CHAPTER 5

THE OHIO STATE UNIVERSITY (OSU) SCHEME

The OSU scheme was one of the first attempts to show the power of the explicit rate feedback method having O(1) execution complexity. It was developed as an alternative to the MIT scheme which had O(N) complexity. The EPRCA and APRC were some of the approaches proposed in parallel to the OSU scheme. It should be noted that the OSU scheme was developed at a time when the rate-based framework was being designed in the ATM Forum Traffic Management Group. As we describe the scheme, we shall also discuss the contributions of the scheme towards forming the standards.

5.1 The Scheme

The OSU scheme requires sources to monitor their load and send control cells *periodically* at intervals of T microseconds. These control cells contain source rate information. The switches monitor their own load and use it with the information provided by the control cells to compute a factor by which the source should go up or down. The destination simply returns the control cells to the source, which then adjusts its rate as instructed by the network. This section described the various components of the scheme.

5.1.1 Control-Cell Format

The control cell contains the following fields:

1. Transmitted Cell Rate (TCR): The TCR is the inverse of the minimum inter-cell transmission time and indicates instantaneous peak load input by the source.

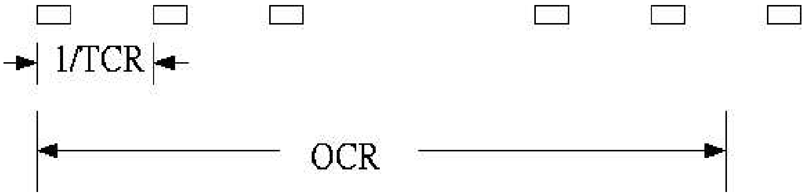


Figure 5.1: Transmitted cell rate (instantaneous) and Offered Average Cell Rate (average).

2. The Offered Average Cell Rate (OCR): For bursty sources which may not send a cell at every transmission opportunity, TCR is not a good indication of overall load. Therefore, the average *measured* load over T interval is indicated in the OCR field of the control cell. The inter-cell time is computed based on the transmitted cell rate. However, the source may be idle in between the bursts and so the average cell rate is different from the transmitted cell rate. This average is called the offered average cell rate and is also included in the cell. This distinction between TCR and OCR is shown in Figure 5.1. Notice that TCR is a control variable (like the knob on a faucet) while the OCR is a measured quantity (like a meter on a pipe). This analogy is shown in Figure 5.2.

3. Load Adjustment Factor (LAF): This field carries the feedback from the network. At the source, the LAF is initialized to zero. Switches on the path can

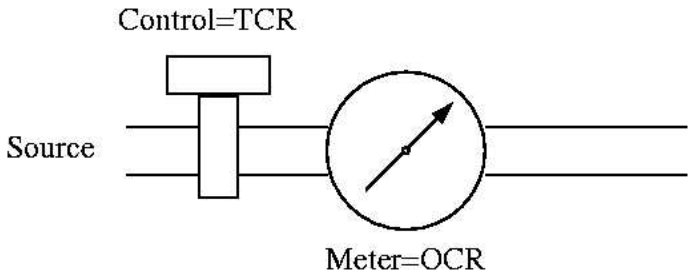


Figure 5.2: Transmitted cell rate (controlled) and Offered Average Cell Rate (measured).

only increase LAF. Increasing the LAF corresponds to decreasing the allowed source rate. Hence, successive switches only reduce the rate allowed to the source. Thus, the source receives the rate allowed by the bottleneck along the path

4. Averaging interval (AI): The OSU scheme primarily uses measured quantities instead of parameters for control. These quantities are measured at the source (eg., OCR) and the switch (eg., current load level z discussed in section 5.1.3). The measurements are done over intervals (called "averaging intervals") to smoothen out the variance in these quantities. To ensure coorelation of the measured quantities at the switch and at the source, we require the source averaging intervals to be the maximum of the averaging interval of the switches along the path. This maximum value is returned in the AI field. The AI field is initialized to zero at the source.

5. The direction of feedback (backward/forward)

6. Timestamp containing the time at which the control cell was generated at the source

The last two fields are used in the backward congestion notification option described in Section 5.2.8 and need not be present if that option is not used.

5.1.2 The Source Algorithm

The source algorithm consists of three components:

1. How often to send control cells

2. How to measure the offered average cell rate

3. How to respond to the feedback received from the network

These three questions are answered in the next three subsections.

Control-Cell Sending Algorithm

The sources send a control cell into the network every T microseconds. The source initializes all the fields. The network reads only the OCR, LAF and AI fields and modifies only the LAF and AI fields. The TCR field is used by the source to calculate the new TCR as discussed in the next section.

LAF and AI are both initialized to zero as discussed earlier. The initialization of the OCR and TCR fields are discussed in the next section.

Measuring Offered Average Load

Unlike any other scheme proposed so far, each source also measures its own load. The measurement is done over the same averaging interval that is used for sending the control cells. The transmission cell rate (TCR), as defined, is the inverse of minimum inter-cell transmission time at the source. However, when the source is not always active, the average rate of the source is different from the transmitted cell rate. This average is called the offered average cell rate and is also included in the cell.

Normally the OCR should be less than the TCR, except when the TCR has just been reduced. In such cases, the switch will actually see a load corresponding to the previous TCR and so the feedback will correspond to the previous TCR. The OCR, in such cases, is closer to the previous TCR. Putting the maximum of current TCR and OCR in the TCR field helps overcome unnecessary oscillations caused in such instances. In other words,

$$\text{TCR in Cell} \leftarrow \max\{\text{TCR, OCR}\}$$

During an idle interval, no control cells are sent. If the source measures the OCR to be zero, then one control cell is sent, subsequent control cells are sent only after the rate becomes non-zero.

Responding to Network Feedback

The control cells returned from the network contain a “load adjustment factor” along with the TCR. The current TCR may be different from that in the cell. The source computes a new TCR by dividing the TCR in the cell by the load adjustment factor in the cell:

$$\text{New TCR} \leftarrow \frac{\text{TCR in the Cell}}{\text{Load Adjustment Factor in the Cell}}$$

If the load adjustment factor is more than one, the network is asking the source to decrease. If the new TCR is less than the current TCR, the source sets its TCR to the new TCR value. However, if the new TCR is more than current TCR, the source is already operating below the network's requested rate and there is no need make any adjustments.

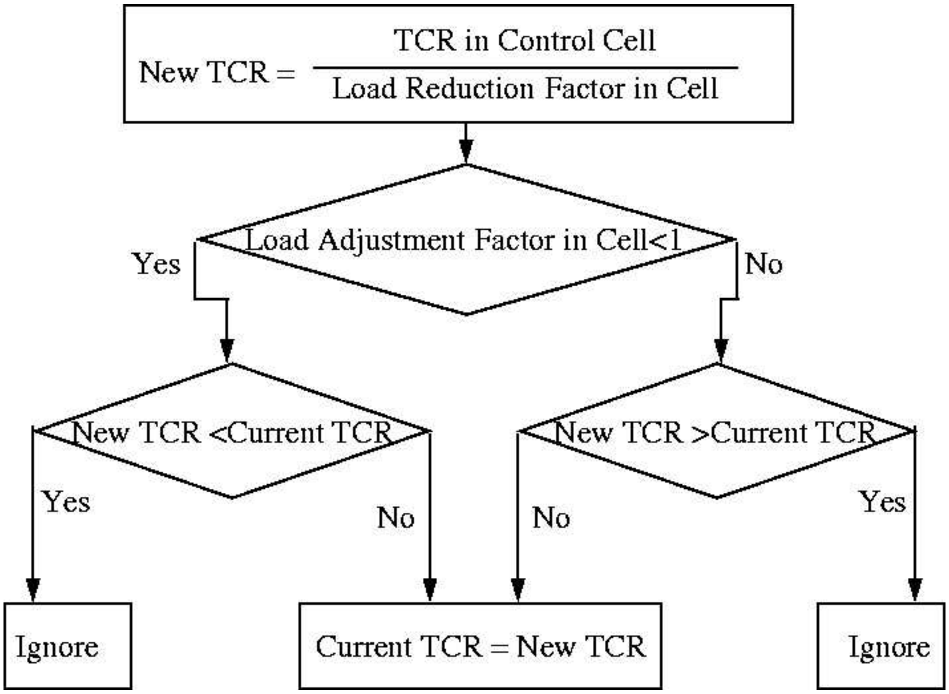

New TCR = TCR in Control Cell / Load Reduction Factor in Cell
Load Adjustment Factor in Cell<1
Yes
No
New TCR <Current TCR
New TCR >Current TCR
Yes
No
No
Yes
Ignore
Current TCR = New TCR
Ignore

Figure 5.3: Flow chart for updating TCR

Similarly, if the load adjustment factor is less than one, the network is permitting the source to increase. If the current TCR is below the new TCR, the source increases its rate to the new value. However, if the current TCR is above the new TCR, the new value is ignored and no adjustment is done. Figure 5.3 presents a flow chart explaining the rate adjustment.

5.1.3 The Switch Algorithm

The switch algorithm consists of two parts: measuring the current load level periodically and calculating the feedback whenever a control cell is received. The feedback calculation consists of an algorithm to achieve efficiency and an algorithm to achieve fairness. The measured value of the current load level is used to decide whether the efficiency or the fairness algorithm is used to calculate feedback.

Measuring The Current Load z

The switch measures its current load level, z , as the ratio of its "input rate" to its "target output rate". The input rate is measured by counting the number of cells *received* by the switch during a fixed averaging interval. The target output rate is set to a fraction (close to 100 %) of the link rate. This fraction, called Target Utilization (U), allows high utilization and low queues in steady state. The current load level z is used to detect congestion at the switch and determine an overload or underload condition.

$$\text{Target Output Cell Rate} = \frac{\text{Target Utilization (U)} \times \text{Link bandwidth in Mbps}}{\text{Cell size in bits}}$$

$$z = \frac{\text{Number of cells received during the averaging interval}}{\text{Target Output Cell Rate} \times \text{Averaging Interval}}$$

The switches on the path have averaging intervals to measure their current load levels (z). These averaging intervals are set locally by network managers. A single value of z is assumed to correspond to *one* OCR value of every source. If two control cells of a source with different OCRs are seen in a single interval (for one value of z), the above assumption is violated and conflicting feedbacks may be given to the source. So, when feedback is given to the sources the AI field is set to the maximum of the AI field in the cell and the switch averaging interval:

$$\text{AI in cell} \leftarrow \text{Max(AI in cell, switch averaging interval)}$$

Achieving Efficiency

Efficiency is achieved as follows:

$$\text{LAF in cell} \leftarrow \text{Max(LAF in cell, } z\text{)}$$

The idea is that if all sources divide their rates by LAF, the switch will have $z = 1$ in the next cycle. In the presence of other bottlenecks, this algorithm converges to $z = 1$. In fact it reaches a band $1 \pm \Delta$ quickly. This band is identified as an efficient operating region. However, it does not ensure fair allocation of available bandwidth among contending sources. When $z = 1$, sources may have an unfair distribution of rates.

Achieving Fairness

Our first goal is to achieve efficient operation. Once the network is operating close to the target utilization, we take steps to achieve fairness. The network manager declares a target utilization band (*TUB*), say, 90±9% or 81% to 99%. When the link utilization is in the TUB, the link is said to be operating efficiently. The TUB is henceforth expressed in the U(1±Δ) format, where U is the target utilization and Δ is the half-width of the TUB. For example, 90±9% is expressed as $90(1 \pm 0.1)\%$. Equivalently, the TUB is identified when the current load level z lies in the interval $1 \pm \Delta$.

We also need to count the number of active sources for our algorithm. The number of active sources can be counted in the same averaging interval as that of load measurement. One simple method is to mark a bit in the VC table whenever a cell from a VC is seen. The bits are counted at the end of each averaging interval and are cleared at the beginning of each interval. Alternatively a count variable could be incremented when the bit is changed from zero to one. This count variable and the bits are cleared at the end of the interval.

Given the number of active sources, a fair share value is computed as follows:

$$\text{FairShare} = \frac{\text{Target Cell Rate}}{\text{Number of Active Sources}}$$

Underloading sources are sources that are using bandwidth less than the FairShare and overloading sources are those that are using more than the FairShare. To achieve fairness, we treat underloading and overloading sources differently. If the current load level is z, the underloading sources are treated as if the load level is $z/(1+\Delta)$ and the overloading sources are treated as if the load level is $z/(1-\Delta)$.

$$\text{If (OCR in cell} < \text{FairShare)}\ \ \text{LAF in cell} \leftarrow \text{Max(LAF in cell, } \frac{z}{(1+\Delta)})\}$$

$$\text{else LAF in cell} \leftarrow \text{Max(LAF in cell, } \frac{z}{(1-\Delta)})\}$$

We prove later in this chapter that this algorithm guarantees that the system, once in the TUB, remains in the TUB, and consistently moves towards fair operation. We note that all the switch steps are O(1) w.r.t. the number of VCs.

If Δ is small, as is usually the case, division by $1+\Delta$ is approximately equivalent to a multiplication by $1-\Delta$ and vice versa.

What Load Level Value to Use?

The OCR in the control cell is correlated to z when the control cell *enters* the switch queue. This is because the queue state at arrival more accurately reflects the effect of the TCR indicated in the control cell. The value of z may change before the control cell leaves the switch queue. The OCR in the cell at the time of leaving the queue is not necessarily co-related with z. As shown in Figure 5.4, the queue state at the time of departure (instant marked “2” in the figure) depends upon the load that the source put after the control cell had left the source. This subsequent load may be very different from that indicated in the cell.

