The OSU Scheme for Congestion Avoidance in ATM Networks Using Explicit Rate Indication[1]

OSU-CIS Technical Report Number: OSU-CISRC-1/96-TR02

Raj Jain, Shiv Kalyanaraman and Ram Viswanathan
Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210-1277
Email: Jain@ACM.Org

Abstract

An explicit rate indication scheme for congestion avoidance in computer and telecommunication networks is proposed. The sources monitor their load and provide the information periodically to the switches. The switches, in turn, compute the load level and ask the sources to adjust their rates up or down. The scheme achieves high link utilization, fair allocation of rates among contending sources and provides quick convergence. A backward congestion notification option is also provided. The conditions under which this option is useful are indicated.

[1] OSU Tech Report OSU-CISRC-1/96-TR02. Available through http://www.cis.ohio-state.edu/~jain

Contents

1 Introduction

The next generation of computer and telecommunication networks will use the asynchronous transfer mode (ATM). ATM networks are connection-oriented networks in which the information is transmitted using fixed size 53-byte cells. The cells flow along predetermined paths called virtual channels (VCs). End systems set up constant bit rate (CBR) or variable bit rate (VBR) virtual channels (VCs) before transmitting information. For data traffic, which is highly bursty and does not have strict delay requirements, it is best to dynamically divide all available bandwidth fairly among VCs that need it at any moment of time. Such traffic is called available bit rate (ABR) traffic.

The main problem in supporting ABR traffic is that it is possible that more traffic may come into a switch then can get out and the switches can get congested. To control congestion, the switches typically inform the sources to reduce the traffic rate using a feedback mechanism. The feedback can consist of a single bit which can take two values 0 or 1 meaning increase or decrease, respectively. This may take several round trips before the sources will adjust to the right rate. A better strategy for connection-oriented networks is for the switches to send an "resource management" (RM) cell to the source containing the rate that it should change to.

Any time the total demand for a resource is more than the available resource, the problem of congestion arises. The bandwidth, buffers, computational capacity are examples of resources in a network. The design goal of most network resource management algorithms is to provide maximum link bandwidth utilization while minimizing the buffers (queue length) and computation overhead.

The OSU scheme is also an explicit rate indication scheme similar to the MIT scheme [11, 12]. However, it does not necessarily require the switches to remember the rates of all VCs. Thus, the minimal storage requirements as well as the computational complexity becomes O(1), that is, the computation or storage does not change as the number of VCs is changed. Also, it uses the exact overload as measured at the switch to determine the allowed rate. The OSU scheme has several other desirable features and design goals that are described later in Section 5 of this paper.

In this report, we have described both the problem and the solutions in terms of ATM networks. However, most of the discussion applies to packet switching networks as well. In particular, if the packets are large, the feedback can be included in the header and the need for special control cells can be avoided.

Each virtual circuit has one source and one destination and passes through a number of switches. Throughout this paper, we have used the term "source" and "virtual circuit" (VC) interchangeably. The term "host" is used to denote an end system, which may have several VCs.

2 Performance Requirements

In order to compare various congestion schemes, it is important to agree on the measures of goodness. Three performance metrics most commonly used for this purpose are efficiency, delay, and fairness. These along with the optimal operation are explained below.

2.1 Optimal Operation

One of the first requirements for good performance is high thoroughput. In a shared environment the throughput for a source depends upon the demands by other sources. The most commonly used criterion for what is the correct share of bandwidth for a source in a network environment is the so called "max-min allocation." It provides the maximum allocation possible to the source receiving the least among all contenting sources. Mathematically, it is defined as follows. Given a configuration with n contenting sources, suppose the ith source gets a bandwidth x_i. The allocation vector $\{x_1, x_2, \ldots, x_n\}$ is feasible if all link load levels are less than or equal to 100%. The total number of feasible vectors is infinite. Given any allocation vector, the source that is getting the least allocation is in some sense, the "unhappiest source." Given the set of all feasible vectors, find the vector that gives the maximum allocation to this unhappiest source. Actually, the number of such vectors is also infinite although we have narrowed down the search region considerably. Now we take this "unhappiest source" out and reduce the problem to that of remaing n-1 sources operating on a network with reduced link capacities. Again, we find the unhappiest source among these n-1 sources, give that source the maximum allocation and reduce the problem by one source. We keep repeating this process until all sources have been given the maximum that they could get.

The following example illustrates the above concept of max-min fairness. Figure 3 shows a network with four switches connected via three 150 Mbps links. Four VCs are setup such that the first link L1 is shared by sources S1, S2, and S3. The second link is shared by S3 and S4. The third link is used only by S4. Let us divide the link bandwidths fairly among contending sources. On link L1, we can give 50 Mbps to each of the three contending sources S1, S2, and S3. One link L2, we would give 75 Mbps to each of the sources S3 and S4. On link L3, we would give all 155 Mbps to source S4. However, source S3 cannot use its 75 Mbps share at link L2 since it is allowed to use only 50 Mbps at link L1. Therefore, we give 50 Mbps to source S3 and construct a new configuration shown in Figure 4, where Source S3 has been removed and the link capacities have been reduced accordingly. Now we give 1/2 of the link L1's remaining capacity to each of the two contending sources: S1 and S2; each gets 50 Mbps. Source S4 gets the entire remaining bandwidth (100 Mbps) of link L2. Thus, the fair allocation vector for this configuration is (50, 50, 50, 100). This is the max-min allocation.

Notice that max-min allocation is both fair and efficient. It is fair in the sense that all sources get an equal share on every link provided that they can use it. It is efficient in the sense that each link is utilized to the maximum load possible.

2.2 Fairness

The max-min allocation is the desired goal. Any scheme that results in max-min allocation is called max-min fair. If a scheme gives an allocation that is different from the max-min allocation, its unfairness is quantified as follows.

Suppose a scheme allocates $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n\}$ instead of the max-min allocation $\{\hat{x}_1, \hat{x}_2, ..., \hat{x}_n\}$. Then, we calculate the normalized allocations $x_i = \tilde{x}_i/\tilde{x}_i$ for each source and compute the fairness index as follows [7, 3]:

$$\text{Fairness} = \frac{\left(\sum_i x_i\right)^2}{n \sum_i x_i^2}$$

Since allocations x_i's usually vary with time, the fairness can be plotted as a function of time. Alternatively, throughputs over a given interval can be used to compute overall fairness.

2.3 Efficiency

The efficiency of a scheme relates to its making full use of its resources. A scheme that results in underload or overload is considered inefficient. Given a network, it is the bottleneck link (the link with maximum utilization) whose proper loading is important. Thus, a efficient scheme tries to control sources such that the bottleneck link is neither underloaded nor overloaded.

2.4 Delay

Given two schemes with the same fairness and efficiency, one with lower end-to-end delay is preferred. Generally, there is a tradeoff between efficiency and delay in the sense that if one tries to use a link to 100% capacity, the queue lengths may become too large and the delays may become excessive. While data traffic is generally delay insensitive, extremely large delays are harmful since they may result in timeouts at higher layers and result in unnecessary retransmissions. Therefore, it is often preferable to keep link utilizations below 90-95%.

2.5 Fast Convergence

Most practical schemes take some time to reach fair and efficient operating point. Given two schemes with the same fairness and efficiency at the end of simulation, we would prefer one which achieve efficiency and fairness faster. We used this preference to compare different design alternatives. Given the same starting point, we compared the time taken to reach steady state and the alternative that produced faster convergence was selected. The steady state is defined informally as a small region around the final operating point. With deterministic simulations, it is easy to identify the steady state since the system starts to oscillate around the final point.

3 Survey of Other Schemes

The problem of congestion control has been known to be the critical part of network architecture design for several decades and hundreds of papers have been written on various schemes. Rather than give a survey of all schemes, we intend to concentrate here on schemes that are (or were) leading candidates for adoption in ATM networks. At ATM Forum, which is an organization of over 400 computer and telecommunication equipment manufacturers, the traffic management subgroup is responsible for finding the right congestion control scheme. In particular, the members have been discussing the congestion control for the so called "available bit rate (ABR)" traffic since May 1993. By September of 1993, two distinct approaches emerged: The credit based and the rate based.

3.1 The Credit-Based Approach

The credit-based approach consists of using window (or credit) based flow control on every link. Each node (switch or the source) keeps a separate queue for each VC. At each hop, the receiving node tells the transmitting node how many cells it can send for each VC. The number of cells that can be transmitted is called "credits". The number of cells received are carefully monitored so that lost cells can be detected. This approach has a potential to provide full link utilization and guarantee zero loss due to congestion. However, this scheme requires per-VC queueing, per-VC service, and per-VC monitoring. The number of VCs that exist at any time is large and, therefore, per-VC operations are considered undesirable by most switch manufacturers. They would prefer to keep all per-VC operations (except switching) at the end systems. The complexity and cost of implementation has been the main objection to this approach. The vendors are not willing to pay the high cost of per-VC operations for the noble goal of "zero loss." They would rather take the small probability of loss particularly if it results in considerable savings in cost.

3.2 The Rate-Based Approach

This approach is based on end-to-end rate control using feedback from the network. Initially, a backward explcit congestion notification (BECN) method was proposed However, it was dropped in favor the forward explicit congestion notification (FECN). In either case, the cells contain a single bit which is marked by the switches if they are congested. In FECN, the destination end station monitors these bits and sends a control cell back to the source asking it to adjust the rate up or down. In the BECN version, the congested switches directly send the control cell to the source (and the bit is actually not required).

3.2.1 PRCA

A sequence of FECN schemes have been proposed at the Forum. The latest one is called the Proportional Rate Control Algorithm (PRCA) [13]. In this proposal, the sources would set the FECN bit to one except in every nth cell (where n is a parameter). The switches set to bit to one when they are congested (and do nothing if not congested). If the destination receives a cell with FECN bit set to zero, it concludes that the network is not congested and sends a control cell to the source asking it to increase its rate. The sources continually decrease their rates (after sending each cell) unless they receive the control cell from the destination. A multiplicative decrease and additive increase is used to achieve fairness.

3.2.2 Explicit Rate Indication

The single-bit feedback, while OK for window-based schemes is too slow for rate-based schemes. In window-based scheme, if the control is slow to change (and therefore remains constant for a while), the queue length cannot exceed the specified window size. This is not true for rate-based schemes. If the rate is over the optimal even by a small amount, the queues will keep building, leading to overflow and cell loss. It is important to measure the rate fast and let the sources know about it as soon as possible. This argument lead to the following two explicit rate indication proposals at the ATM Forum meeting of July 1994.

3.2.3 The MIT Scheme

This scheme, developed at the Massachusetts Institute of Technology, consists of the sources periodically sending their rates to switches in control cells. The switches reduce the rate value if necessary. The cells are returned to the source by the destination node.

The control cells contain a “Reduced bit” and the source’s “Desired rate.” Each switch monitors its traffic and calculates its available capacity per VC. This quantity is called the “fair share.”

If the “desired rate” is higher than or equal to the “fair share,” the desired rate is reduced to the “fair share” and the reduced-bit is set. If the desired rate is less than the fair share, the switch does not change the fields of the control cell.

The destination sends the control cell back to the source. If the souce finds the reduced bit set, it adjusts its rate to that returned in the “desired rate” field of the control cell. Next time, the source sends this new rate in the next control cell transmitted. If the reduced bit is clear, the source can increase its rate but it must first determine how much it can go up by sending a control cell with a higher desired rate.

The switches maintain a list of all of its VCs and their last seen desired rates. All VCs whose desired rate is higher than the switch’s fair share are considered “overloading VCs.” Similarly, VCs with desired rate below the fair share are called “underloading VCs.” The underloading VCs are bottlenecked at some other switch and, therefore, cannot use additional capacity at this switch even if available.

The capacity unused by the underloading VCs is divided equally among the overloading VCs. Thus, the fair share of the VCs is calculated as follows:

$$\text{Fair Share} = \frac{\text{Capacity} - \sum \text{Bandwidth of underloading VCs}}{\text{total number of VCs} - \text{Number of underloading VCs}}$$

It is possible that that after this calculation some VCs that were previously underloading with respect to the old fair share can become overloading with respect to the new fair share. In this case these VCs are re-marked as overloading and the fair share is recalculated.

