1994.

[114] International Telecommunications Union. http://www.itu.ch.

[115] Bobby Vandalore, Shiv Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy, Xiangrong Cai, and Seong-Cheol Kim. Performance of Bursty World Wide Web (WWW) Sources over ABR. ATM Forum 97-0425, April 1997.

[116] Bobby Vandalore, Shiv Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy, and Pradeep Samudra. Worst case TCP behavior over ABR and buffer requirements. ATM Forum 97-0617, July 1997.

[117] L. Wojnaroski. Baseline text for traffic management sub-working group. *ATM Forum 94-0394R4*, 1994.

[118] Gary R. Wright and W. Richard Stevens. *TCP/IP Illustrated, Volume 2.* Addison-Wesley, Reading, MA, 1995.

[119] Lixia Zhang, Scott Shenker, and D.D.Clark. Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic. In *Proceedings of the ACM SIGCOMM*, August 1991.