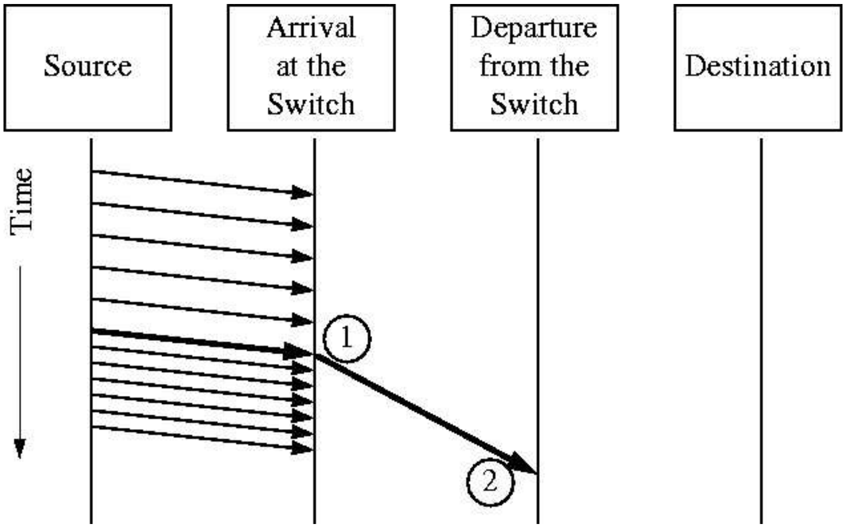


Figure 5.4: Correlation of Instantaneous Queue States to TCR

5.1.4 The Destination Algorithm

The destination simply returns all control cells back to the source.

5.1.5 Initialization Issues

When a source first starts, it may not have any idea of the averaging interval or what rate to use initially. There are two answers. First, since ATM networks are connection-oriented, the above information can be obtained during connection setup. For example, the averaging interval and the initial rate may be specified in the connection accept message. Second, it is possible to send a control cell (with TCR=OCR=0) and wait for it to return. This will give the averaging interval. Then the source can pick any initial rate and start transmitting. It can use the averaging interval returned in the feedback to measure OCR, and at the end of the averaging interval send a control cell containing this OCR. When the control cell returns, it will have the information to change to the correct load level.

Since the averaging intervals depend upon the path, averaging interval may be known to the source host from other VCs going to the same destination host. Also, a network manager may hardcode the same averaging interval in all switches and hosts. We do not recommend this procedure since not all switches that a host may eventually use may be in the control of the network manager.

The initial transmission cell rate affects the network operation for only the first few (one or two) round trips. Therefore, it can be any value below (and including) the target cell rate of the link at the source. However, network managers may set any other initial rate to avoid startup impulses.

5.2 Key Features and Contributions of the OSU scheme

The OSU scheme was presented to the ATM Forum traffic management working group in its September and October 1994 meetings. It highlighted several new ideas that have now become common features of most such schemes developed since then. This includes applying the concept of congestion avoidance to rate-based algorithms and the use of input rate instead of queue length for congestion detection. The number of parameters is small and their effects are well understood.

5.2.1 Congestion Avoidance

The OSU scheme is a congestion *avoidance* scheme. As defined in [48], a congestion avoidance scheme is one that keeps the network at high throughput and low delay in the steady state. The system operates at the *knee* of the throughput delay-curve as shown in Figure 3.1.

The OSU scheme keeps the steady state bottleneck link utilization in the target utilization band (TUB). The utilization is high and the oscillations are bounded by the TUB. Hence, in spite of oscillations in the TUB, the load on the switch is always less than one. So the switch queues are close to zero resulting in minimum delay to sources.

The target utilization and target utilzation band per-link parameters are set by the network manager based on the cost of the bandwidth, and the anticipated degree of variance in the network demand and capacity. The target utilization affects the rate at which the queues are drained during overload. A higher target utilization reduces unused capacity but increase the time to reach the efficient region after a disturbance. A lower target utilization may be necessary to cope with the effects of variance in capacity and demand due to the introduction of errors introduced in measurement as a result of variance. A wide TUB results in a faster progress towards fairness. In most cases, a TUB of $90\%(1 \pm 0.1)$ is a good choice. This gives a utilization in the range of 81% to 99%.

5.2.2 Parameters

The OSU scheme requires just three parameters: the switch averaging interval (AI) , the target link utilization (U) , and the half-width of the target utilization band (Δ).

The target utilization (U) and the TUB present a few tradeoffs. During overload (transients), U affects queue drain rate. Lower U increases drain rate during transients, but reduces utilization in steady state. Further, higher U also constrains the size of the TUB.

A narrow TUB slows down the convergence to fairness (since the formula depends on Δ) but has smaller oscillations in steady state. A wide TUB results in faster progress towards fairness, but has more oscillations in steady state. We find that a TUB of $90\%(1 \pm 0.1)$ used in our simulations is a good choice.

The switch averaging interval affects the stability of z. Shorter intervals cause more variation in the z and hence more oscillations. Larger intervals cause slow feedback and hence slow progress towards steady state.

The OSU scheme parameters can be set relatively independent of the target workload and network extent. Variance in measurement is the key error factor in the OSU scheme, and a larger interval is desirable to smooth the effect of such variance. Some schemes, on the other hand, are very sensitive to the workload and network diameter in their choice of parameter values. An easy way to identify such schemes is that they recommend different parameter values for different network configurations. For example, a switch parameter may be different for WAN configurations than in a LAN configuration. A switch generally has some VCs travelling short distances while others travelling long distances. While it is ok to classify a VC as a local or wide area VC, it is often not correct to classify a switch as a LAN switch or a WAN switch. In a nationwide internet consisting of local networks, all switches could be classified as WAN switches. Note that the problem becomes more difficult when the scheme uses many parameters, and/or the parameters are not independent of each other.

5.2.3 Use Measured Rather Than Declared Loads

Many schemes prior to OSU scheme, including the MIT scheme, used source declared rates for computing their allocation without taking into account the actual load at the switch. In the OSU scheme, we measure the current total load. All unused capacity is allocated to contending sources. We use the source's declared rate to compute a particular sources' allocation but use the switch's measured load to decide whether to increase or decrease. Thus, if the sources lie or if the source's information is out-of-date, our approach may not achieve fairness but it still achieves efficiency.

For example, suppose a personal computer connected to a 155 Mbps link is not be able to transmit more than 10 Mpbs because of its hardware/software limitation. The source declares a desired rate of 155 Mbps, but is granted 77.5 Mbps since there is another VC sharing the link going out from the switch. Now if the computer is unable to use any more than 10 Mbps, the remaining 67.5 Mbps is reserved for it and cannot be used by the second VC and the link bandwidth is wasted.

The technique of measuring the total load has become minimum required part of most switch algorithms. Of course, some switches may measure each individual source's cell rate rather than relying on the information in the RM cell

5.2.4 Congestion Detection: Input Rate vs Queue Length

Most congestion control schemes for packet networks in the past were window based. Most of these schemes use queue length as the indicator of congestion. Whenever the queue length (or its average) is more than a threshold, the link is considered congested. This is how initial rate-based scheme proposals were also being designed.

We argued that the queue length is not a good indicator of load when the control is rate-based.

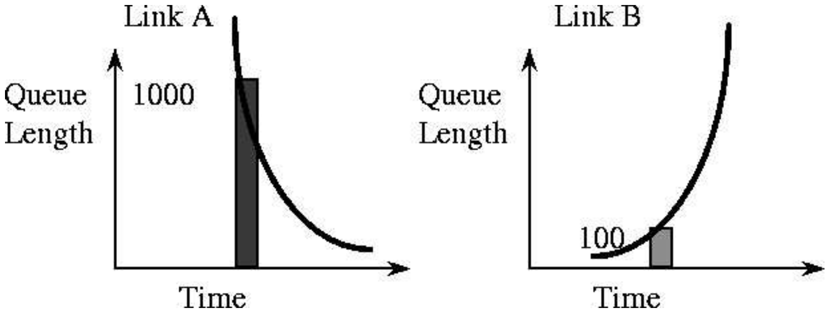


Figure 5.5: Congestion Detection Metric: Queue Length or Input Rate ?

As an example, consider two rate controlled queues as shown in figure 5.5. Suppose the first queue is only 100 cells long while the other is 1000 cells long. Without further information it is not possible to say which queue is overloaded. For example, if the first queue is growing at the rate of 1000 cells per second, it is overloaded while the second queue may be decreasing at a rate of 1000 cells per second and may actually be underloaded. Further, if the first queue can be processed at 622 Mbps, the queueing delay is much smaller than that of a 100 cell queue processed at 1.54 Mbps. This factor becomes important because the capacity available to ABR can be quite variable.

Another important reason for the choice of the input rate metric has to do with rate and window controls. For a detailed discussion of rate versus window, see Jain (1990) [65]. In particular, a window controls the queue length, while the rate controls the queue growth rate. Given a particular window size, the maximum queue length can be guaranteed to be below the window. Given an input rate to a queue, the queue growth rate can be guaranteed below the input rate but there is nothing that can be said about the maximum queue length. Queue length gives no information about the difference between current input rate and the ideal rate.

With rate-based control, the input rate is a better indicator of congestion. If the input rate is lower than available capacity, the link is not congested even if the queue lengths are high because the queue will be decreasing. Similarly, if the input rate is higher than the available capacity, the system should start taking the steps to reduce congestion since the queue length will be increasing.

Monitoring input rates not only gives a good indication of load level, it also gives a precise indication of overload or underload. For example, if the input rate to a queue is 20 cells per second when the queue server can handle only 10 cells per second, we know that the queue overload factor is 2 and that the input rate should be decreased by a factor of 2. No such determination can be made based on instantaneous queue length. The input rate can hence be used as a metric to compute the new rate allocations. The use of input rates as a metric avoids the use of unnecessary parameters.

The OSU scheme uses the input rate to compute the overload level and adjust the source rates accordingly. Each switch counts the number of cells that it received on a link in a given period, computes the cell arrival rate and hence the overload factor using the known capacity (in cells per second) of the link. It tries to adjust the source rate by a factor equal to the overload level and thus attempts to bring it down to the correct level as soon as possible.

In the later ERICA+ work described in this dissertation, we use the ***queueing delay*** as a *secondary metric* for congestion detection with input rate being the primary metric.

5.2.5 Bipolar Feedback

A network can provide two kinds of feedback to the sources. Positive feedback tells the sources to increase their load. Negative feedback tells the sources to decrease their load. These are called two polarities of the feedback. Some schemes are bipolar in the sense that they use both positive and negative feedback. The OSU scheme uses both polarities. The DECbit scheme [63] is another example of a bipolar scheme.

Some schemes use only one polarity of feedback, say positive. Whenever, the sources receive the feedback, they increase the rate and when they don't receive any feedback, the network is assumed to be overloaded and the sources automatically decrease the rate without any explicit instruction from the network. Such schemes send feedback only when the network is underloaded and avoid sending feedback during overload. The PRCA scheme [28] is an example of a unipolar scheme with positive polarity only.

Unipolar schemes with negative polarity are similarly possible. Early versions of PRCA used negative polarity in the sense that the sources increased the rate continuously unless instructed by to network to decrease. The slow start scheme used in TCP/IP is also an example of unipolar scheme with negative polarity although in this case the feedback (packet loss) is an implicit feedback (no bits or control packets are sent to the source).

The MIT scheme is unipolar with only negative feedback to the source. The switches can only reduce the rate and not increase it. For increase, the source has to send another control cell with a higher desired rate. Thus, increases are delayed resulting in reduced efficiency.

The key problem with some unipolar schemes is that the load is changed continuously – often on every cell. This may not be desirable for some workloads, such as compressed video traffic. Every adjustment in rate requires the application to adjust its parameters. Bipolar schemes can avoid the unnecessary adjustments by providing explicit instructions to the sources only when a load change is required.

One reason for prefering unipolar feedback in some cases is that the number of feedback messages is reduced. However, this is not always true. For example, the MIT and OSU schemes have the same data cell to control cells ratio. In the MIT scheme, a second control cell has to be sent to determine the increase amount during underload. This is avoided in the OSU scheme by using a bipolar feedback.

Since current ATM specifications allow the switches to increase or decrease the rate of a source, all ATM switch implementations are expected to be bipolar.

5.2.6 Count the Number of Active Sources

The OSU scheme introduced the concept of averaging interval and active sources. Most of the virtual circuits (VCs) in an ATM network are generally idle. Its the number of active VCs rather than the total number of VCs that is meaningful. We compute use the number of active VCs to compute fairshare. As discussed in section 5.10, if the measured value is wrong (which is possible if the averaging interval is short), fairness may be affected.

Other schemes like EPRCA attempt to achieve fairness without measuring the number of active sources. The technique they use is to advertise a single rate to all sources and parametrically increase or decrease the advertized rate.

5.2.7 Order 1 Operation

The MIT scheme uses an iterative procedure to calculate the feedback rate. Further it requires the switches to remember the rates for all VCs and. Therefore, its computation and storage complexity is of the order of n, O(n). This makes it somewhat undesirable for large switches that may have thousands of VCs going through it at any one time. The basic OSU scheme does not need all the rates at the same time and has a computational complexity of O(1).

5.2.8 Backward Congestion Notifications Cannot Be Used to Increase

One problem with end-to-end feedback schemes is that it may take long time for the feedback to reach the source. This is particularly true if the flow of RM cells has not been established in both directions. In such cases, switches can optionally generate their own RM cell and send it directly back to the source.

The OSU scheme research showed that indiscriminate use of BECNs can cause problems. For example, consider the case shown in Figure 5.6. The source is sending at 155 Mbps and sends a RM cell. The switch happens to be underloaded at that time and so lets the first RM cell (C1) go unchanged. By the time the second RM cell (C2) arrives, the switch is loaded by a factor of 2 and sends a BECN to the source to come down to 77.5 Mbps. A little later C1 returns asking the source to change to 155 Mbps. The RM cells are received out of order rendering the BECN ineffective. To ensure correct operation of the BECN option, we established a set of rules. These rules are described later in Section 5.3.3. The first two of the six rules described there are now part of the Traffic Management specifications.