Researchers at the University of California, Irvine modified the MIT scheme slightly [14]. In particular the switch algorithm was simplified. The switch does not remember any VCs rate. Instead, it computes an exponentially weighted average of the declared desired rates and uses the average as a fair share. The weighting coefficient used for averaging is different during overload and during underload. The MIT scheme requires an O(n) computation in the sense that the number of instructions to compute fair share increase linearly with the number of VCs. The UCI modification makes it order 1, O(1), in the sense that the computational overhead to process a control cell does not depend upon the number of VCs. However, its ability to achieve efficient and fair operation remains to be shown.

The use of exponentially weighted average of ”desired rates” as the fair share does not seem meaningful. First of all ”desired rates” may not be close to the actual transmission rates. Secondly, any average is meaningful only if the quantities are related and close to each other. The desired rates of various sources can be far apart. Thirdly, the exponentially weighted average may become biased towards higher rates. For example, consider two sources running at 1000 Mbps and 1 Mbps. In any given interval, the first source will send 1000 times more control cells than the second source and so the exponentially weighted average is very likely to be 1000 Mbps regardless of the value of the weight used for computing the average.

4 The OSU Scheme

The OSU scheme also requires sources to monitor their load and periodically send control cells that contain the load information. The switches monitor their own load and using it in combination with the information provided in the control cells, compute a factor by which the source should go up or down. At the destination, the control cell is simply returned to the source, which then adjusts its rate as instructed by the network. The key difference between OSU and other schemes is in the way, the rates adjustment factor is computed.

4.1 Control-Cell Format

The control cell contains the following the fields:

1. Transmitted Cell Rate (TCR)

2. The Offered Average Cell Rate (OCR) as measured at the source

3. Rate Adjustment Factor
4. Averaging interval
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 10.3 and need not be present if that option is not used. Other fields are explained later in this sections.

4.2 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.

4.2.1 Control-Cell Sending Algorithm

The control cells are sent periodically every T interval. Although it could be done by the cell count, using interval allows the scheme to work on networks with widely varying link speeds. As shown later, the averaging interval used throughout the path should be the same. The network manager sets the averaging interval parameter for each switch. The maximum of the averaging interval along a path is returned in the control cell. This is the interval that the source uses to send the control cells.

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.

4.2.2 Measuring Offered Average Load

Unlike any other scheme proposed so far, each source also measures its own load. The measurment is done over the same averaging interval that is used for sending the control cells. Notice that there are two separate parameters: transmitted cell rate and offered average cell rate. The first is the instantaneous cell rate during burst transmissions. The cells are sent equally spaced in time. 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. 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 6.

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,

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

4.2.3 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.

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 7 presents a flow chart explaining the rate adjustment.

4.3 The Switch Algorithm

The switch algorithm consists of the following components:

1. How to measure the available capacity
2. How to achieve efficiency
3. How to achieve fairness

These issues and others arising from these are discussed next.

4.3.1 Measuring The Current Load

This consists of simply counting the number of cells received during a fixed averaging interval. The interval is set by the network manager. Based on the known capacity of the link, the switch can compute the load level and determine whether it is overloaded or underloaded.

Since running a link at full load generally results in large queues, it is best to target the link utilization at close to but not quite at 100%. To achieve this the network manager selects a target utilization, say 90%. Whenever the input rate is more than 90% of the nominal capacity, the link is said to be overloaded and whenever the utilization is less than 90%, the link is said to be underloaded. The link cell rate when the network is operating at the target utilization is computed:

$$\text{Target Cell Rate} = \frac{\text{Target Utilization} \times \text{Link bandwidth in Mbps}}{\text{Cell size in bits}} \tag{1}$$

The current load level is then given by:

$$\text{Current Load level} = \frac{\text{Number of cells received during the averaging interval}}{\text{Target Cell Rate} \times \text{Averaging Interval}} \tag{2}$$

4.3.2 Achieving Efficiency

To achieve efficiency, all we need is to replace the load adjustment factor in each control cell by the maximum of the the current load level and the load adjustment value already in the cell.

$$\begin{array}{l}\text{Load Adjustment Factor}\leftarrow \\ \max(\text{Load Adjustment Factor in the cell, Current Load Level in this Switch})\end{array} \tag{3}$$

This simple algorithm is sufficient to bring the network to efficient operation within the next round trip. However, the allocation of the available bandwidth among not be fair. To achieve fairness we need to make use of the other information in the control cells as discussed later in Section 4.3.4.

4.3.3 Counting the Number of Active Sources

Like the MIT scheme, the switches in our scheme may also remember the rates declared by various sources and use them in computing the fair share. However, there are two differences. First, the rates declared by the sources are "Offered Average Cell Rates (OCRs)" and not the desired cell rates, which may or may not be related to the actual rates. Secondly, in the simplest version of our scheme rates of all sources are not required. All we need is the number of active sources, which can be counted either by counting the number of sources with non-zero OCRs or by marking 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.

4.3.4 Achieving Fairness

In resource allocation, the top priority is to bring the network to efficient operation. Once the network is operating close to the target utilization, we need to take steps to achieve fairness. The network manager declares a target utilization band (TUB), say, 90±9% or 81% to 99%. Whenever the link utilization is in TUB, the link is said to be operating efficiently. As will be seen later, it is better to express TUB in the U(1±Δ) format, where U is the target utilization level. For example, 90±9% is expressed as $90(1 \pm 0.1)\%$.

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

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

To achieve fairness, we treat the underloading and overloading sources differently. Underloading sources for our scheme are those sources that are using less than the fair share. While overloading sources are those that are using more than the fair share.

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 $z/(1-\Delta)$. Here Δ is the half-width of the TUB.

If the OCR in the control cell is less than the fair share, the load adjustment factor in the cell is changed as follows:

$$\begin{array}{l}\text{Load Adjustment Factor}\leftarrow\\ \max(\text{Load Adjustment Factor in the cell, } \frac{z}{(1+\Delta)})\}\end{array}$$

On the other hand, if the OCR in the control cell is more than the fair share, the load adjustment factor in the cell is adjusted as follows:

$$\begin{array}{l}\text{Load Adjustment Factor}\leftarrow\\ \max(\text{Load Adjustment Factor in the cell, } \frac{z}{(1-\Delta)})\}\end{array}$$

As shown in Appendix A, this algorithm guarantees that the system consistently moves towards more fair operation. Also, once inside the TUB, the network remains in the TUB unless the number of sources or their load pattern changes. In other words, TUB is a "closed" operating region. These statements are true for any value of Δ less than 0.5.

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.

4.3.5 What Load Level Value to Use?

Under highly overloaded conditions, the queues in a system may become long. The control cells may remain in the system for more than one averaging interval and the question arises as to what load level value should be use for efficiency or fairness computation. Should it be the value at the time of control cell arrival or the latest value at the time of control cell departure? The correct answer is: the value at the cell arrival time should be used. This is because the queue state at arrival more accurately reflects the effect of the TCR indicated in the control cell. This is shown in Figure 8. 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.

4.4 The Destination Algorithm

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

4.5 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 is that ATM networks are connection oriented and so 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 pick any initial rate and start transmitting. 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 Unique Features of the OSU scheme

5.1 High Throughput

In the OSU scheme, the bottleneck links utilization remains in the efficient region or the target utilization band (TUB) selected by the network manager. Based on the cost of the bandwidth, the network manager sets the target utilization band for each link. 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 wide TUB results in a faster progress towards fairness. In most cases, a TUB of 90%(1 ± 0.1) is a good choice. This gives a utilization in the range of 81% to 99%.

5.2 Bounded Oscillations

With the OSU scheme, once the network reaches the efficient region, the oscillations in the link utilizations are bounded to be within the TUB. In other rate-based schemes, average utilization levels as low as 30% have been observed for some WAN configurations. This is particularly bad given that WAN links are extremely expensive.

5.3 Minimum Delay

Under steady state, the OSU scheme operates with input rate just below the nominal capacity of the link. The queue lengths are close to zero and as a result round trip delays are close to the minimum possible.

Other rate based schemes, particularly those using queue thresholds as congestion indicators, attempt to keep the queue lengths close to the thresholds. Thereby, introducing unnecessay delay in the path.

Even the credit-based schemes keep the a certain queue length at each hop and as a result the round trip delays are generally an order of magnitude larger than the minimum.

5.4 Congestion Avoidance

The OSU scheme is a congestion avoidance scheme. As defined in Jain (1986) [6], a congestion avoidance scheme is one that tries to keep the network at high throughput and low delay. A simple test to see if a scheme is a congestion avoidance scheme is to see if its operating point will change as the number of buffers in the switches is increased enormously. Most congestion control schemes base their operating point on buffer availability. Therefore, the delay goes up as the buffer size is increased. Note that the credit-based scheme has this characteristics. This has the undesirable propoerty that as the network owners put more memory resources in their network, their delay performance deteriorates. A congestion avoidance scheme's operating point does not depend upon buffers. The OSU scheme will work the same way provided the switches have reasonable amount of buffers.

In general, a properly designed rate-based scheme will be better than a credit based scheme in terms of end-to-end delay. This is because the effective rate of flow of cells belonging to a particular VC changes at every hop in credit based scheme. The cells have to be buffered at the switch because of rate variations.

5.5 Using Measured Rather Than Declared Overload

The MIT scheme uses "desired cell rate" to compute the fair share. It is possible that a source may not be able to use the declared rate. The unused capacity is wasted since it is not allocated to other sources. 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. The link bandwidth is wasted.

In the OSU scheme, we measure the current load and all unused capacity is allocated to contending sources.

5.6 The Scheme works for Bursty Traffic

In MIT scheme, the source does not transmit anything during the interval between bursts. Again the unused bandwidth cannot be allocated to other sources unless the inter-burst time is so large that the switch times out and allocates the bandwidth to other contending sources.

In the OSU scheme, we measure the offered average cell rate and, therefore, no capacity is wasted. Simulation results for the OSU scheme under bursty traffic are presented later in Section 8.

5.7 Minimal number of parameters

Schemes with too many parameters are difficult to use and can be easily mistuned by improper setting of these parameters.

In one version of PRCA, there were more than 10 parameters including the multiplicative decrease factor, additive increase rate, Additive decrease rate, EFCI setting interval n, RM Cell opportunity interval, etc.

In the OSU scheme, the network manager sets just three parameters: the averaging interval for switches, the target link utilization, and the half-width of the target utilization band.

5.8 Parameter Insensitivity

Some schemes are very sensitive to the parameter value. 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.

The parameters of the OSU scheme do not depend upon the lengths of the link or the distances travelled by the VCs.

5.9 Ease of Setting Parameters

Setting the three parameters of the OSU scheme is rather easy. The desired link utilization levels provide a tradeoff between efficiency and time to achieve fairness. High link utilizations will lead to higher queue lengths and slower progress towards fairness.

The switch averaging interval affects the stability of measured load and provides a tradeoff between oscillations and time to achieve optimality. Shorter intervals cause more variation in the measured load and hence more oscillations. Larger intervals cause slow feedback and hence slow progress towards optimality.

5.10 Order 1 Operation

The MIT scheme requires the switches to remember the rates for all VCs and, therefore, its storage requirements as well as computation 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. Therefore, the computation of fair share is O(1).

5.11 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 [5] 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 [13] 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 avoid the unnecessary adjustments by providing explicit instructions to the sources when to change the load.

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.

5.12 Using input rates rather than queue length as the load measure

Most congestion control schemes for packet networks in the past were window based. It is rather common to take these window based control scheme and simply change windows to rate. This does not work well. For a detailed discussion of rate versus window, see Jain (1990)[3]. 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.

As an example, consider two rate controlled queues. Suppose the first queue is only 10 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.

Any rate based scheme which uses queue threshold to control input rate is bound to be wrong. While queue length is a good load indicator for window controlled queues, queue growth rate or input rate is the correct load indicator for the rate controlled queues. Missing this fundamental point is the cause of ineffectiveness of many rate-based schemes.

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 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.