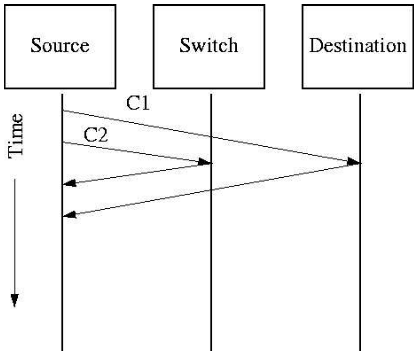


Figure 5.6: Space time diagram showing out-of-order feedback with BECN

5.3 Extensions of The OSU Scheme

5.3.1 Aggressive Fairness Option

In the basic OSU scheme, when a link is outside the TUB, all input rates are adjusted simply by the load level. For example, if the load is 200%, all sources will be asked to halve their rates regardless of their relative magnitude. This is because our goal is to get into the efficient operation region as soon as possible without worrying about fairness. The fairness is achieved after the link is in the TUB.

Alternatively, we could attempt to take steps towards fairness by taking into account the current rate of the source even outside the TUB. However, one has to be careful. For example, when a link is underloaded there is no point in preventing a source from increasing simply because it is using more than its fair share. We cannot be sure that underloading sources can use the extra bandwidth and if we don't give it to an overloading (over the fair share) source, the extra bandwidth may go unused.

The aggressive fairness option is based on a number of considerations. These considerations or heuristics improve fairness while improving efficiency. However, these heuristics do not guarantee convergence to fair operation. We will hence use them outside the TUB, and the TUB algorithm inside the TUB.

The considerations for increase are:

1. When a link is underloaded, all its users will be asked to increase. No one will be asked to decrease.

2. The amount of increase can be different for different sources and can depend upon their relative usage of the link.

3. The maximum allowed adjustment factor should be less than or equal to the current load level. For example, if the current load level is 50%, no source can be allowed to increase by more than a factor of 2 (which is equivalent to a load adjustment factor of 0.5).

4. The load adjustment factor should be a continuous function of the input rate. Any discontinuities will cause undesirable oscillations and impulses. For example, suppose there is a discontinuity in the curve when the input rate is 50 Mbps. Sources transmitting 50-δ Mbps (for a small δ) will get very different feedback than those transmitting at 50+δ Mbps.

5. The load adjustment factor should be a monotonically non-decreasing function of the input rate. Again, this prevents undesirable oscillations. For example, suppose the function is not monotonic but has a peak at 50 Mbps. The sources transmitting at 50+δ Mbps will be asked to increase more than those at 50 Mbps.

The corresponding considerations for overload are similar to the above.

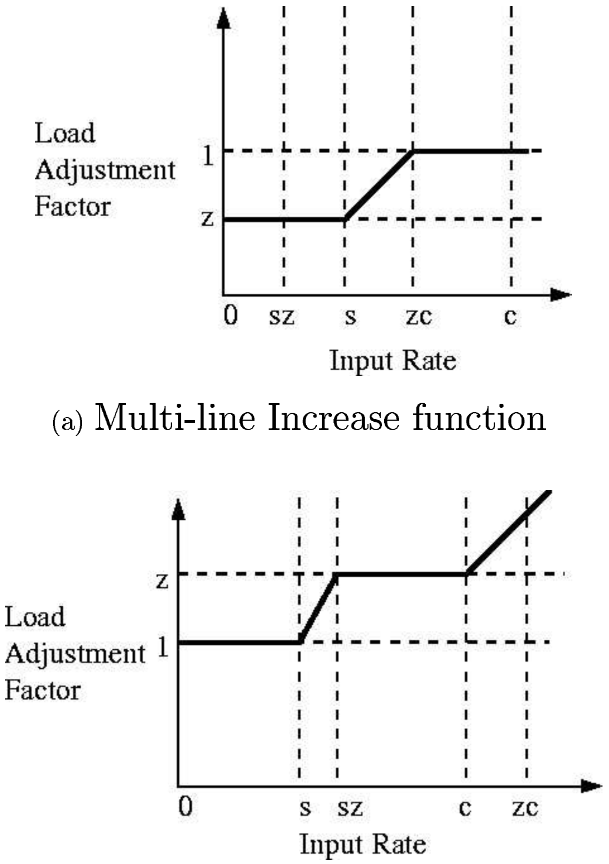
As noted, these heuristics do not guarantee convergence to fairness. To guarantee fairness in the TUB, we violate all of these heuristics except monotonicity.

A sample pair of increase and decrease functions that satisfy the above criteria are shown in Figure 5.7. The load adjustment factor is shown as a function of the input rate. To explain this graph, let us first consider the increase function shown in Figure 5.7(a). If current load level is z, and the fair share is s, all sources with input rates below zs are asked to increase by z. Those between zs and z are asked to increase by an amount between z and 1.

Figure 5.7(b) shows the corresponding decrease function to be used when the load level z is greater than 1. The underloading sources (input rate $x <$ fair share) are not decreased. Those between s and zs are decreased by a linearly increasing factor between 1 and z. Those with rates between zs and c are decreased by the load level z. Those above c are decreased even more. Notice that when the load level z is 1, that is, the system is operating exactly at capacity, both the increase and decrease functions are identical (a horizontal line at load reduction factor of 1). This is important and ensures that the load adjustment factor is a continuous function of z. In designing the above function we used linear functions. However, this is not necessary. Any increasing function in place of sloping linear segments can be used. The linear functions are easy to compute and provide the continuity property that we seek.

The detailed pseudo code of aggressive fairness option is given in appendix B.

Figure 5.8 shows the simulation results for the transient configuration with the aggressive fairness option.



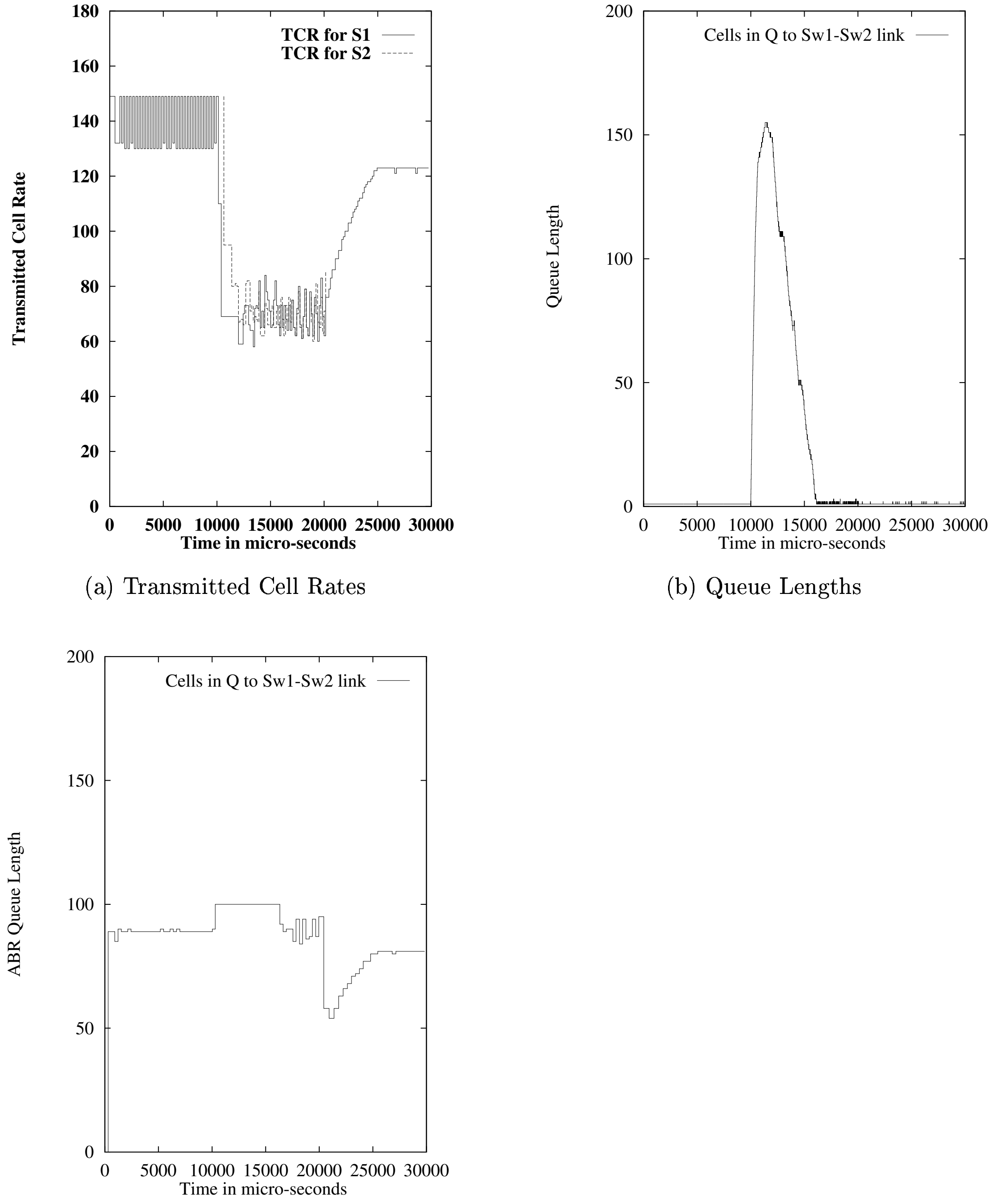
(a) Multi-line Increase function

(b) Multi-line Decrease function

Figure 5.7: Multi-line Increase and Decrease Functions

5.3.2 Precise Fair Share Computation Option

Given the actual rates of all active sources, we can exactly calculate the fair share using the MIT algorithm [20, 19] (MIT scheme uses desired rates). Thus, instead of using only the number of active VCs, we could use the OCRs of various sources to compute the fair share. This option yields a performance much better than that possible with MIT scheme because of the following features that are absent in the MIT scheme:



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 5.8: Simulation results for the experiment with transients and Multi-line fairness option

1. Provide a bipolar feedback. The switches can increase as well decrease the rate in the RM cell. This avoids the extra round trip required for increase in the MIT scheme.

2. Measure the offered average cell rate at the source and use it also to compute the fair share. Using measured value is better than using desired rates.

The detailed pseudo code of precise fair share computation is given in appendix B.

5.3.3 BECN Option

For long-delay paths, backward explicit congestion notifications (BECNs) may help reduce the feedback delay. Experiments with BECNs showed that, BECNs may cause problems unless handled carefully. In particular, we established the following rules for correct operation of the BECN option with OSU scheme:

1. The BECN should be sent only when a switch is overloaded **AND** the switch wants to decrease the rate below that obtained using the LAF field of the RM cell. There is no need to send BECN if the switch is underloaded.

2. The RM cell contains a bit called "BECN bit." This bit is initialized to zero at the source and is set by the congested switch in the BECN cell. The cells that complete the entire path before returning to the source are called forward explicit congestion notification (FECN) cells. They have the bit cleared.

3. All RM cells complete a round-trip. The switch which wants to send a BECN waits until it receives an RM cell, makes two copies of it and sends one copy in the forward direction. The other, called the "BECN cell," is sent back to the source.

4. The RM cell contains a timestamp field which is initialized by the source to the time when the RM cell was generated. The timestamp is ignored everywhere except at the source.

5. The source remembers the timestamp of the last BECN or FECN cell that it has acted upon in a variable called "Time already acted (Taa)." If the timestamp in an returned RM (BECN or FECN) cell is less than Taa, the cell is ignored. This rule helps avoid out-of-order RM cells.

6. If the timestamp of an RM cell received at the source is equal to or greater than Taa, the variable New TCR is computed as in section 5.1.2. In addition, if the BECN bit is set, we ignore the feedback if it directs a rate increase :

   $$\text{IF BECN\_bit AND (TCR} < \text{New TCR) THEN Ignore}$$

   The rate increase has to wait until the corresponding FECN cell returns. BECN is therefore useful only for decrease on long feedback paths.

The ATM forum has adopted the first two of the above rules. The RM cells as specified in the ATM Forum Traffic Management specifications do not contain the timestamps and the last three rules are not relevant to them. These are specific to the OSU scheme. The detailed pseudo code of BECN option is given in appendix B.

One obvious disadvantage of the BECN scheme is that the number of control cells that sent back to the source are increased. Also, since BECN does not have any significant effect in the LAN environment, we recommend its use only in large WANs. This problem was recognized by the ATM Forum which limited the number of BECN cells sent by a switch to 10 cells/sec per-connection.

A complete layered view of various components of the OSU scheme is shown in Figure 5.9. The minimum that we need for correct operation is the fairness algorithm. The aggressive fairness option allows fairness to be achieved faster. The precise fair share computation option allows both fairness and efficiency to be achieved quickly but requires the switches to use all declared OCRs in computing the fair share. The BECN option helps reduce the feedback delay in large WAN cases. As shown in Figure 5.9, these options can be used individually or in a layered manner.

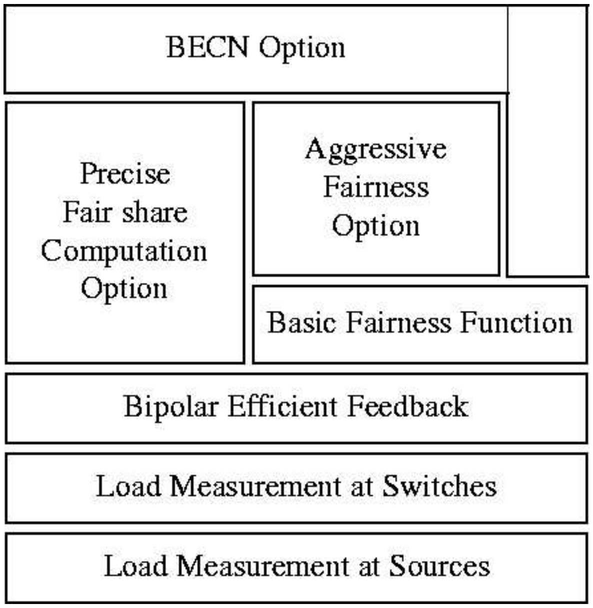

BECN Option
Precise Fair share Computation Option
Aggressive Fairness Option
Basic Fairness Function
Bipolar Efficient Feedback
Load Measurement at Switches
Load Measurement at Sources

Figure 5.9: A layered view of various components and options of the OSU scheme

5.4 Other Simple Variants of the OSU Scheme

Some variations that do not materially change the performance of the OSU scheme are:

1. The source offered average cell rate is measured at the entry switch rather than at the source. This option may be preferable for policing and for operation in public network environments, where a sources' measurements cannot be trusted.

2. The offered average cell rate of a VC is measured at every switch. This is unnecessary since the average rate of a VC should not change from switch to switch. This may be used only if the VC crosses many ATM networks under different administrative domains.

3. Use multiplicative load adjustment factors instead of divisors. In OSU scheme, divisors are used for rates. However, for the inter-cell transmission time, the same factor is used as a multiplier.

4. Use dynamic averaging intervals. The averaging interval at the switch and the source are kept constant in the OSU scheme. It is possible to use regeneration intervals as the averaging interval as was done in the DECbit scheme [63]. However, our experience with DECbit scheme was that implementors didn't like the the regeneration interval and queue length averaging because of the number of instructions required in the packet forwarding path.

5. Use cell counts rather than cell rates. Since the averaging interval is constant, the cell rates are proportional to the counts.

5.5 Simulation Results

In this section, we present simulation results for several configurations. These configurations have been specially chosen to test a particular aspect of the scheme. In general, we prefer to use simple configurations that test various aspects of the scheme. Simple configurations not only save time but also are more instructive in finding problems than complex configurations.

The configurations are presented later in this section in the order in which we use them repeatedly during design phase. For each design alternative, we always start with the simplest configuration and move to the next only if the alternative works satisfactorily for the simpler configurations.

5.5.1 Default Parameter Values

Unless specified otherwise, we assume all links are 1 km long running at 155 Mbps. The infinite source model is used for traffic initially. The burst traffic is considered in Section 5.7. The averaging interval of 300 μs and a target utilization band of $90(1\pm 0.1)\%$ is used.

5.5.2 Single Source

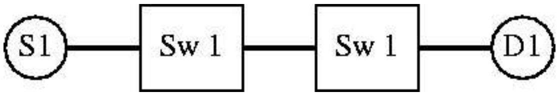
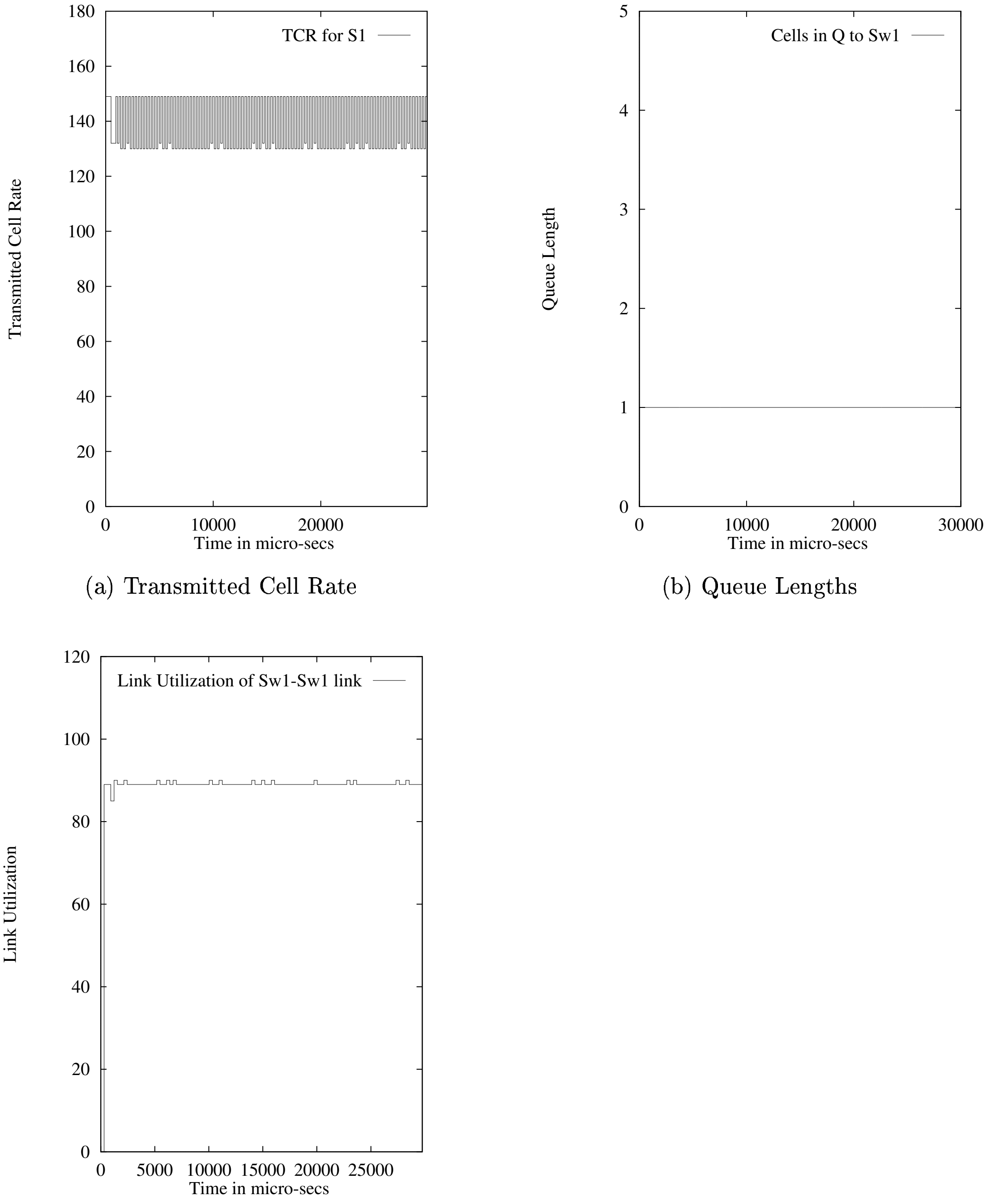


Figure 5.10: Single source configuration

This configuration shown in Figure 5.10 consists of one VC passing through two switches connected via a link. This configuration was helpful in quickly discarding many alternatives. Figure 5.11 shows plots for TCR, link utilization, and queue length at the bottleneck link. Notice that there are no oscillations.



(a) Transmitted Cell Rate

(b) Queue Lengths

(c) Link Utilization

Figure 5.11: Simulation results for the single source configuration

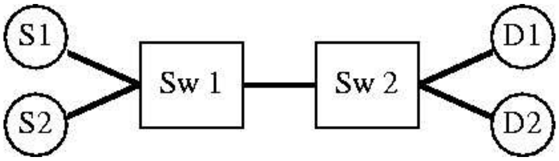


Figure 5.12: Two-source configuration

5.5.3 Two Sources

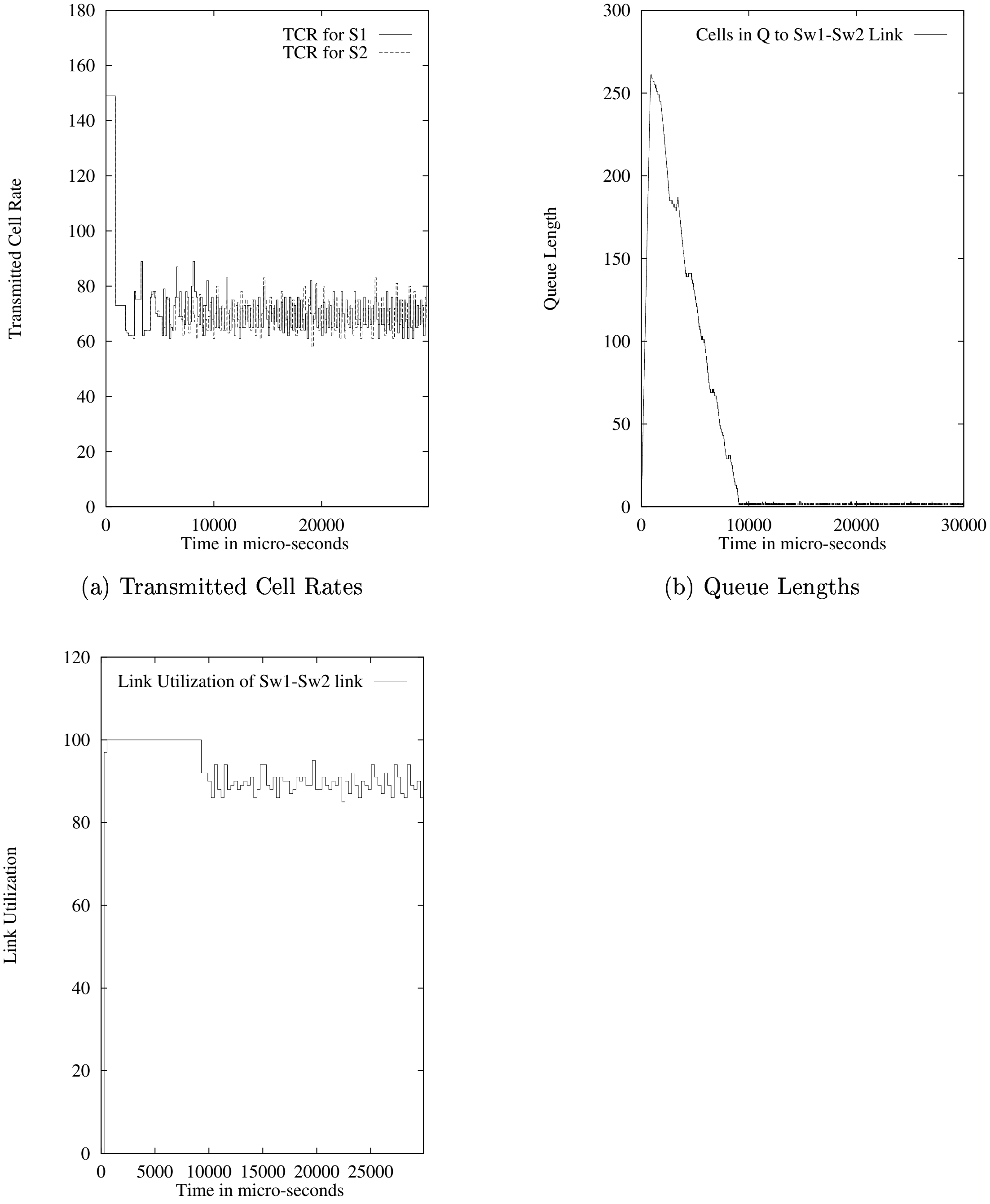
This configuration helps study the fairness. It is similar to the single source configuration except that now there are two sources as shown in Figure 5.12. Figure 5.13 shows the configuration and plots for TCR, link utilization, and queue length at the bottleneck link. Notice that both sources converge to the same level.

5.5.4 Three Sources

As shown in Figure 5.14, this is a simple configuration with one link being shared by three sources. The purpose of this configuration is to check what will happen if the load is such that the link is operating efficiently but not fairly. The starting rates of the three sources are specifically set to values that add up to the target cell rate for the bottleneck link. Figure 5.15 shows the simulation results for this configuration.

5.5.5 Transient Sources

In order to study the effect of new sources coming in the network, we modified the two-source simulation such that the second source comes on after one third of the simulation run and goes off at two third of the total simulation time. The speed at which the TCRs of the two sources decrease and increase to the efficient region can be seen from Figure 5.16.