5.13 Fairness is achieved without any fair queueing

One of the basic requirement of the rate-based camp at ATM forum was that the implementors don't want to use per-VC queueing or scheduling. The credit based approach is fair only if fair queueing is used at each switch. Since all cells are of the same size, fair queueing for ATM networks is equivalent to the round-robin service. The MIT and OSU schemes provides fairness with the usual first-in first out (FIFO) service.

5.14 Feedback is Related to Control.

One of the fundamental principles in designing a congestion control scheme (or any control scheme for that matter) is that it helps to know what value of control the feedback is related to. Forgetting this golden rule often leads to congestion control algorithms that do not work. For example, when the network tells the source that it is overloaded, it would be helpful for the source to know what was its control (load) which caused the network to get overloaded. Since the control is a dynamic quantity and there is a nonzero feedback delay, the current control may not be what the feedback is related to.

One example of violation of this rule is the proposal that the switches should put feedback in the control cells going in the reverse direction. The queue state in the switch at the time of feedback has nothing to do with the transmission rate that is indicated in the control cell.

It is to follow this golden rule of keeping feedback and control related that we include TCR and OCR in the control cell and that we use the load level at control cell arrival rather than at departure in computing the feedback.

Another example of feedback not related to the control is the idea that the control cells should be put in a separate queue and given priority over data cells. Thus, the feedback will return fast. We tried this and found that it does not work because the queue state at the time when the control cell reaches a switch may or may not be related to the load indicated in the control cells.

6 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.

6.1 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 8. The averaging interval of 300 μs and a target utilization band of 90(1± 0.1)% is used.

6.2 Single Source

This configuration shown in Figure 9 consists of one VC passing through two switches connected via a link. This configuration was helpful in quickly discarding many alternatives. Figure 10 shows plots for TCR, link utilization, and queue length at the bottleneck link. Notice that there are no oscillations.

6.3 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 11. Figure 12 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.

6.4 Three Sources

As shown in Figure 13, 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 14 shows the simulation results for this configuration.

6.5 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 15.

6.6 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 16. The simlation results are shown in Figure 18. Notice that all VCs receive the same throughput without any fair queueing.

6.7 Upstream Bottleneck

This configuration consists of four VCs and three switches as shown in Figure 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 20 show the simulation results for this configuration. In particular, the TCR for VC2 and VC4 are shown. Notice that VC4 does get the remaining bandwidth.

7 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 21 shows the simulation results for two sources WAN configuration with transient.

8 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 [7]. 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.

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 22 and 23 show simulation results for the transient and the upstream bottleneck configurations using the packet train model.

9 Effect of Various Parameters

Unlike other schemes, the OSU scheme has very few parameters. We have deliberately kept the number of parameters low and the parameters are easy to understand, so that even unskilled network managers can set the parameters correctly. Setting of the two parameters, load averaging interval and the target utilization band is the topic of this section.

9.1 Load Averaging Interval

The load averaging interval controls the variance in the load estimate and the time to adopt to load changes. Very small intervals can cause high variance in the estimate causing too many oscillations. However, if the load changes significantly (for example, a high bandwidth source becomes quiet), the system will become aware of the change faster. Very large intervals provide smooth estimates of the load resulting in less oscillation but the load changes will be sensed much later.

Since the same load averaging interval is used by the sources, the load averaging interval affects the number of control cells and hence the overhead caused by the congestion control mechanism. For example, if the averaging interval is equal to 200 cell times, one-half of one percent of the bandwidth will be used by the control cells.

9.2 Target Utilization Band (TUB)

There are two characteristics of the target utilization band: the target utilization level, and the width of the TUB. For example, if the TUB is set at 90(1±0.1)%, the target utilization level is 90% and and the width is 18%.

The target utilization level sets the utilization goal under overload. It controls the drain rate of the queue under overload. For example, when the target utilization is set at 90%, the switch attempts to bring the input rate down whenever it exceeds 90%. The queue is still served at 100% and the difference 10% is the rate of decrease of queue length.

The width of the TUB determines the size of the input rate oscillations under steady state. For example, with a TUB of 90(1±0.1), the input rate will stay between 81 to 99% of the link rate. From this point of view, the width should be small. However, the width also affects the rate at which fairness is achieved. Larger width results in fairness more quickly. Thus, the width provides a tradeoff between time to fairness and the size of the oscillations.

10 Additional Optional Improvements of the OSU scheme

The scheme as described so far is the basic necessary part to achieve fairness and efficiency. Optional enhancements that improve the performance under certain circumstances are described next.

10.1 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 load level 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 discouraging a source from increasing simply because it is using more than its fair share. We can't be sure that underloading sources can use the extra bandwidth and if we don't give it to a overloading (over the fair share) source, the extra bandwidth may go unused.

The aggressive fairness option, which is described later in this section, is based on a number of considerations. The considerations for increase are:

1. When a link is underloaded, all of its user 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 50Mbps. 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 increasing 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.

6. The new rate (input rate/load adjustment factor) should also be a continuous and monotonically increasing function of the input rate.

7. The new rate should be a continuous and monotonically decreasing function of the load level.

The corresponding considerations for overload should be obvious from the above. These are:

1. When a link is overloaded, all of its user will be asked to decrease. No one will be allowed to increase.

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

3. The minimum required decrease factor should be less than or equal to the current load level. For example, if the current load level is 200%, no source can be allowed to decrease by less than a factor of 2.

4. The load adjustment factor should be a continuous function of the input rate.

5. The load adjustment factor should be a monotonicaly increasing function of the input rate.

6. The new rate should also be a continuous and monotonically increasing function of the input rate.

7. The new rate should be a continuous and monotonically decreasing function of the load level.

It must be emphasized that the above considerations for increase and decrease apply only outside the TUB. Once inside, TUB, we violate almost all of the above except monotonicity.

A sample pair of increase and decrease functions that satisfy the above criteria are shown in Figure 24. 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 24a. If current load level is z, and the fair share is s, all sources with input rates below the zs are asked to increase by z. Those between zs and z are asked to increase by an amount between z and 1.

Figure 24b 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 will do. The linear functions are easy to compute and provide the continuity property that we seek.

Figure 26 shows the simulation results for the transient configuration with the aggressive fairness option.

Similar results are obtained for other configurations.

10.2 Precise Fair Share Computation Option

Given the actual rates of all active sources, we could exactly calculate the fair share using the algorithm specified in Section 3.2.3. Thus, in place of using only the number of active VCs, we could the OCRs of various sources to compute the fair share. This option provides a performance much better than that possible with MIT scheme because the following features that are absent in the MIT scheme:

1. Provide a bipolar feedback. The switches can increase as well decrease the rate in the control 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.

3. Measure actual load level at the switch and use it to scale the fair share. This helps with bursty sources. The bandwidth not used during inter-burst periods is allocated to other sources.

Figure 27 shows the simulation results for the OSU scheme with precise fair share computation option for the upstream configuration of Figure 19. Notice that with precise knowledge of the fair share, the oscillations during steady periods are all gone. The oscillations happen only when there is a change in the workload.

10.3 Backward Congestion Notification Option

One common criticism of end-to-end feedback control schemes is that the control is slow if the round-trip delays are large. This is particularly important for high-speed networks since the propagation delays become significant compared to the data transmission delays. One way to overcome this is for the switch to send the feedback right back to the source and thus avoiding the round-trip delay of the remaining path. Although this option, commonly known as backward Explicit Congestion Notification (BECN) has been known for quite some time and is even allowed by the frame-relay and ATM UNI standards, no satisfactory schemes have been developed for this. This is because, most designers of the BECN have missed the key point, that correlating feedback with the correct control is the most important aspect of a congestion control scheme (and for that matter, any control system) design.

The problem with BECN can be seen easily by the configuration of Figure 29. The source is sending at 155 Mbps and sends a control cell. The switch happens be unloaded at that time and so lets the first control cell go unchanged. By the time, the second control cell 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 bit later the first control cell returns telling the source that the rate of 155 Mbps is ok. The control cell are received out of order rendering the BECN ineffective.

To ensure correct operation for BECN, we have set up the following rules for using the BECN option of the OSU scheme:

1. The BECN should be sent only when a switch is overloaded AND the switch wants to decrease the rate below that indicated in the load adjustment factor field of the control cell. There is no need to send BECN if the switch is underloaded. This avoids the problem of one switch asking a source to go up and a subsequent switch asking it to go down. Similarly, there is no need to confuse the source if the switch can only allow a load adjustment factor lower than that indicated in the control cell.

2. The source should include a timestamp in the control cell indicating the time when the control cell was generated. This helps distinguish successive cells. The timestamp is ignored at all intermediate switches and the destination and is used only at the source. Thus, no clock synchronization among nodes is required or assumed.

3. All control cells complete a round-trip. If a switch wants to send a BECN, it waits until it receives an control cell. It makes two copies of the it. One copy is forwarded in the forward direction. The other is sent back to the source.

4. The control cell also includes a bit called "BECN bit." This bit is initialized to zero at the source and is set by the congested switch in the copy of control cell that is sent backward. This helps the source know whether a received control cell has visited the complete path or only a part of it. The cells that have completed only a part of the path are called "BECN cells" as opposed to "FECN cells" that have completed the entire path.

5. The source remembers the time stamp 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 control (BECN or FECN) cell is less than Taa, the cell is ignored. This rule helps avoid out-of-order control cells.

6. If the timestamp of an control cell received at the source is equal to or greater than Taa, the new value of TCR is computed:

   $$\text{New TCR} = \text{TCR/Load adjustment factor}$$

   and the transmission rate is adjusted as shown in Table 1.

   Table 1: Source Behavior on Receiving an control cell with Timestamp $\geq$ Taa

	BECN	FECN
TCR $<$ New TCR	Ignore	TCR $\leftarrow$New TCR
TCR $\geq$ New TCR	TCR $\leftarrow$New TCR	TCR $\leftarrow$New TCR

   The four cases are:

   (a) BECN Cell Granting Increase: Thi happens if the new TCR is more than the current TCR. Ignore this cell. In other words, a BECN cell cannot result in a rate increse. The rate increase has to wait until the corresponding FECN cell returns.

   (b) BECN Cell Requesting Decrease: Comply. The TCR is decreased to the new TCR in the cell.

   (c) FECN Cell Granting Increase: Comply. The TCR is increased to the new TCR.

   (d) FECN Cell Requesting Decrease: Comply. The TCR is decreased to the new TCR.

With above rules, the BECN option of the OSU scheme reduces the time to reach the efficiency zone. The reduction is significant only in those WAN cases where the remaining path length is large. A sample example is shown in Figure 30 for the upstream congestion configuration of Figure 19. The corresponding result without BECN option was shown earlier in Figure 28.

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.

A complete layered view of various components of the OSU scheme is shown in Figure 31. 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 31, these options can be used individually or in a layered manner.

11 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 [5]. 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.

12 Summary

We have developed a end-to-end rate based congestion avoidance scheme for ABR traffic on ATM networks. In the scheme, the sources periodically send control cells containing the measured offered average cell rate. The switches also measure the usage of links and allocate all unused bandwidth fairly among the contending ABR sources. A simple fairness algorithm using only the number of active sources is the minimum required component. The basic scheme performs very well for bursty sources and takes the network to max-min optimal. It is a congestion avoidance scheme in the sense that it provides maximum throughput and minimal delay and is therefore ideally suited if weakly delay-sensitive video traffic is sent using ABR connections. The scheme has been designed with minimal number of parameters that can be easily set.

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

References

[1] D. Chiu and R. Jain, "Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks," *Journal of Computer Networks and ISDN Systems*,[2] Vol. 17, No. 1, June 1989, pp. 1-14.

[2] R. Jain, "Myths about Congestion Management in High Speed Networks," *Internetworking: Research and Experience*, Vol 3, 1992, pp. 101-113.

[3] R. Jain, "Congestion Control in Computer Networks: Issues and Trends," *IEEE Network Magazine*, May 1990, pp. 24-30.

[4] R. Jain, "A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks," *Computer Communications Review*, Vol. 19, No. 5, October 1989, pp. 56-71.

[5] R. Jain, K. K. Ramakrishnan, and D. M. Chiu, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer," Digital Equipment Corporation, Technical Report, DEC-TR-506, August 1987, 17 pp. Also in C. Partridge, Ed., *Innovations in Internetworking*, Artech House, Norwood, MA, 1988, pp. 140-156.