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 5.13: Simulation results for the two-source configuration

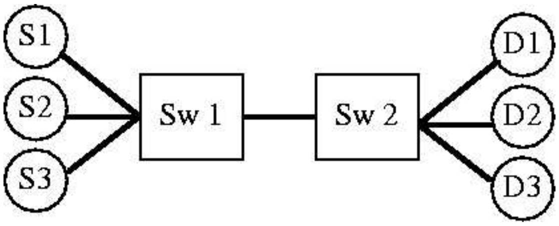


Figure 5.14: Three-source configuration

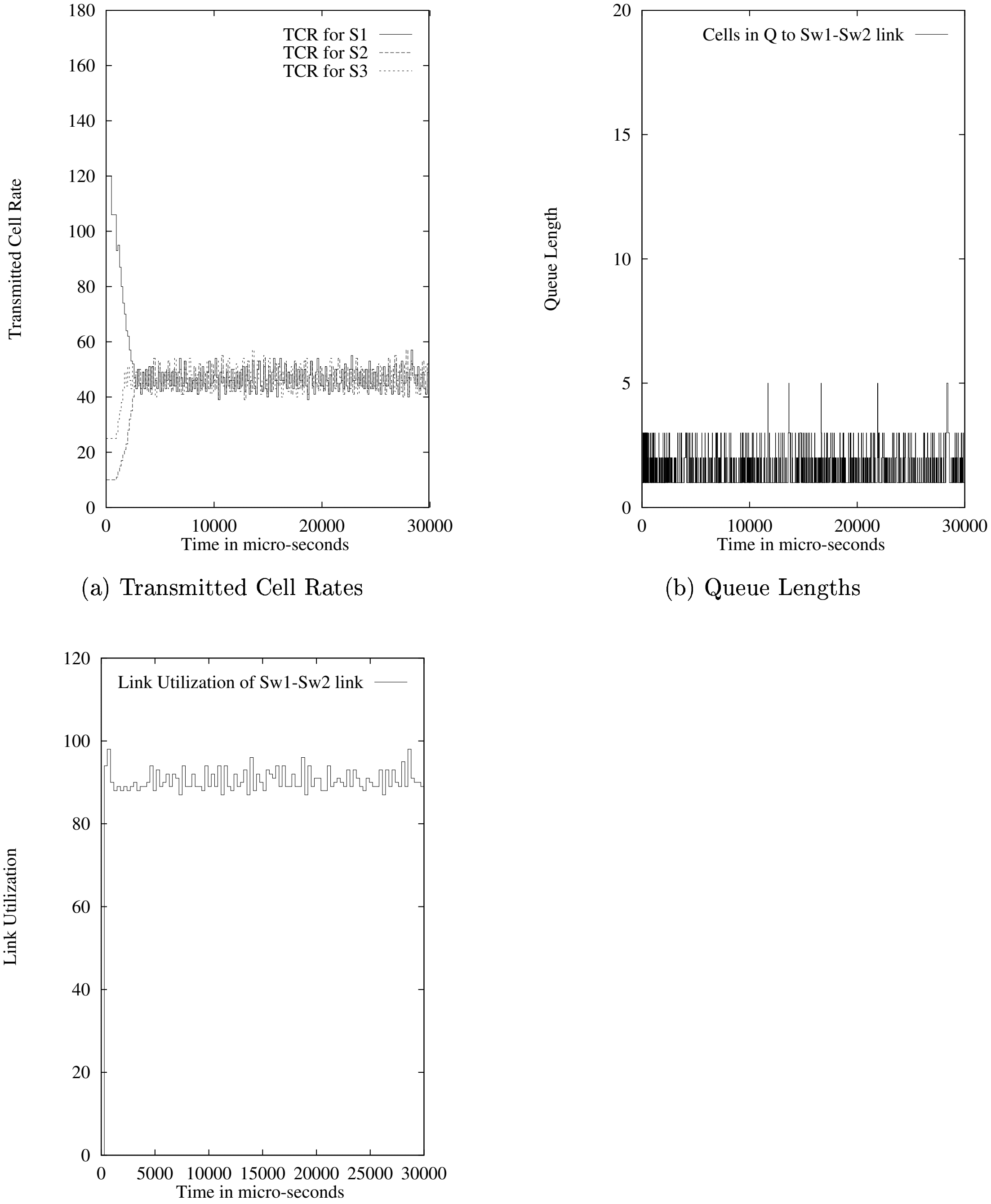
5.5.6 Parking Lot

This configuration is popular for studying fairness. The configuration and its name was derived from theatre parking lots, which consist of several parking areas connected via a single exit path. At the end of the show, congestion occurs as cars exiting from each parking area try to join the main exit stream.

For computer networks, an n-stage parking lot configuration consists of n switches connected in a series. There are n VCs. The first VC starts from the first switch and goes to the end. For the remaining ith VC starts at the $i-1$th switch. A 3-switch parking lot configuration is shown in Figure 5.17. The simlation results are shown in Figure 5.18. Notice that all VCs receive the same throughput without any fair queueing.

5.5.7 Upstream Bottleneck

This configuration consists of four VCs and three switches as shown in Figure 5.19. The second link is shared by VC2 and VC4. However, because of the first link, VC2 is limited to a throughput of 1/3 the link rate. VC4 should, therefore, get 2/3 of the second link. This configuration is helpful in checking if the scheme will allocate all unused capacity to those source that can use it. Figure 5.20 show the simulation

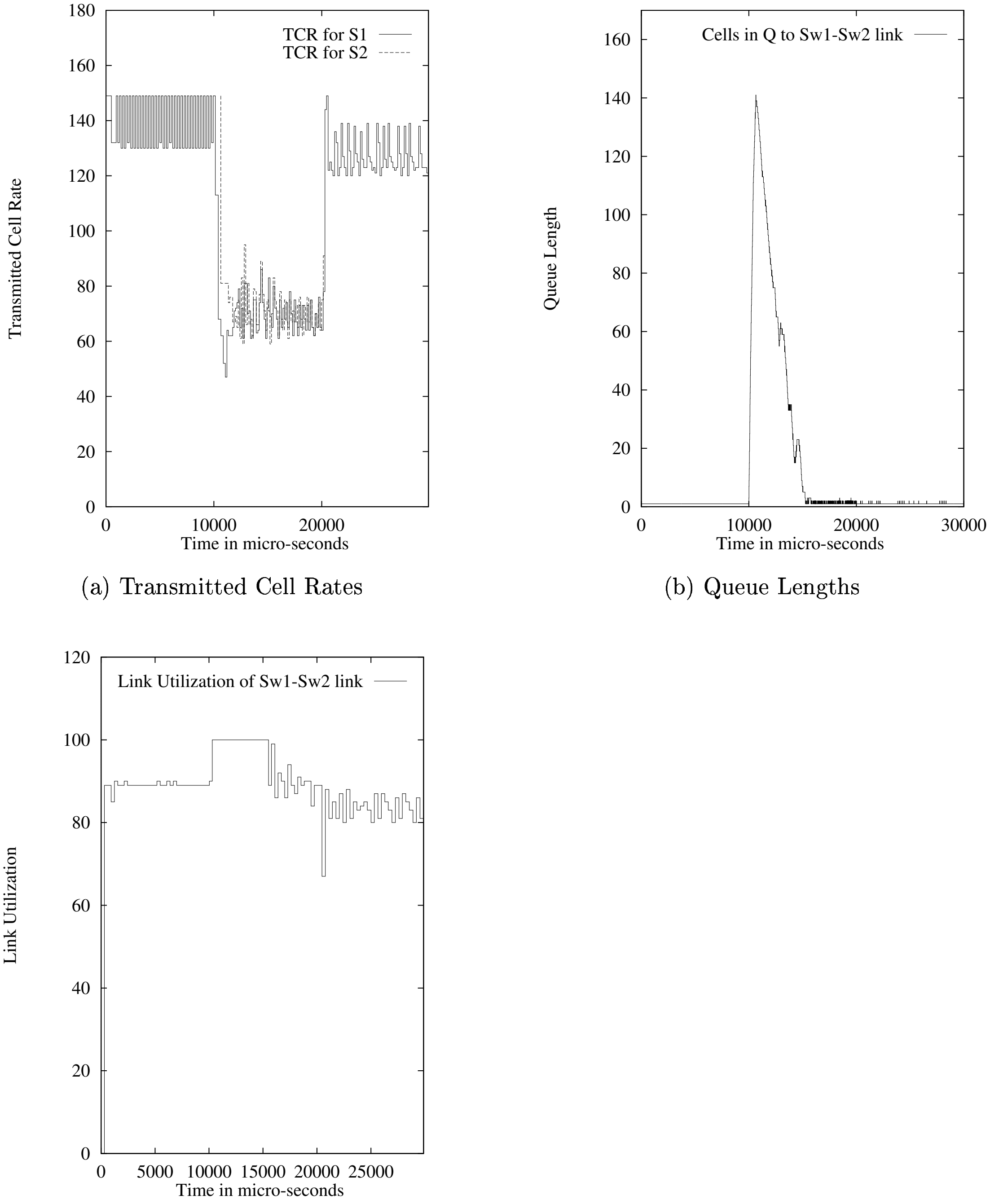


(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 5.15: Simulation results for the three-source configuration



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 5.16: Simulation results for the transient experiment

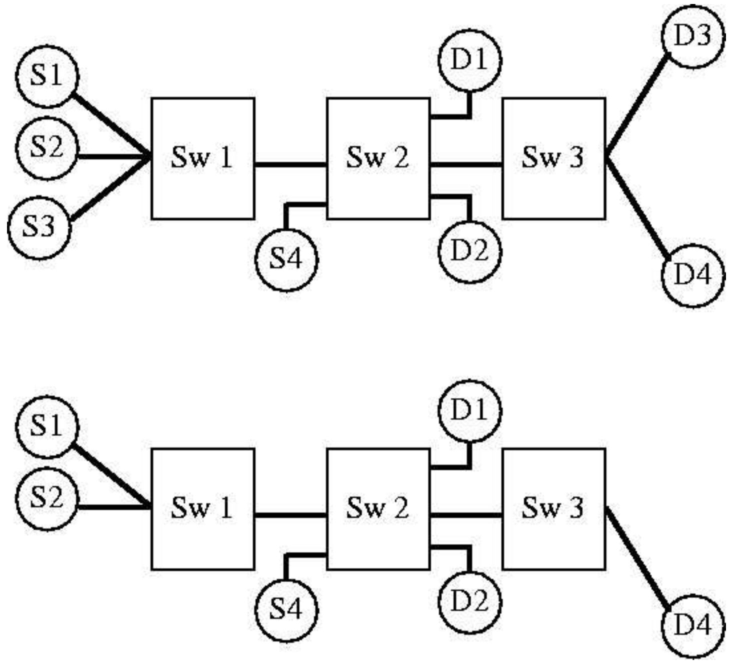
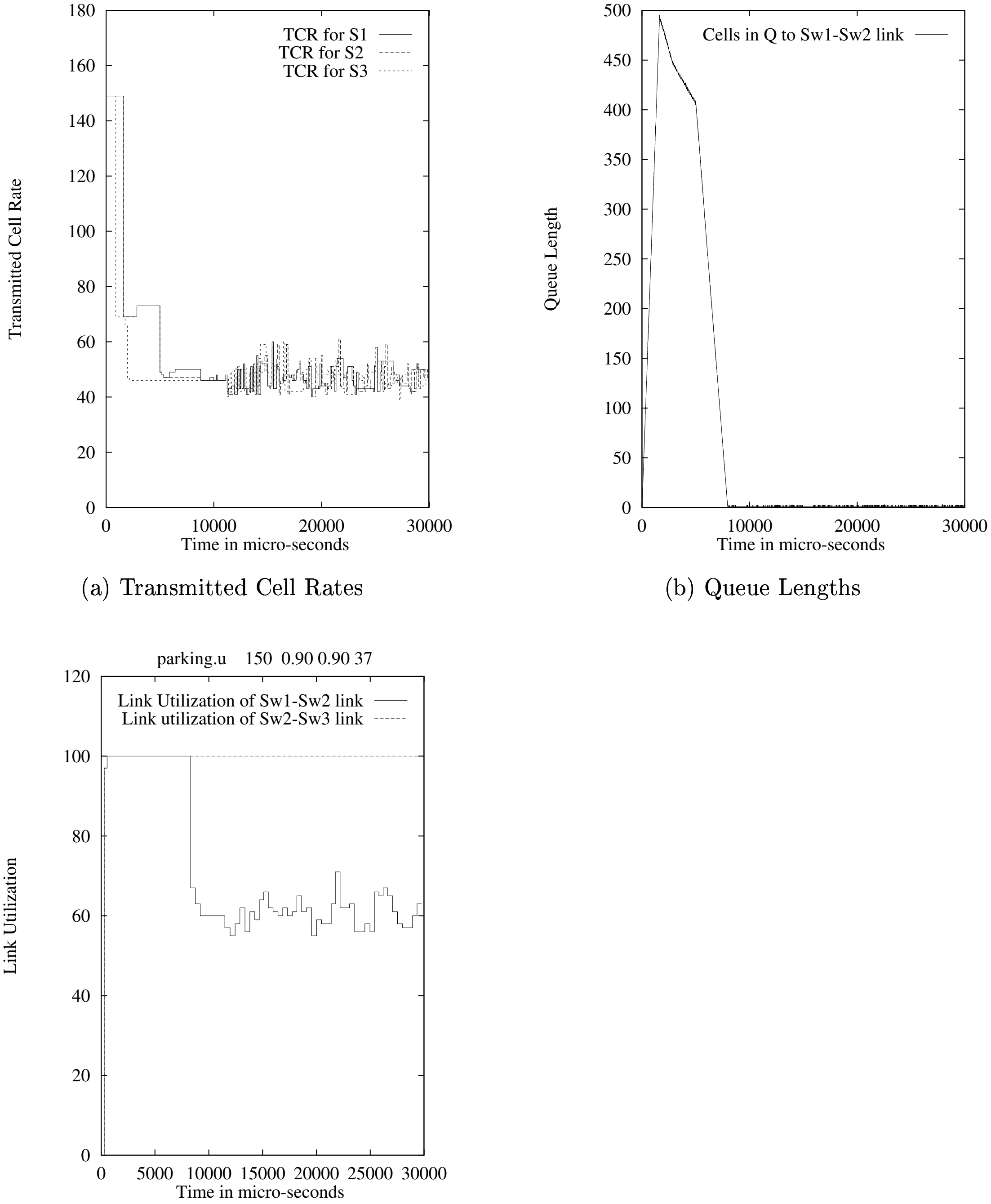


Figure 5.17: The parking lot fairness problem. All users should get the same throughput regardless of the parking area used.

results for this configuration. In particular, the TCR for VC2 and VC4 are shown. Notice that VC4 does get the remaining bandwidth.

5.6 Results for WAN Configuration

The results presented so far assumed link lengths of 1 km. The scheme works equally well for longer links. We have simulated all configurations with 1000 km links as well. Figures 5.21 shows the simulation results for two sources WAN configuration with transient.