[6] R. Jain, "A Timeout-Based Congestion Control Scheme for Window Flow-Controlled Networks," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-4, No. 7, October 1986, pp. 1162-1167.

[2] All our papers and ATM Forum contributions are available through http://www.cis.ohio-state.edu/~jain/

[7] R. Jain, D. M. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems," *Digital Equipment Corporation, Technical Report DEC-TR-301*, September 1984, 37 pp.

[8] R. Jain, "Delay Based Congestion Avoidance in Computer Networks," *U.S. Patent #5,193,151*, issued on March 9, 1993. *Australian patent #639017*, issued November 23, 1993. Patent pending in Canada, Europe, and Japan.

[9] K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with Connectionless Network Layer," *ACM Transactions on Computers*, May 1990. Reprinted in Amit Bhargava, Ed., "Integrated Broadband Networks" Artech House, Norwood, MA, 1990.

[10] 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," *Digital Equipment Corporation, Technical Report DEC-TR-510*, August 1987, 41 pp.

[11] Anna Charny, David Clark, Raj Jain, " Congestion Control With Explicit Rate Indication", *AF-TM 94-0692*,[3] July 1994.

[12] Anna Charny, "An Algorithm for Rate Allocation in a Cell-Switching Network with Feedback", *MIT TR-601*, May 1994.

[13] Larry Roberts et al," Closed-Loop Rate-Based Traffic Management", *AF-TM 94-0438R1*, July 1994.

[14] Kai-Yeung Siu, Hong Yi Tzeng, "Adaptive Proportional Rate Control for ABR Service in ATM Networks", *UCI Tech Report No.94-07-01*, Dept of Electrical and Computer Engineering, University of California, Irvine. *preprint*

[3] Throughout this section, AF-TM refers to ATM Forum Traffic Management sub-working group contributions.

A Proof: Fairness Algorithm Improves Fairness

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

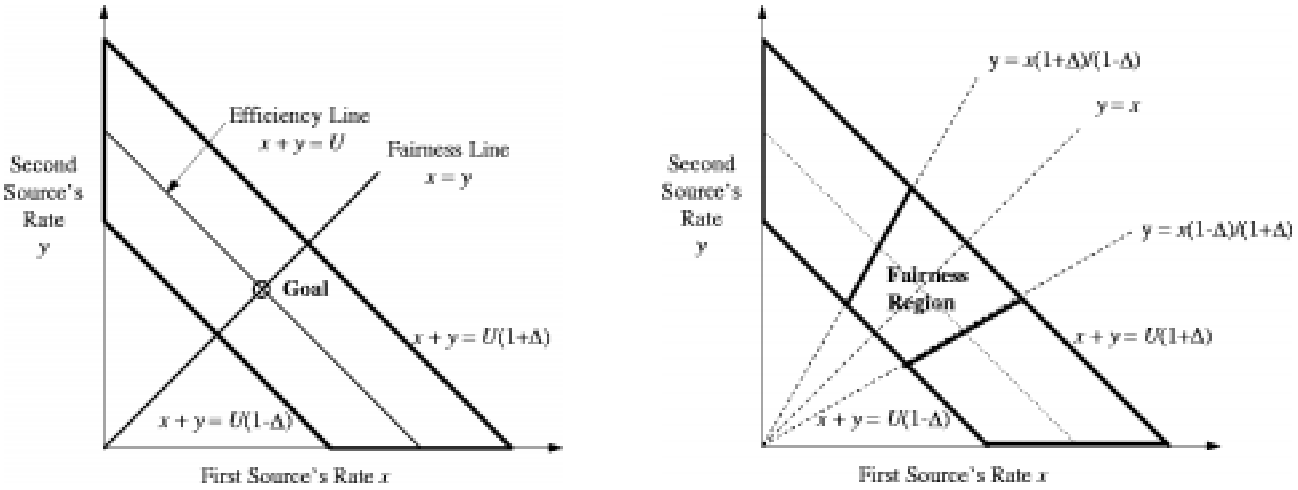
C1. Once inside TUB, the fairness algorithm keeps the link in TUB.

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

Our proof methodology is similar to that used in Chiu and Jain (1989)[1], 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 1(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.



(a) Ideal Fairness Goal

(b) The Fairness Region

Figure 1: A geometric representation of efficiency and fairness for a link shared by two sources

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 1(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 1(b). Mathematically, the conditions defining the fairness region are:

$$\frac{(1+\Delta)}{(1-\Delta)}x \geq y \geq \frac{(1-\Delta)}{(1+\Delta)}x \tag{4}$$

$$U(1+\Delta)\geq x+y\geq U(1-\Delta) \tag{5}$$

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.

A.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 2(a). These regions correspond to the following inequalities:

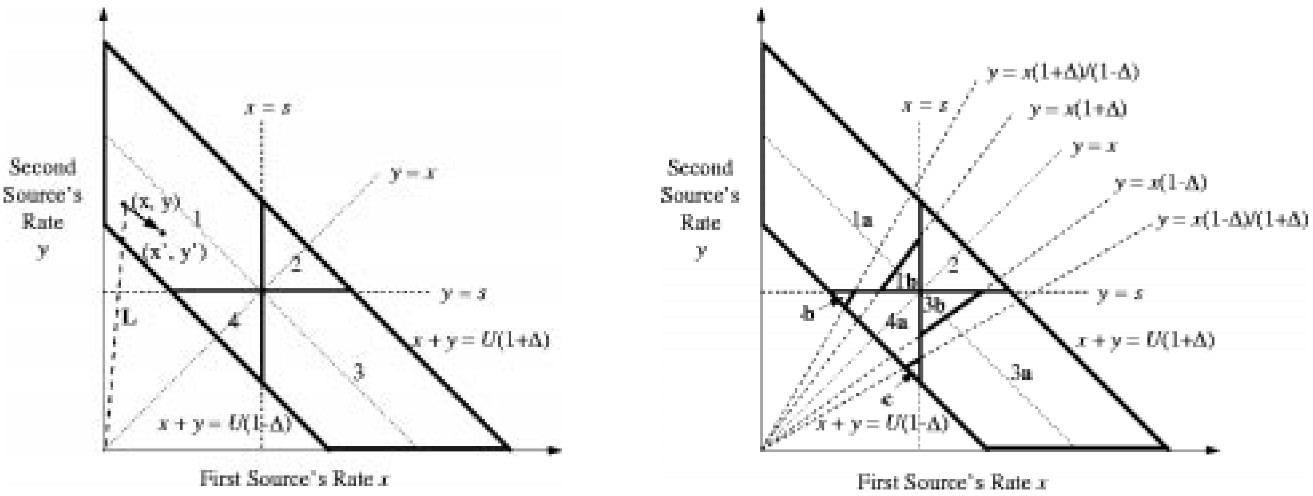
Region 1: $s>x>0$ and $y\geq s$ and $U(1+\Delta)\geq x+y\geq U(1-\Delta)$

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.



(a) Regions used to prove Claim C1

(b) Regions used to prove Claim C2

Figure 2: Subregions of the TUB used to prove Claims C1 and C2

A.1.1 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 $(x', y') = (\frac{x(1+\Delta)}{z}, \frac{y(1-\Delta)}{z})$.

$$
\begin{aligned}
x' + y' &= \frac{x(1+\Delta) + y(1-\Delta)}{z} && (6)\\
&= U(1+\Delta) - \frac{2x\Delta}{z} && (7)\\
&= U(1-\Delta) + \frac{2\Delta}{z} y && (8)\\
& && (9)
\end{aligned}
$$

The quantity on the left hand side of the above equation is the new total load. Since the last terms of equations 7 and 8 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.

A.1.2 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.

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

A.2.1 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,

$$\frac{y'}{x'} = \frac{y}{x} \times \frac{1-\Delta}{1+\Delta} \tag{10}$$

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

$$\frac{y'}{x'} < \frac{y}{x} \tag{11}$$

Further since y/x is greater than $1+\Delta$, equation 10 also implies:

$$\frac{y'}{x'} > (1-\Delta) \tag{12}$$

Equation 11 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 12 says that the new point does not overshoot the fairness region. This proves Claim C2 for all points in region 1a.

A.2.2 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 4 and 5 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:

$$\frac{1-\Delta}{1+\Delta} < \frac{y'}{x'} \leq (1-\Delta) \tag{13}$$

Condition 13 ensures that the new point remains in the fairness region defined by conditions 4 and 5.

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.

A.2.3 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.

A.2.4 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 2(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.

A.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 2(a). If we move from (x, y) to (x, y'), we have:

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

and

$$\begin{aligned} x + y' &= \frac{xz + y(1-\Delta)}{z} && (14) \\ &= U(1-\Delta) + \frac{x\{z-(1-\Delta)\}}{z} && (15) \\ &= U(1+\Delta) - \frac{x\{(1+\Delta)-z\} + 2y\Delta}{z} && (16) \\ & && (17) \end{aligned}$$

Since, the last terms of equations 15 and 16 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.

B Detailed Pseudocode

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{∑ 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 ≥ Lower_Load_Bound) and (Load_Level ≤ Upper_Load_Bound)
        THEN BEGIN
             IF OCR_In_CELL > Fair_Share_Rate
             THEN Load_Adjustment_Decision ←Load_Level/Lower_Load_Bound
             ELSE Load_Adjustment_Decision ←Load_Level/Upper_Load_Bound
        END (*IF *)
        ELSE Load_Adjustment_Decision ←Load_Level;

   IF (Aggressive_Fairness_Option)
        THEN BEGIN
             Load_Adjustment_Decision ←1;
             IF (Load_Level < Lower_Load_Bound)
                  THEN IF ((OCR_In_Cell < Fair_Share_Rate×Load_Level) OR
                       (Num_VC_Active =1))
                       THEN Load_Adjustment_Decision ←Load_Level
                       ELSE IF (OCR_In_Cell < Target_Cell_Rate×Load_Level)
                            THEN Load_Adjustment_Decision ←Load_Level + (1-
                                 Load_Level)×(OCR_In_Cell/(Load_level×
                                     Fair_Share)-1)/(Num_VC_Active-1)
                            ELSE Load_Adjustment_Decision ←1
                  ELSE IF Load_Level ≥ 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 *)
```

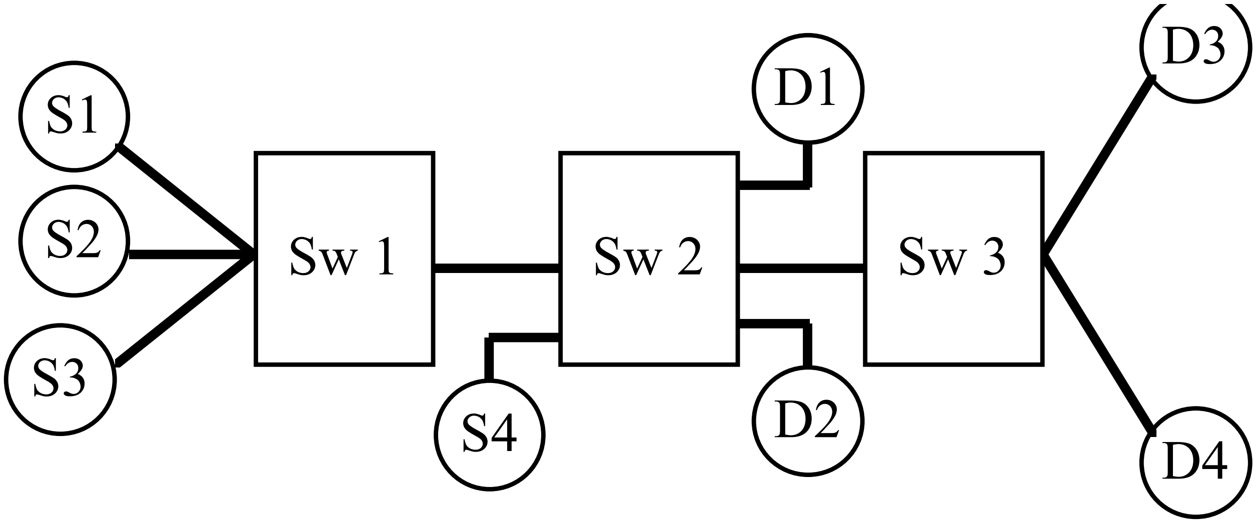


Figure 3: Network configuration for max-min fairness example.

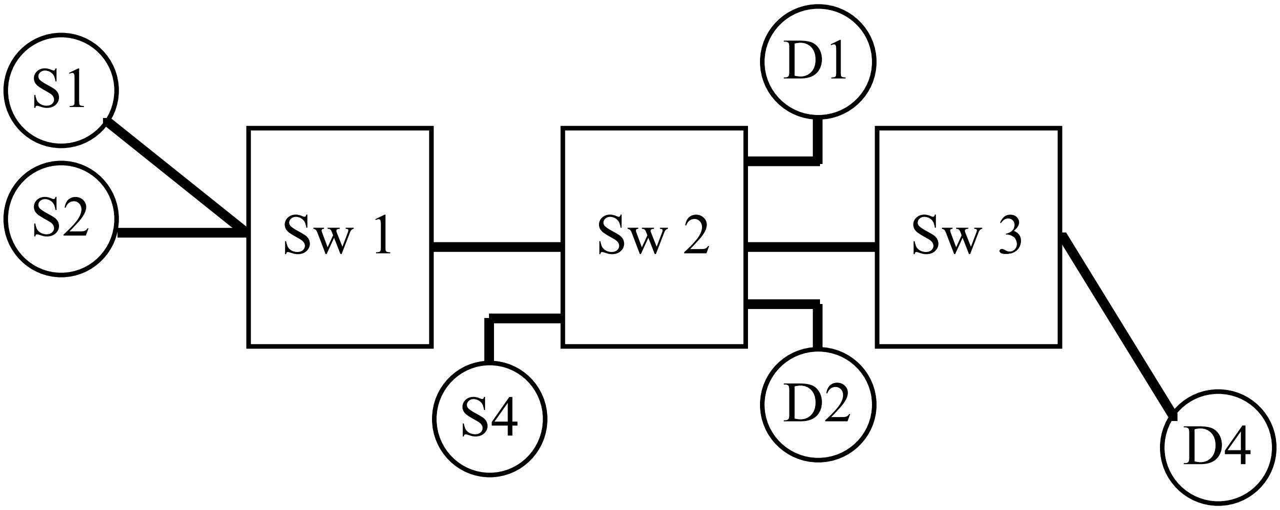


Figure 4: Network configuration for max-min fairness example with source S3 removed