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 5.18: Simulation results for the parking lot configuration

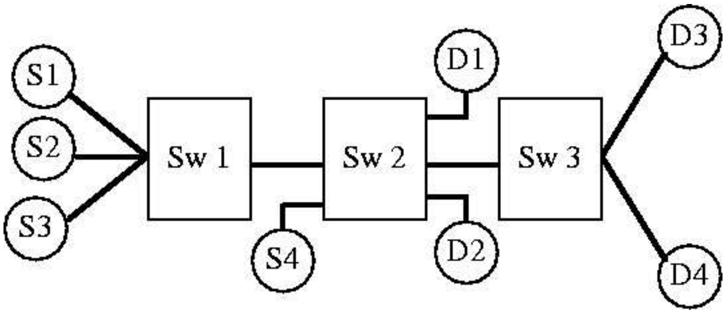
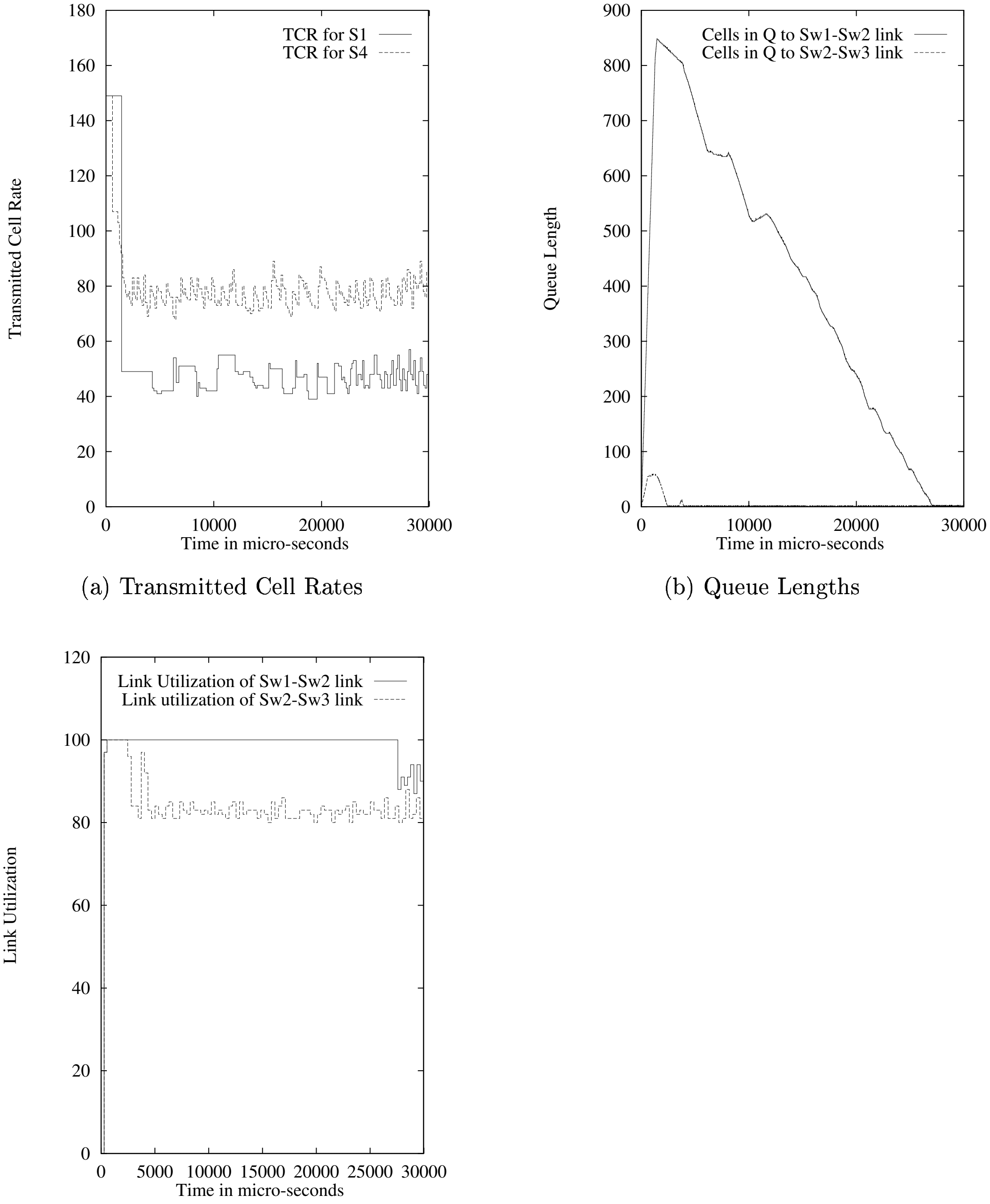


Figure 5.19: Network configuration with upstream bottleneck.

5.7 Results with Packet Train Workload

The most commonly used traffic pattern in congestion simulations is the so called "infinite source model." In this model, all sources have cells to send at all times. It is a good starting configuration because, after all, we are comparing schemes for overload and if a scheme does not work for infinite source it is not a good congestion scheme. In other words, satisfactory operation with infinite source model is necessary. However, it is not sufficient. We have found that many schemes work for infinite source models but fail to operate satisfactorily if the sources are bursty, which is usually the case.

In developing the OSU scheme, we used a packet train model to simulate bursty traffic [64]. A packet train is basically a "burst" of k cells (probably consisting of segments of an application PDU) sent instantaneously by the host system to the adapter. In real systems, the burst is transfered to the adapter at the system bus rate which is very high and so simulating instantaneous transfers is justified. The adapter outputs all its cells at the link rate or at the rate specified by the network in case of rate feedback schemes. If the bursts are far apart, the resulting traffic on the link will look like trains of packets with a gap between trains.

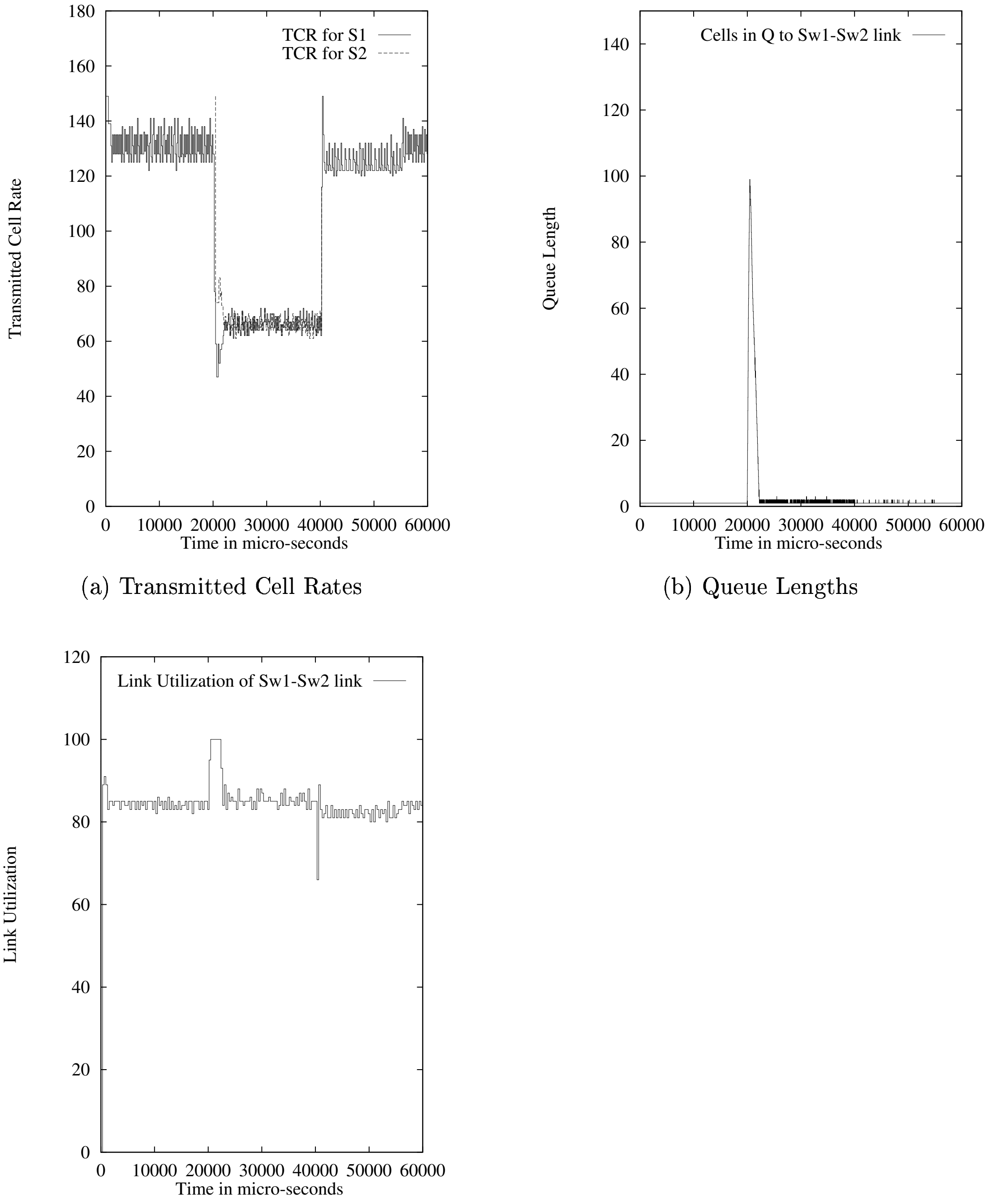


(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 5.20: Simulation results for the upstream bottleneck configuration



(a) Transmitted Cell Rates

(b) Queue Lengths

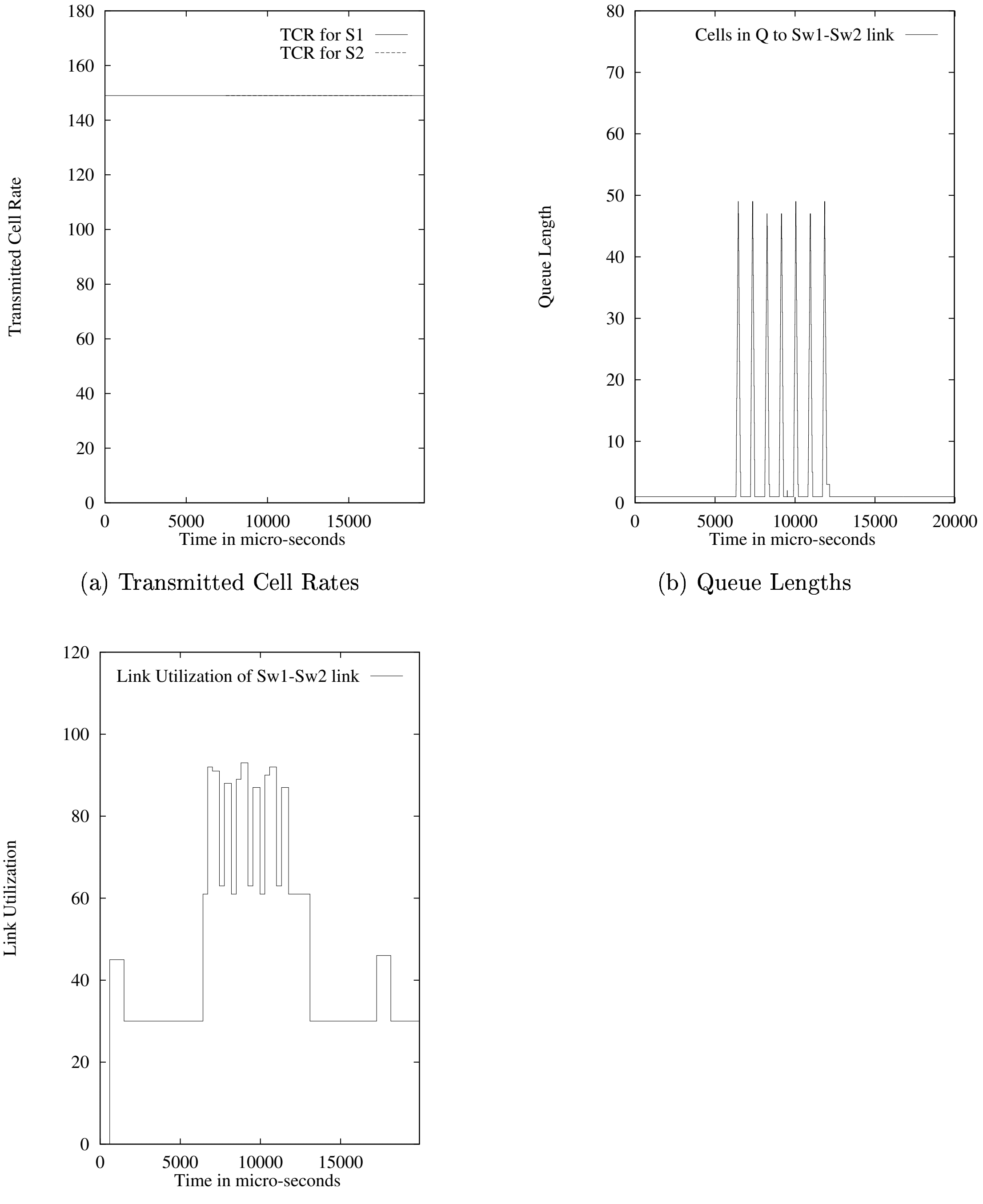
(c) Link Utilization

Figure 5.21: Simulation results for the transient configuration with 1000 km inter-switch links

The key question in simulating the train workload is what happens when the adapter queue is full? Does the source keep putting more bursts into the queue or stops putting new bursts until permitted. We resolve this question by classifying the application as continuous media (video, etc) or interruptible media (data files). In a real system, continuous media cannot be interrupted and the cells will be dropped by the adapter when the network permitted rate is low. With interruptible media, the host stops generating new PDUs until permitted to do so by the adapter. We are simulating only interruptible packet trains for ABR traffic.