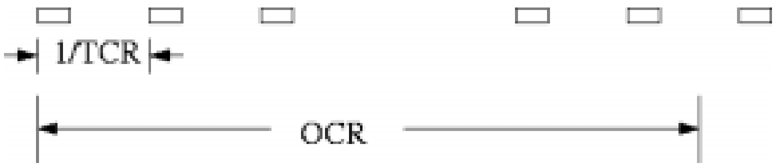


Figure 5: Transmitted cell rate (instantaneous) and Offered Average Cell Rate (average).

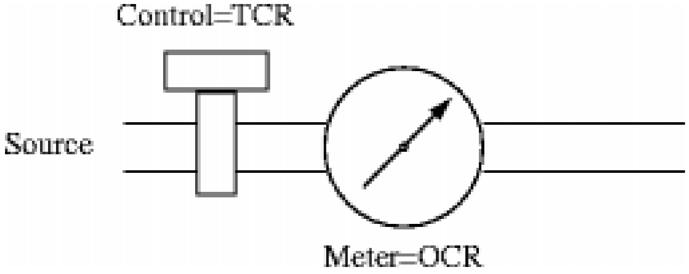


Figure 6: Transmitted cell rate (controlled) and Offered Average Cell Rate (measured).

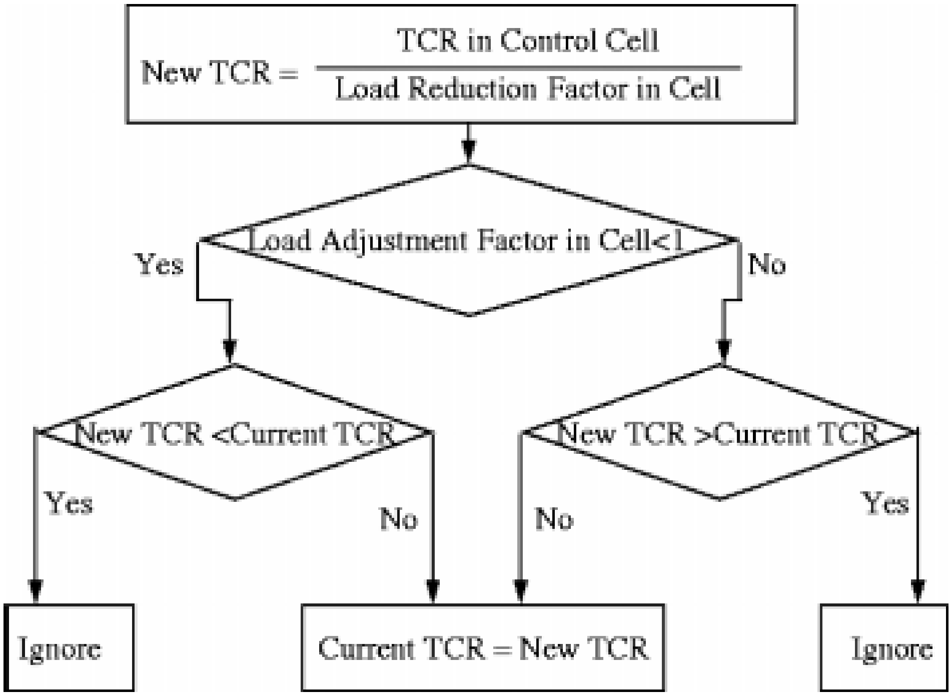

New TCR = TCR in Control Cell / Load Reduction Factor in Cell
Load Adjustment Factor in Cell<1
Yes
No
New TCR <Current TCR
New TCR >Current TCR
Yes
No
No
Yes
Ignore
Current TCR = New TCR
Ignore

Figure 7: Flow chart for updating TCR.

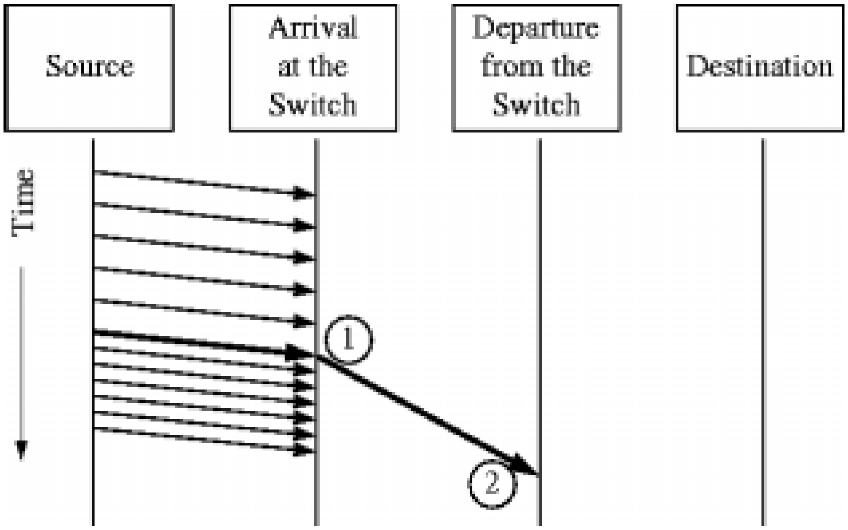


Figure 8: The queue state at the time of arrival is related to the TCR in the control cell. The state at departure may not be.

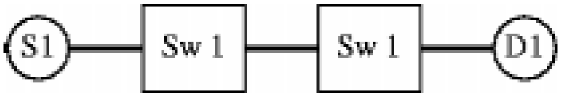
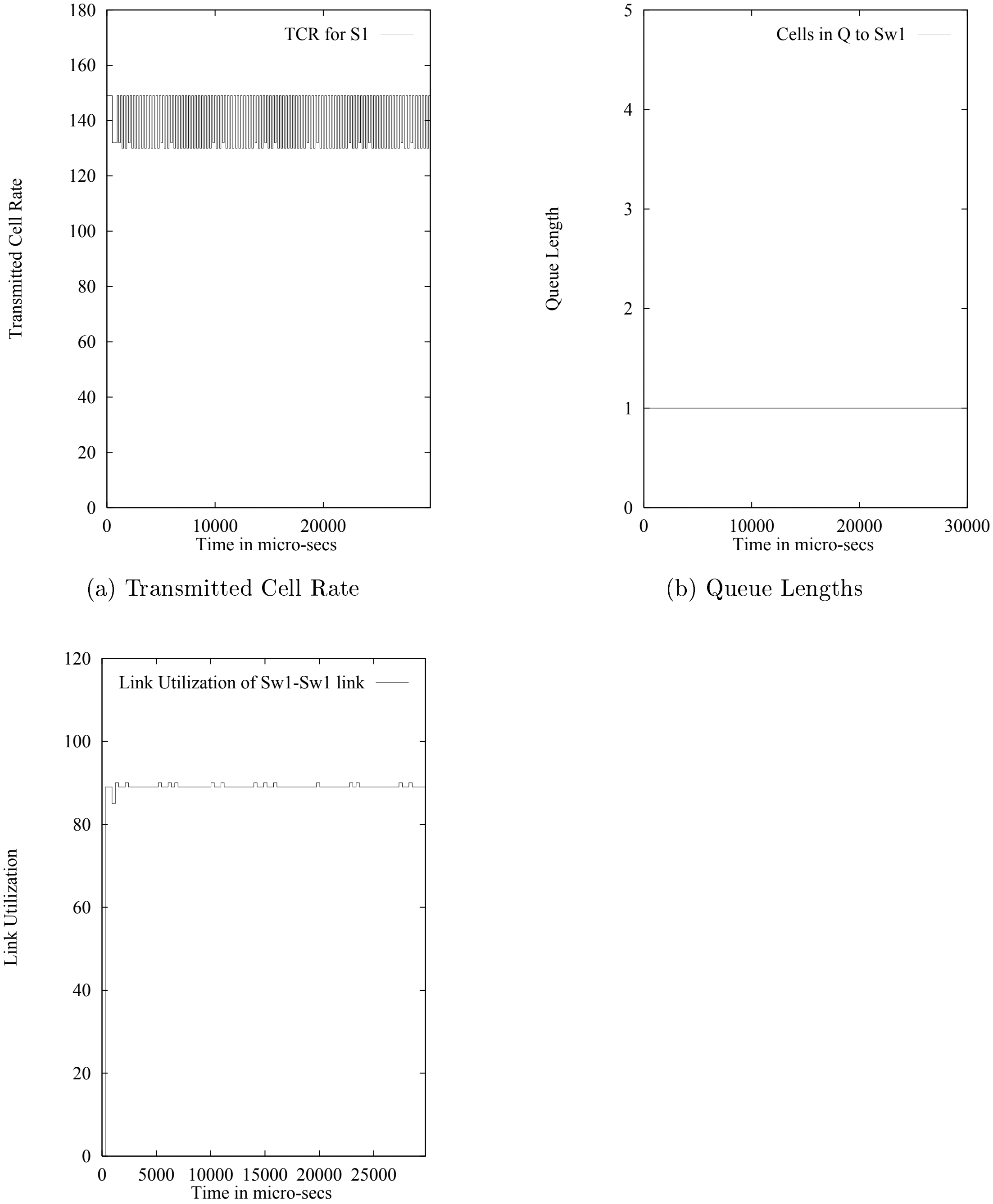


Figure 9: Single source configuration.



(a) Transmitted Cell Rate

(b) Queue Lengths

(c) Link Utilization

Figure 10: Simulation results for the single source configuration

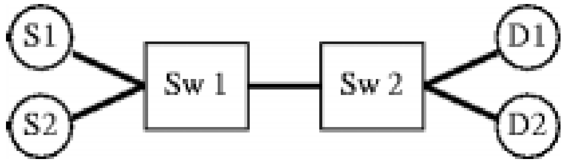
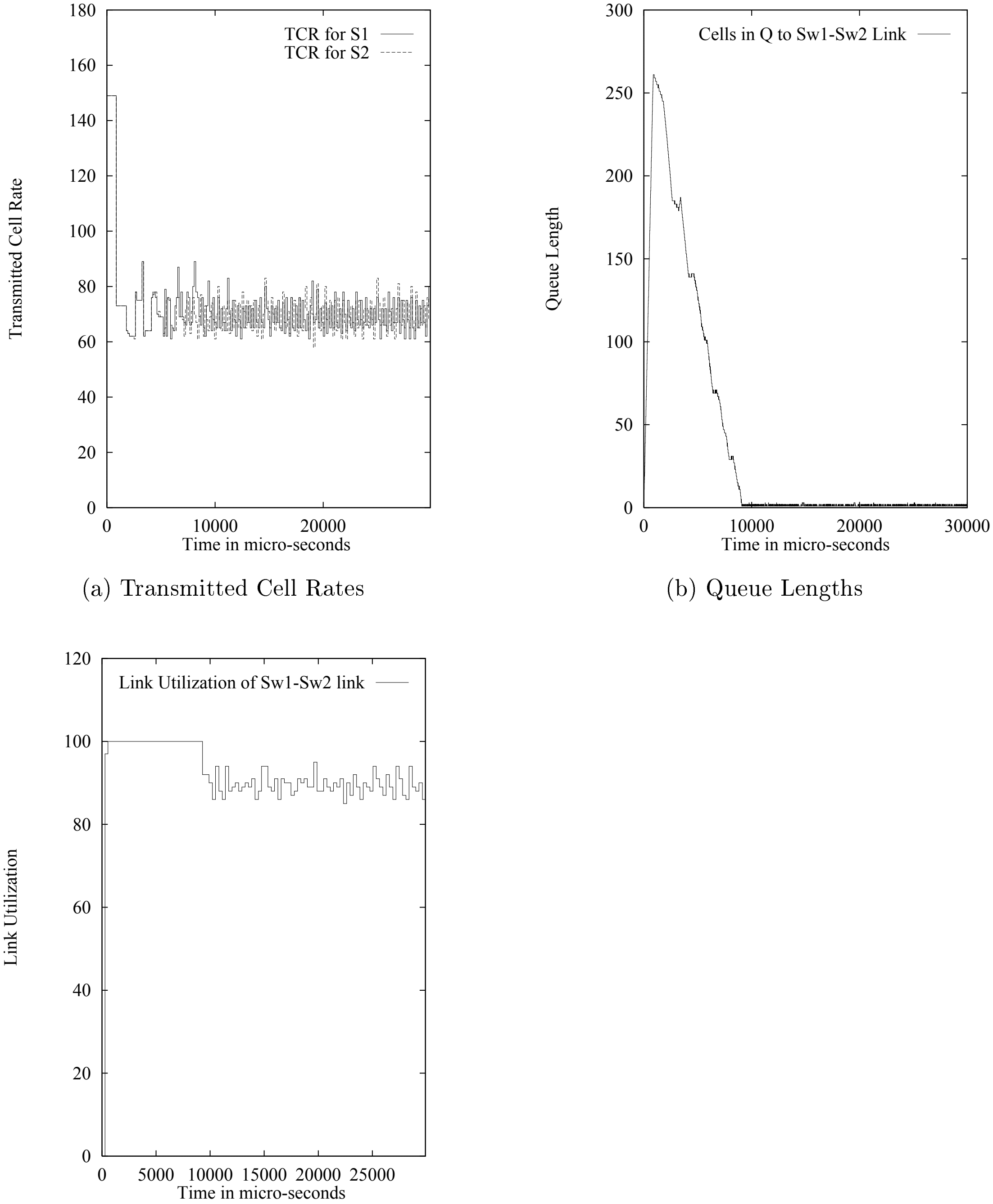


Figure 11: Two-source configuration



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 12: Simulation results for the two-source configuration

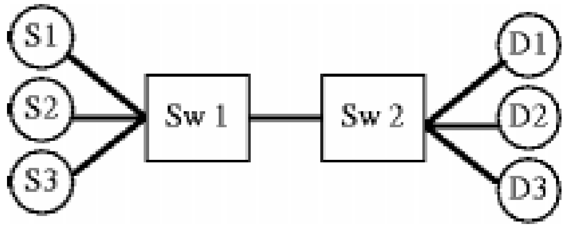
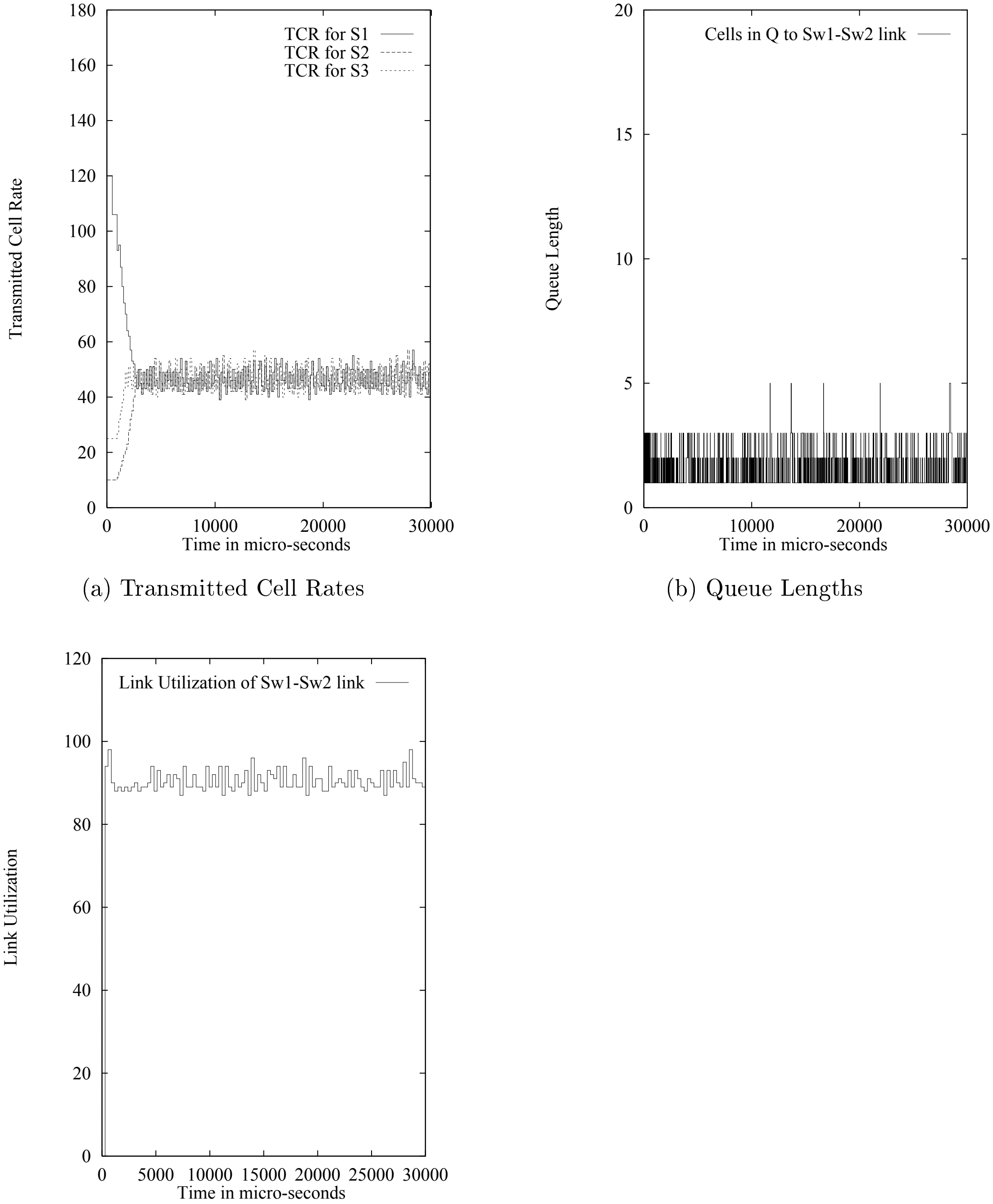


Figure 13: Three-source configuration

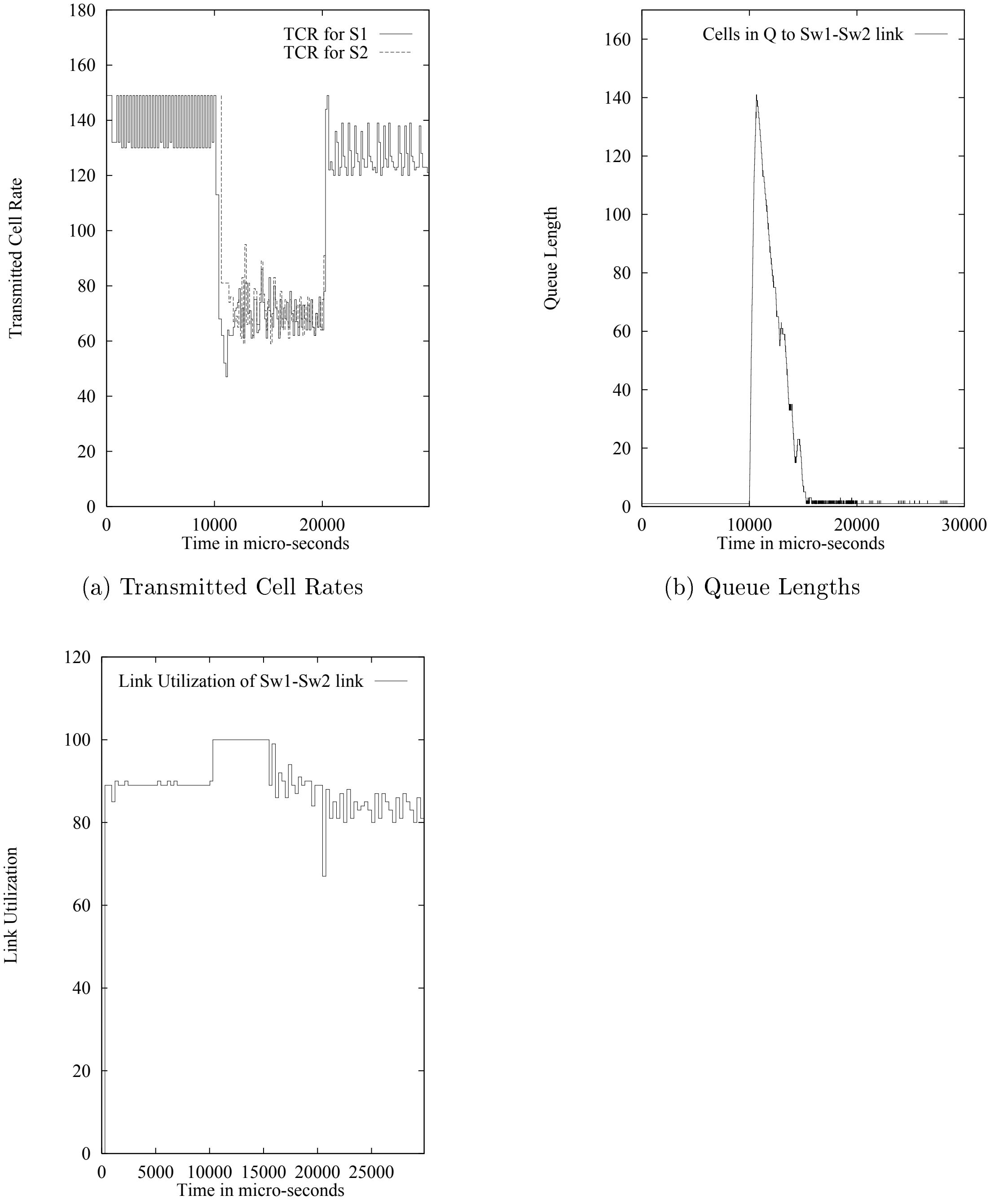


(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 14: Simulation results for the three-source configuration



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 15: Simulation results for the transient experiment

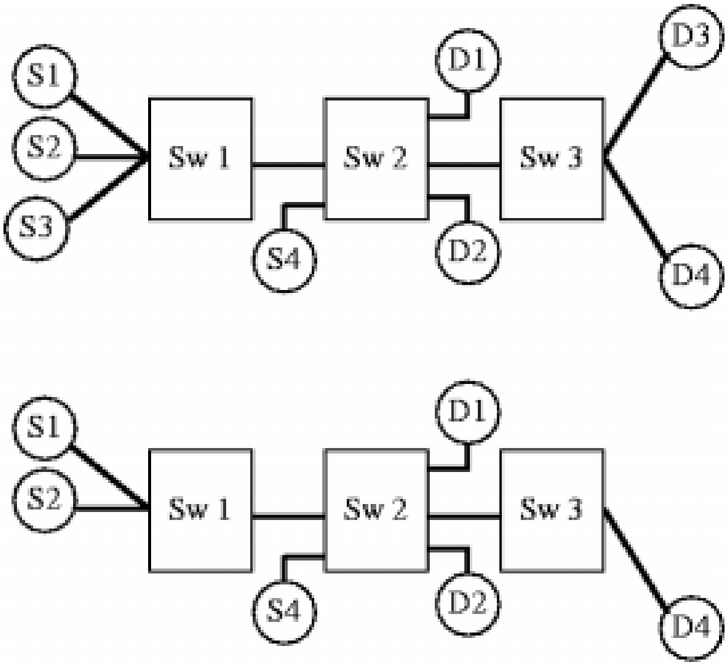


Figure 16: The parking lot fairness problem. All users should get the same throughput regardless of the parking area used.

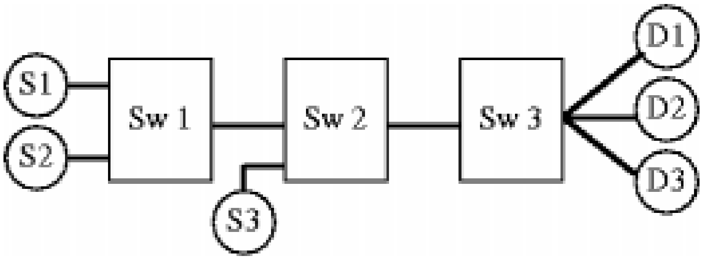
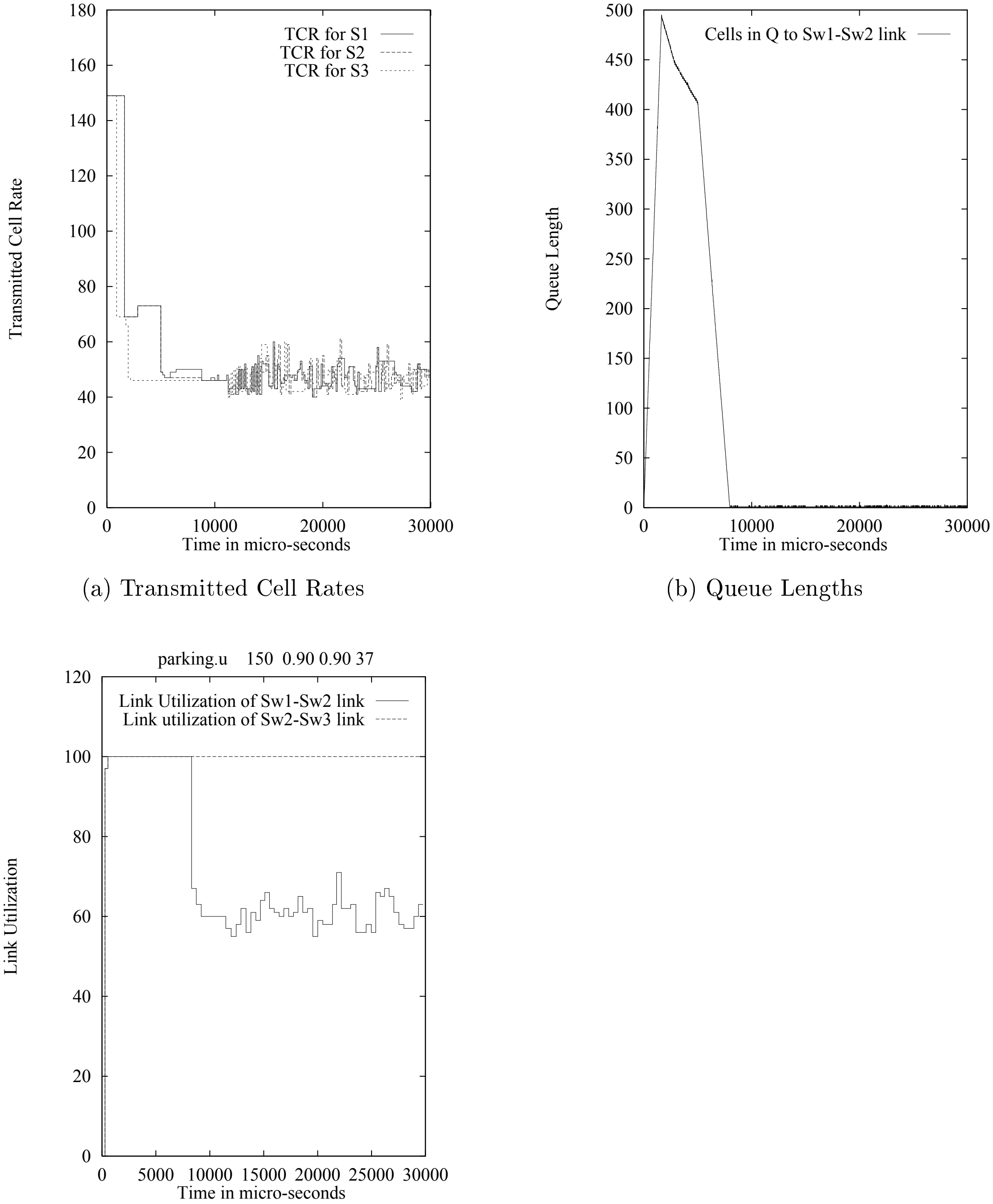


Figure 17: The parking lot configuration



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 18: Simulation results for the parking lot configuration