For interruptible packet trains, the intertrain gap is governed by a statistical distribution such as exponential. We use a constant interval so that we can clearly see the effect of the interval. In particular, we use one-third duty cycle, that is, the time taken to transmit the burst at the link rate is one-third of the inter-burst time. In this case, unless there are three or more VCs, the sources can not saturate the link and interesting effects are seen with some schemes. In real networks, the duty-cycle is very small of the order of 0.01; the inter-burst time may be of the order of minutes and the burst transmission time is generally a fraction of a second. To simulate overloads with such sources would require hundreds of VCs. That is why we selected a duty cycle of 1/3. This allows us to study both underload and overload with a reasonable number of VCs. We used a burst of 50 cells to keep the simulation times reasonable.

Figures 5.22 and 5.23 show simulation results for the transient and the upstream bottleneck configurations using the packet train model.

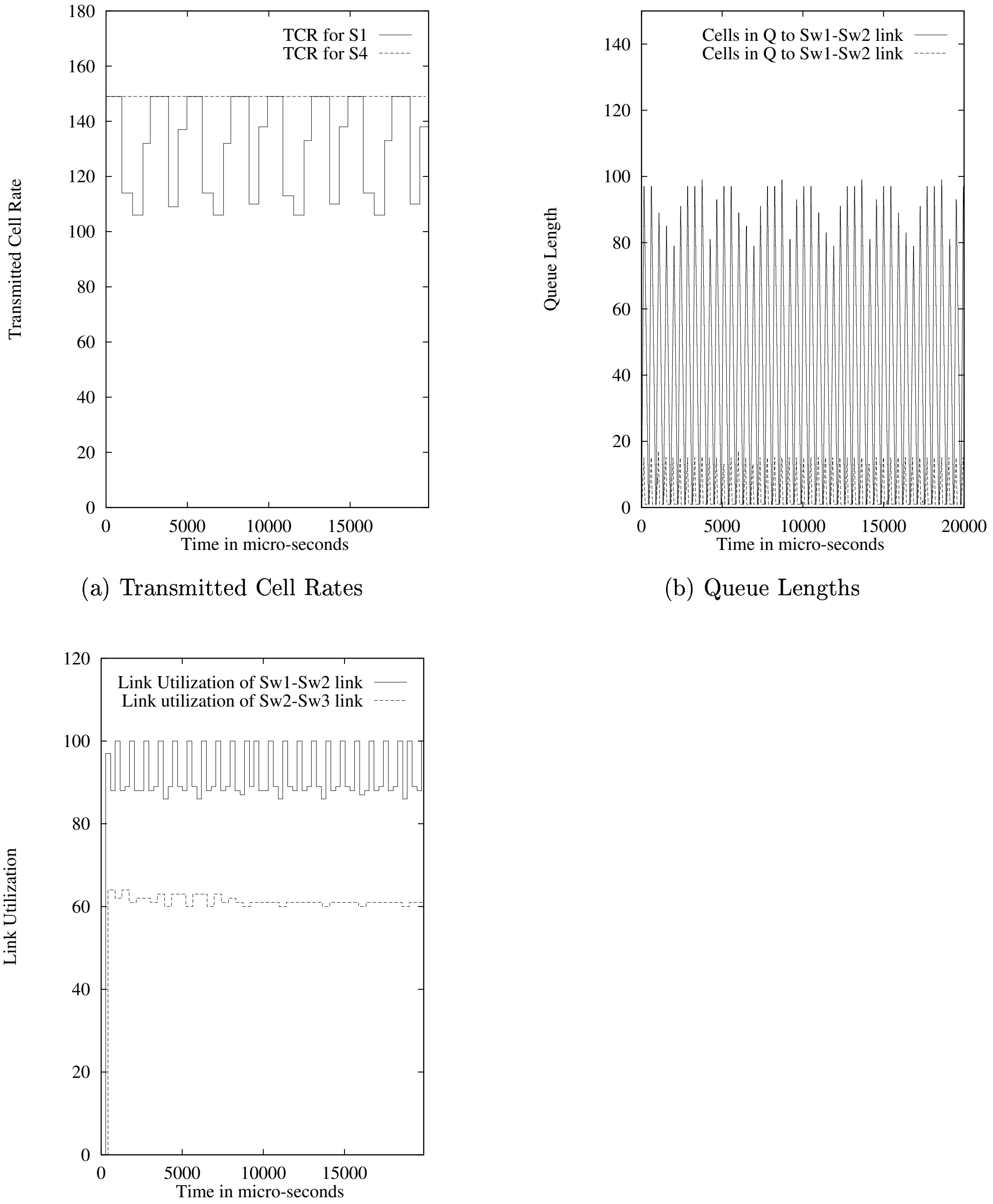


(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 5.22: Simulation results for the transient configuration with packet train workload.



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 5.23: Simulation results for the upstream bottleneck configuration with packet train workload.

5.8 Proof: Fairness Algorithm Improves Fairness

In this section we analytically prove two claims about the simple fairness (TUB) algorithm:

C1. Once inside TUB, the fairness algorithm keeps the link in TUB.

C2. With the fairness algorithm, the link converges towards fair operation.

Our proof methodology is similar to that used in Chiu and Jain (1989)[21], where it was proven that multiplicative decrease and additive increase are necessary and sufficient for achieving efficiency and fairness for the DECbit scheme.

Consider two sources sharing a link of unit bandwidth. Let

x = Input rate of source 1
y = input rate of source 2
z = Load level of the link = $x + y$
U = Target utilization
Δ = Half-width of the target utilization band
s = Fair share rate = U/2

When $x + y = U$, the link is operating efficiently. This is shown graphically by the straight line marked "Efficiency line" in Figure 5.24(a). When $x = y$, the resource allocation is fair. This represents the straight line marked "Fairness line" in the figure. The ideal goal of the load adjustment algorithm is to bring the resource allocations from any point in the two dimensional space to the point marked "Goal" at the intersection of the efficiency and fairness line.

When the network is operating in a region close to the efficiency line, we consider the network to be operating efficiently. This region is bounded by the lines corresponding to $x + y = U(1 - \Delta)$ and $x + y = U(1 + \Delta)$ are in Figure 5.24(a). The quadrangular region bounded by these two lines and the x and y axes is the efficient operation zone also called the target utilization band (TUB). The TUB is described by the four conditions: $x > 0$ and $y > 0$ and $U(1+\Delta) \geq x+y \geq U(1-\Delta)$ Observe that x and y are strictly greater than zero. The case of $x = 0$ or $y = 0$ reduces the number of sources to one.

Similarly, when the network is operating in a region close to the fairness line, we consider the network to be operating fairly. This region is bounded by the lines corresponding to $y = x(1-\Delta)/(1+\Delta)$ and $y = x(1+\Delta)/(1-\Delta)$. The quadrangular region bounded by these two lines in side the TUB is called the fairness region. This is shown in Figure 5.24(b). Mathematically, the conditions defining the fairness region are:

$$\frac{(1+\Delta)}{(1-\Delta)}x \geq y \geq \frac{(1-\Delta)}{(1+\Delta)}x \tag{5.1}$$

$$U(1+\Delta) \geq x+y \geq U(1-\Delta) \tag{5.2}$$

The fair share s is $U/2$. Recall that the TUB algorithm sets the load adjustment factor (LAF) as follows:

IF $(x < s)$ THEN LAF $= \frac{z}{1+\Delta}$ ELSE LAF $= \frac{z}{1-\Delta}$

The rate x is divided by the LAF at the source to give the new rate x'. In other words,

$x' = x\frac{1+\Delta}{z}$ if $x < s$ and $x\frac{1-\Delta}{z}$ otherwise.

5.8.1 Proof of Claim C1

To prove claim C1, we introduce the lines $x = s$ and $y = s$ and divide the TUB into four non-overlapping regions as shown in Figure 5.25(a). These regions correspond to the following inequalities:

Region 1: $s > x > 0$ and $y \geq s$ and $U(1+\Delta) \geq x+y \geq U(1-\Delta)$

Region 2: $y \geq s$ and $x \geq s$ and $U(1+\Delta) \geq x+y$

Region 3: $s > y > 0$ and $x \geq s$ and $U(1+\Delta) \geq x+y \geq U(1-\Delta)$

Region 4: $y < s$ and $x < s$ and $x+y \geq U(1-\Delta)$

In general, triangular regions are described by three inequalities, quandrangular regions by four inequalities and so on.

Proof for Region 1

Consider a point (x, y) in the quadrangular region 1. It satisfies the conditions: $x > 0$ and $y \geq s$ and $U(1+\Delta) \geq x+y \geq U(1-\Delta)$. The link is operating at a load level z given by:

$z = \frac{x+y}{U}$ or $y = Uz - x$

Since (x, y) is in the TUB, we have: $(1+\Delta) \geq z \geq (1-\Delta)$. According to the TUB algorithm, given that $x < s = U/2$ and $y \geq s = U/2$, the system will move the two sources from the point (x, y) to the point $(x', y') = (\frac{x(1+\Delta)}{z}, \frac{y(1-\Delta)}{z})$.

$$\begin{aligned} x' + y' &= \frac{x(1+\Delta) + y(1-\Delta)}{z} && (5.3) \\ &= U(1+\Delta) - \frac{2x\Delta}{z} && (5.4) \\ &= U(1-\Delta) + \frac{2\Delta}{z} y && (5.5) \\ & && (5.6) \end{aligned}$$

The quantity on the left hand side of the above equation is the new total load. Since the last terms of equations 5.4 and 5.5 are both positive quantities, the new total load is below $U(1+\Delta)$ and above $U(1-\Delta)$. In other words, the new point is in TUB. This proves that claim C1 holds for all points in region 1.

Proof for Region 2

Points in the triangular region 2 satisfy the conditions: $y \geq s$, $x \geq s$, and $x+y \leq U(1+\Delta)$

In this region, both x and y are greater than or equal to the fair share $s = U/2$. Therefore, the new point is given by : $(x', y') = (\frac{x(1-\Delta)}{z}, \frac{y(1-\Delta)}{z})$.Hence,

$$x' + y' = \frac{x(1-\Delta) + y(1-\Delta)}{z} = \frac{(x+y)(1-\Delta)}{z} = \frac{Uz(1-\Delta)}{z} = U(1-\Delta)$$

This indicates that the new point is on the lower line of the TUB (which is a part of the TUB) This proves claim C1 for all points in region 2.

The proof of claim C1 for regions 3 and 4 is similar to that of regions 1 and 2, respectively.

5.8.2 Proof of Claim C2

We show convergence to the fairness region (claim C2) as follows. Any point in the fairness region remains in the fairness region. Further, any point (x, y) in the TUB but not in the fairness region moves towards the fairness region at every step. Consider the line L joining the point (x, y) to the origin $(0, 0)$ as shown in Figure 5.25(a). As the angle between this line and the fairness line ($x = y$) decreases, the operation becomes fairer. We show that in regions outside the fairness zone, the angle between the line L and the fairness line either decreases or remains the same. If the angle remains the same, the point moves to a region where the angle will decrease in the subsequent step.

We introduce four more lines to Figure 5.25(a). These lines correspond to $y = (1+\Delta)x$, $y = (1-\Delta)x$, $y = \frac{(1-\Delta)}{(1+\Delta)}x$ and $y = \frac{(1+\Delta)}{(1-\Delta)}x$. This results in the TUB being divided into eight non-overlapping regions as shown in Figure 5.25(b). The new regions are described by the conditions:

Region 1a: $s > x > 0$ and $y \geq s$ and $U(1+\Delta) \geq x+y \geq U(1-\Delta)$ and $y > (1+\Delta)x$

Region 1b: $s > x$ and $(1+\Delta)x \geq y \geq s$

Region 2: $y \geq s$ and $x \geq s$ and $U(1+\Delta) \geq x+y$

Region 3a: $s > y > 0$ and $x \geq s$ and $U(1+\Delta) \geq x+y \geq U(1-\Delta)$ and $y < (1-\Delta)x$

Region 3b: $s > y \geq (1-\Delta)x$ and $x \geq s$

Region 4a: $y < s$ and $x < s$ and $x+y \geq U(1-\Delta)$ and $y \leq \frac{(1+\Delta)}{(1-\Delta)}x$ and $y \geq \frac{(1-\Delta)}{(1+\Delta)}x$

Region 4b: $y < s$ and $x+y \geq U(1-\Delta)$ and $y > \frac{(1+\Delta)}{(1-\Delta)}x$

Region 4c: $x < s$ and $x+y \geq U(1-\Delta)$ and $y < \frac{(1-\Delta)}{(1+\Delta)}x$

The regions 1a and 1b are subdivisions of region 1 in Figure 5.25(a). Similarly, regions 3a and 3b are subdivisions of region 3, and regions 4a, 4b, and 4c are subdivisions of region 4 in Figure 5.25(a) respectively. Observe that regions 1b, 2, 3b and 4a are completely contained in the fairness region.

Proof for Region 1a

Hexagonal region 1a is defined by the conditions: $s > x > 0$ and $y \geq s$ and $U(1+\Delta) \geq x+y \geq U(1-\Delta)$ and $y > (1+\Delta)x$. The new point is given by: $(x', y') = (\frac{x(1+\Delta)}{z}, \frac{y(1-\Delta)}{z})$. Hence,

$$\frac{y'}{x'} = \frac{y}{x} \times \frac{1-\Delta}{1+\Delta} \tag{5.7}$$

Since Δ is a positive non-zero quantity, the above relation implies:

$$\frac{y'}{x'} < \frac{y}{x} \tag{5.8}$$

Further since y/x is greater than $1+\Delta$, equation 5.7 also implies:

$$\frac{y'}{x'} > (1-\Delta) \tag{5.9}$$

Equation 5.8 says that the slope of the line joining the origin to new point (x', y') is lower than that of he line joining the origin to (x, y). While equation 5.9 says that the new point does not overshoot the fairness region. This proves Claim C2 for all points in region 1a.

Proof for Region 1b

Triangular region 1b is defined by the conditions: $s > x$ and $(1+\Delta)x \geq y \geq s$. Observe that region 1b is completely enclosed in the fairness region because it also satisfies the conditions 5.1 and 5.2 defining the fairness region.

To prove claim C2, we show that the new point given by $(x', y') = (\frac{x(1+\Delta)}{z}, \frac{y(1-\Delta)}{z})$ remains in the fairness region.

Since (x, y) satisfies the conditions $1 < y/x \leq (1+\Delta)$, we have:

$$\frac{1-\Delta}{1+\Delta} < \frac{y'}{x'} \leq (1-\Delta) \tag{5.10}$$

Condition 5.10 ensures that the new point remains in the fairness region defined by conditions 5.1 and 5.2.

This proves Claim C2 for all points in region 1b.

Proof of claim C2 for region 3a and 3b is similar to that of regions 1a and 1b, respectively.

Proof for Region 2

Triangular region 2 is defined by the conditions: $y \geq s$ and $x \geq s$ and $x + y \leq U(1 + \Delta)$. This region is completely enclosed in the fairness region. The new point is given by:

$$x' = \frac{x(1 - \Delta)}{z} \text{ and } y' = \frac{y(1 - \Delta)}{z}$$

Observe that:

$$\frac{y'}{x'} = \frac{y}{x} \text{ and } x' + y' = \frac{(x + y)(1 - \Delta)}{z} = U(1 - \Delta)$$

That is, the new point is at the intersection of the line joining the origin and the old point and the lower boundary of the TUB. This intersection is in the fairness region. This proves Claim C2 for all points in region 2.

Proof for Region 4

Triangular region 4 is defined by the conditions: $y < s$ and $x < s$ and $x + y \geq U(1 - \Delta)$. The new point is given by:

$$x' = \frac{x(1 + \Delta)}{z} \text{ and } y' = \frac{y(1 + \Delta)}{z}$$

Observe that:

$$\frac{y'}{x'} = \frac{y}{x} \text{ and } x' + y' = \frac{(x + y)(1 + \Delta)}{z} = U(1 + \Delta)$$

That is, the new point is at the intersection of the line joining the origin and the old point and the upper boundary of the TUB.

As shown in Figure 5.25(b), region 4 consists of 3 parts: 4a, 4b, and 4c. All points in region 4a are inside the fairness region and remain so after the application of the TUB algorithm. All points in region 4b move to region 1a where subsequent applications of TUB algorithm will move them towards the fairness region. Similarly, all points in region 4c move to region 3a and subsequently move towards the fairness region.

This proves claim C2 for region 4.

5.8.3 Proof for Asynchronous Feedback Conditions

We note that our proof has assumed the following conditions:

- Feedback is given to sources instantaneously.
- Feedback is given to sources synchronously.
- There are no input load changes (like new sources coming on) during the period of convergence
- The analysis is for the bottleneck link (link with the highest utilization).
- The link is shared by unconstrained sources (which can utilize the rate allocations).

It may be possible to relax one or more of these assumptions. However, we have not verified all possibilities. In particular, the assumption of synchronous feedback can be relaxed as shown next.

In the previous proof, we assumed that the operating point moves from (x, y) to (x', y'). However, if only one of the sources is given feedback, the new operating point could be (x, y') or (x', y). This is called asynchronous feedback.

The analysis procedure is similar to the one shown in the previous sections. For example, consider region 1 of Figure 5.25(a). If we move from (x, y) to (x, y'), we have:

$$y' = \frac{y(1-\Delta)}{z}$$

and

$$\begin{aligned}
x + y' &= \frac{xz + y(1-\Delta)}{z} && (5.11)\\
&= U(1-\Delta) + \frac{x\{z - (1-\Delta)\}}{z} && (5.12)\\
&= U(1+\Delta) - \frac{x\{(1+\Delta) - z\} + 2y\Delta}{z} && (5.13)\\
& && (5.14)
\end{aligned}$$

Since, the last terms of equations 5.12 and 5.13 are both positive, the new point is still in the TUB. This proves Claim C1.

Further, we have:

$$\frac{y'}{x} = \frac{y}{x}(1-\Delta)$$

Therefore,

$$\frac{y'}{x} < \frac{y}{x} \text{ and } \frac{y'}{x} \geq (1-\Delta)$$

That is, the slope of the line joining the operating point to the origin decreases but does not overshoot the fairness region.

Note that when $z = 1-\Delta$, $y' = y$. That is, the operating point does not change. Thus, the points on the lower boundary of the TUB ($x + y = U(1-\Delta)$) do not move, and hence the fairness for these points does not improve in this step. It will change only in the next step when the operating point moves from (x, y') to (x', y').

The proof for the case (x', y) is similar. This completes the proof of C1 and C2 for region 1. The proof for region 3 is similar.

5.9 Current Traffic Management Specifications vs OSU Scheme

In the previous sections, we have mentioned several features of the OSU scheme that have either been adopted in the standard or have been commonly implemented. In this section, we describe two features that were not adopted.

In the OSU scheme, the sources send RM cells every T microseconds. This is the time-based approach. A count-based alternative is to send RM cells after every n data cells. We argued that the time-based approach is more general. It provides the same feedback delay for all link speeds and source rates.

The ATM forum has adopted the count-based approach mainly because it guarantees that the overhead caused by RM cells will be a fixed percentage (100/n)% of the total load on the network.

The disadvantage with the count-based approach is that if there are many low-rate sources, it will take a long time to control them since the inter-RM cell times will be large. The time-based approach uses a fixed bandwidth per active source for RM cell overhead. For many active sources, this could be excessive.

The RM cells in the OSU scheme contain an averaging interval field. The network manager sets the averaging interval parameter for each switch. The maximum of the averaging interval along a path is returned in the RM cell. This is the interval that the source uses to send the RM cells. With the count-based approach, this field is not required.

Another major difference is the indication of rate. The OSU scheme requires sources to present both average and peak rates (along with the averaging interval) in the RM cell. The standard requires only one rate.

The OSU scheme is, therefore, incompatible with the ATM forum's current traffic management standards. Although, it cannot be used directly, most of its features and results can be ported to design compatible schemes. We have upgraded the ideas from the OSU scheme to create the Explicit Rate Indication for Congestion Avoidance (ERICA) scheme [60]. The ERICA scheme which is described later in this dissertation, is also mentioned in the ATM Traffic Management 4.0 standards as a sample switch algorithm.

5.10 Limitations and Summary of the OSU Scheme

This chapter describes an explicit rate based congestion avoidance scheme for ATM networks. The scheme was developed as the ATM Forum traffic management specifications were being developed. While the strengths of the OSU scheme are its choice of congestion indicator, metric, small number of parameters, and O(1) complexity, its limitations are slow convergence for complex configurations, and slight sensitivity to the averaging interval parameter. The following statements apply to the basic OSU scheme.

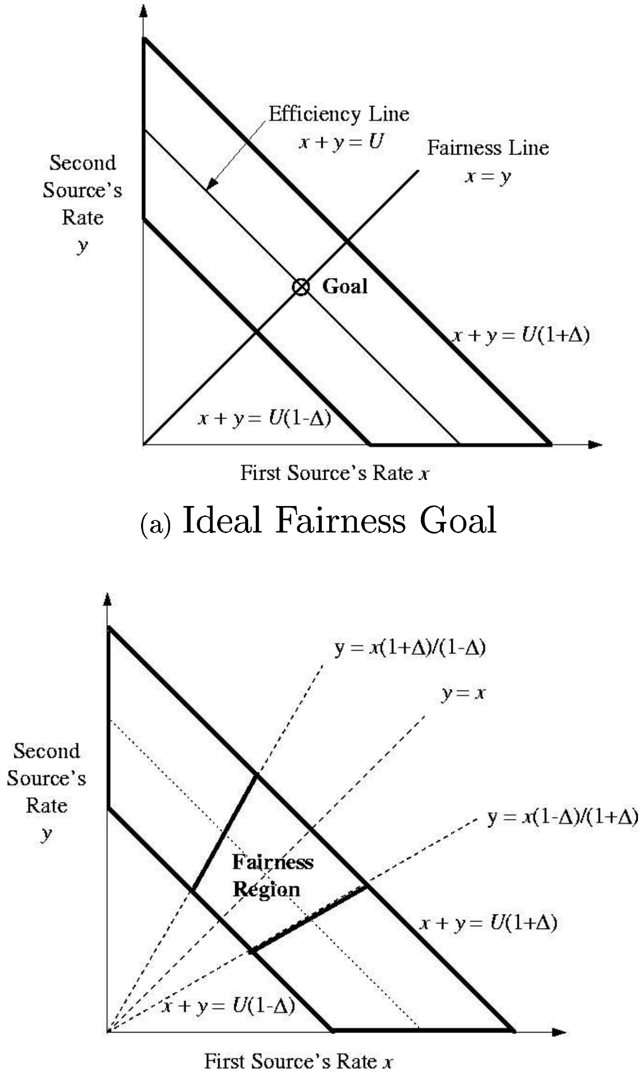
Our proof in section 5.8 is applicable to the bottleneck link (link with the highest utilization) which is shared by unconstrained sources (which can use any given allocation). It assumes that feedback is given to sources instantaneously and synchronously. In the general case, where these assumptions do not hold, the system may take longer to converge to the fair and efficient operating point. If the perturbations to the system (due to VBR, asynchronous feedback, multiple bottlenecks, or rapid changes in source load pattern) are of a time scale smaller than this convergence time, the system may be unstable. This statement is true for the convergence of *any* switch algorithm.

Further, since the scheme is measurement-based, it is slightly sensitive to the averaging interval in the switch. For example, if the number of sources is underestimated, the scheme will attempt to converge to a higher fairshare value and keep moving in and out of the TUB. Note that even then, the bottleneck is maintained at a high utilization level and the excess capacity is used to drain out queues. The number of sources is never overestimated; hence our scheme always achieves efficiency. The second quantity measured in the averaging interval is the current load level, z. If the system is actually overloaded, then the overload is measured correctly in z. However, if the system is underloaded, the averaging interval may not be long enough to exactly measure the underload. In such a case, z may be underestimated, and the system may initially move to an overload region before converging.

Although the scheme itself is no longer strictly compatible with the specifications, many of the results obtained during this research have affected the direction of the specifications. Many features of the scheme are now being commonly used in many switch implementations. A patent on the inventions of this scheme is also pending [77].

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.

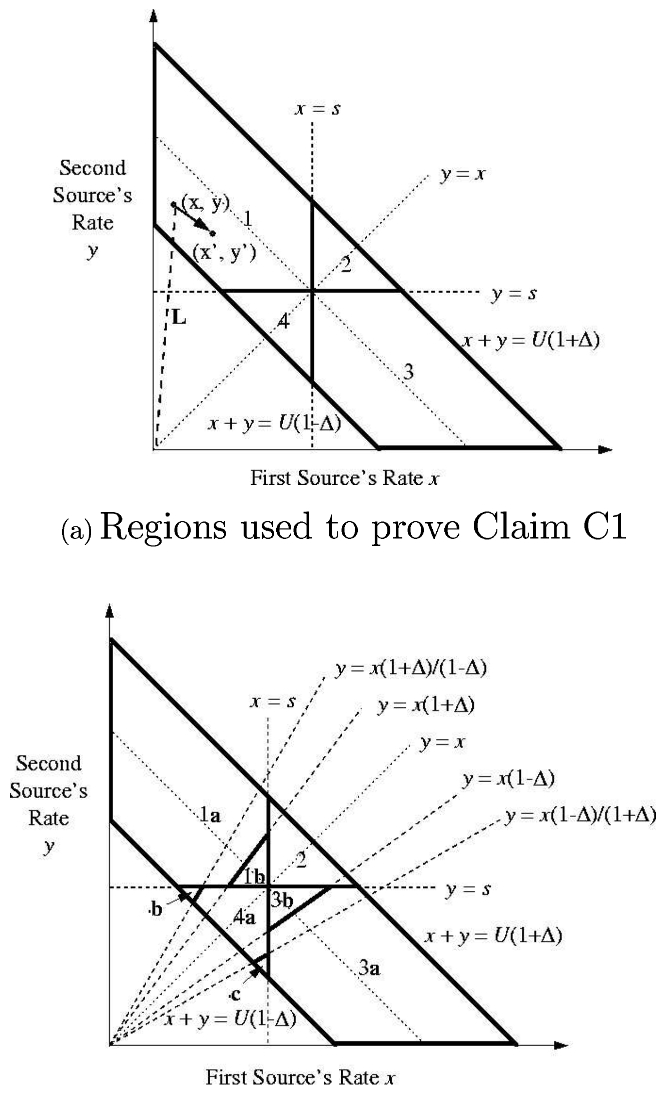
As stated in the previous section, we have developed a new ATM standards compatible algorithm called ERICA. ERICA and its extensions use a new set of algorithms. These algorithms achieve fast convergence and robustness for complex workloads, where input load and capacity may fluctuate arbitrarily. This will be the subject of our future chapters.



(a) Ideal Fairness Goal

(b) The Fairness Region

Figure 5.24: A geometric representation of efficiency and fairness for a link shared by two sources



(a) Regions used to prove Claim C1

(b) Regions used to prove Claim C2

Figure 5.25: Subregions of the TUB used to prove Claims C1 and C2

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.