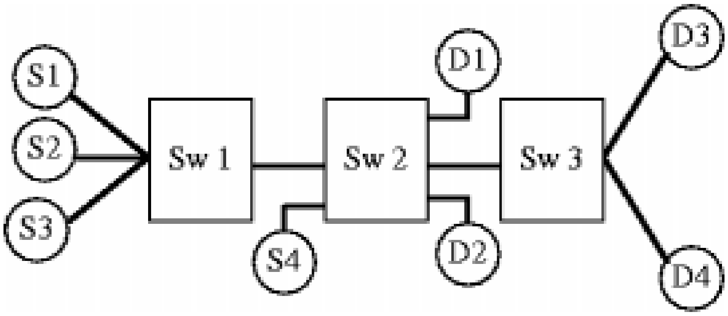
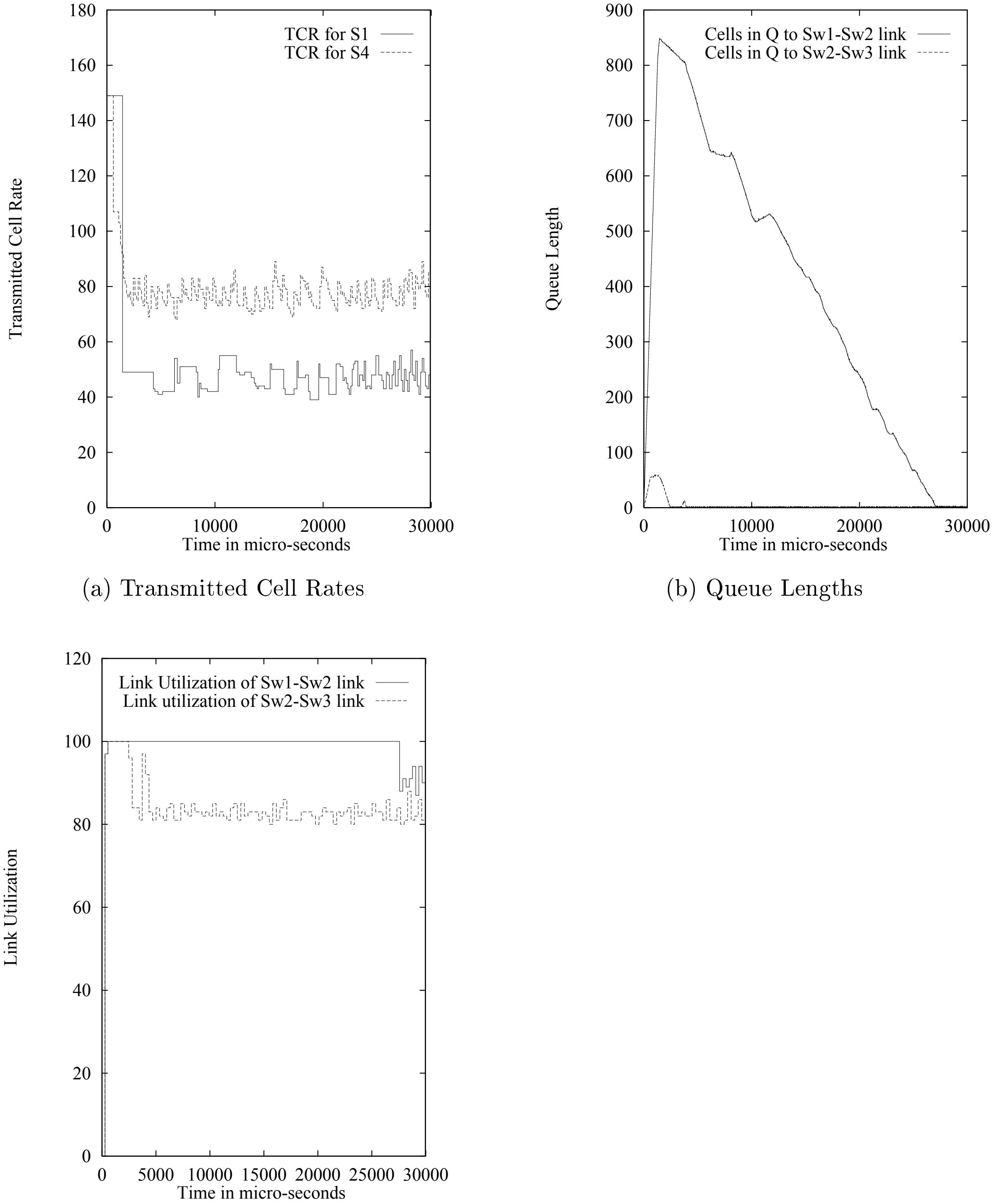


Figure 19: Network configuration with upstream bottleneck.

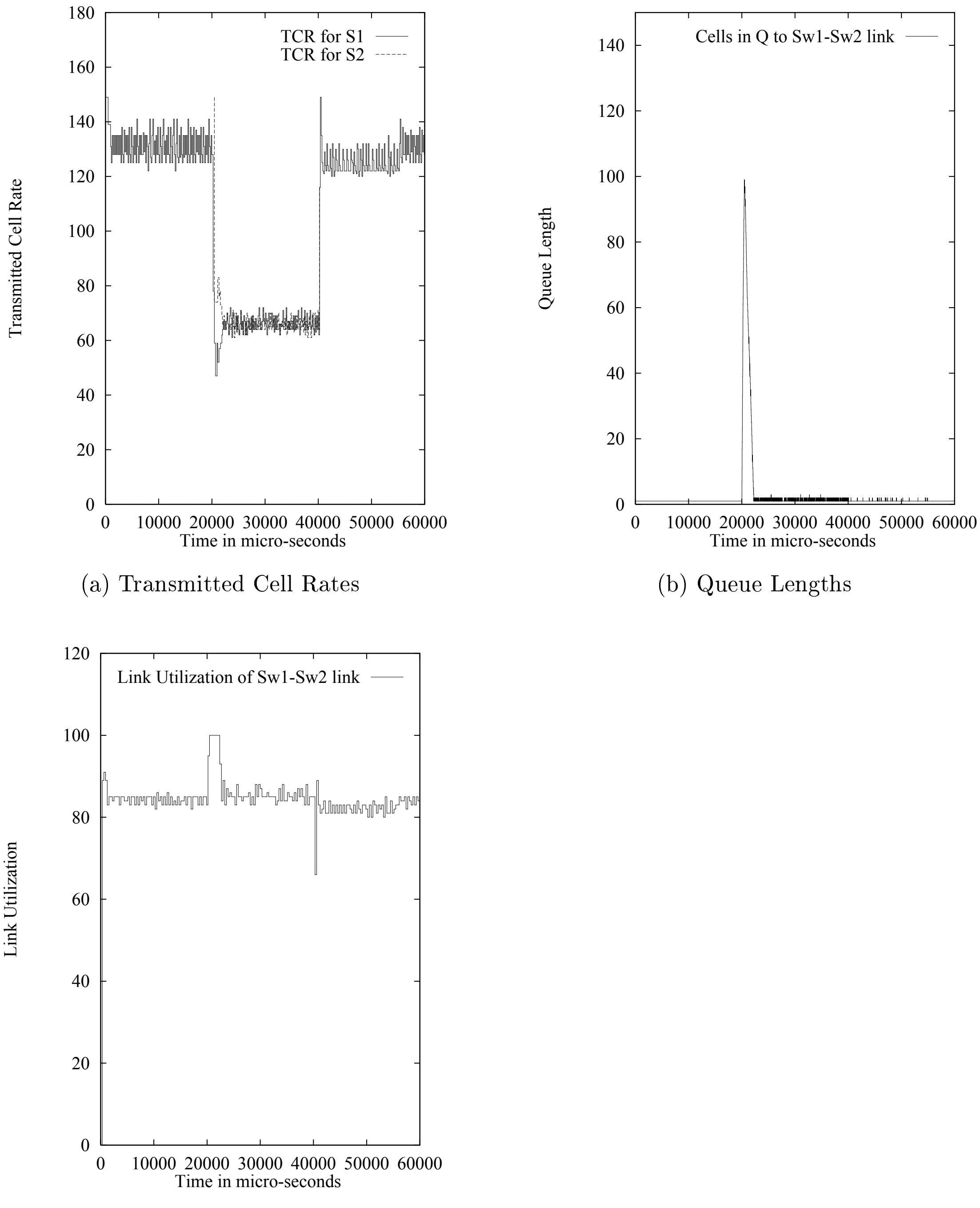


(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 20: Simulation results for the upstream bottleneck configuration

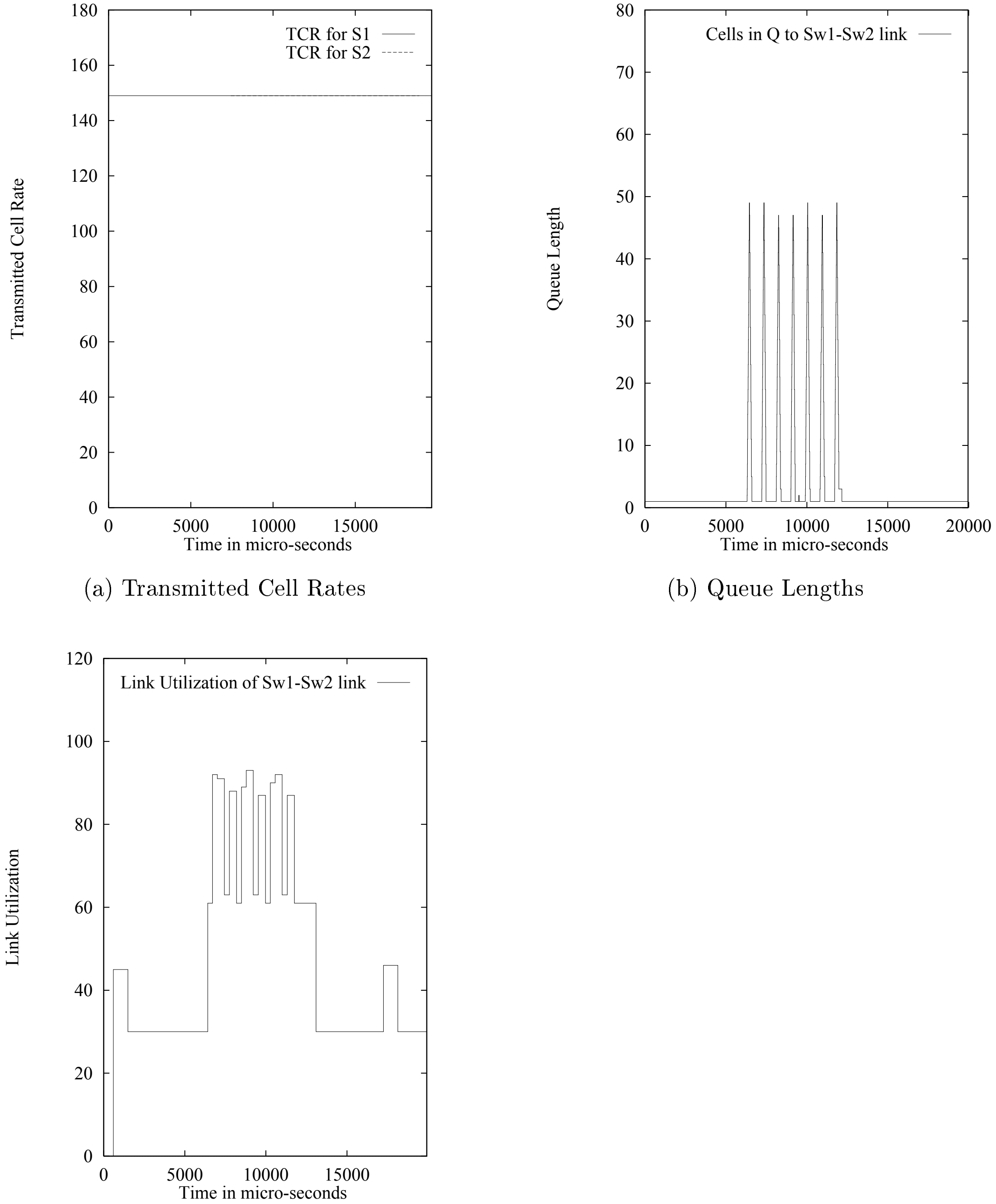


(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 21: Simulation results for the transient configuration with 1000 km inter-switch links

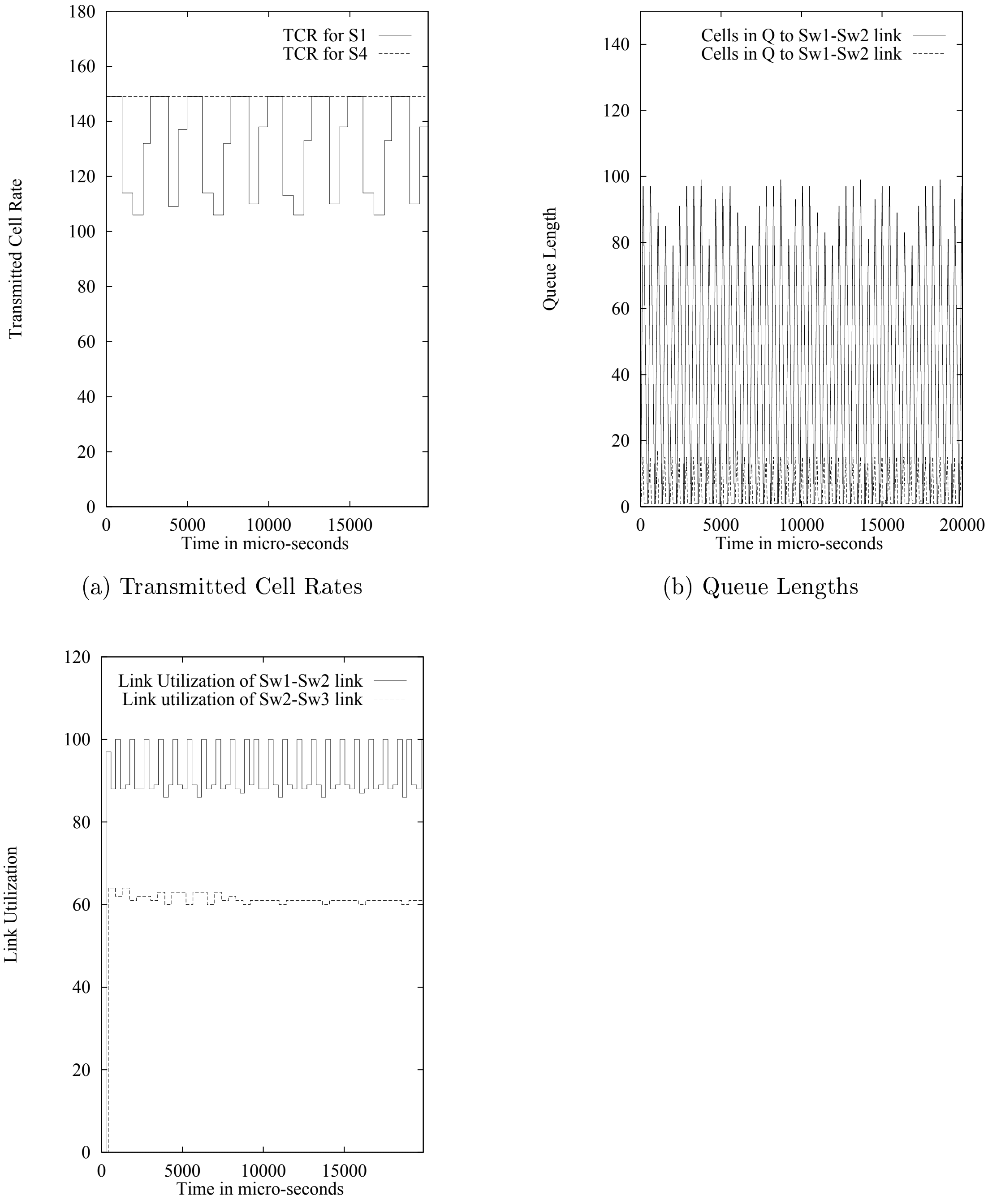


(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 22: Simulation results for the transient configuration with packet train workload.



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 23: Simulation results for the upstream bottleneck configuration with packet train workload.

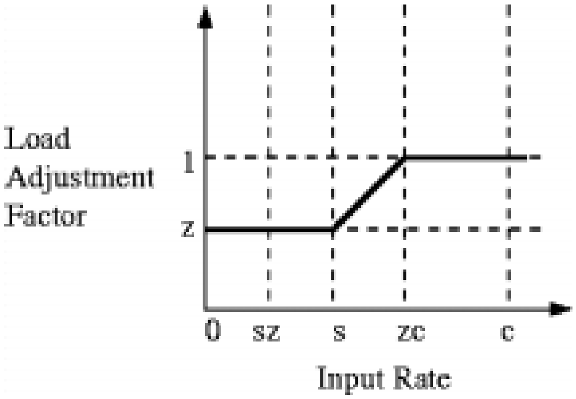


Figure 24: The increase function for the aggressive fairness option

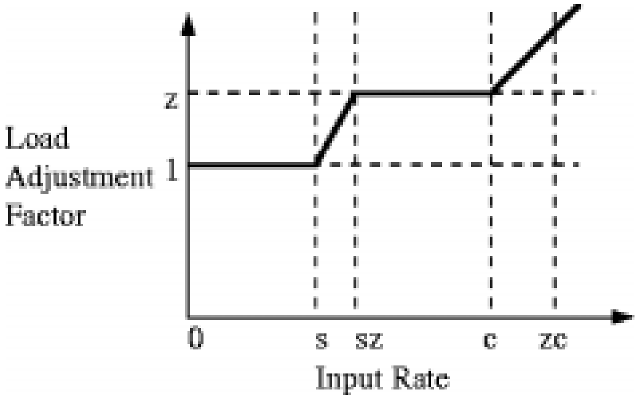
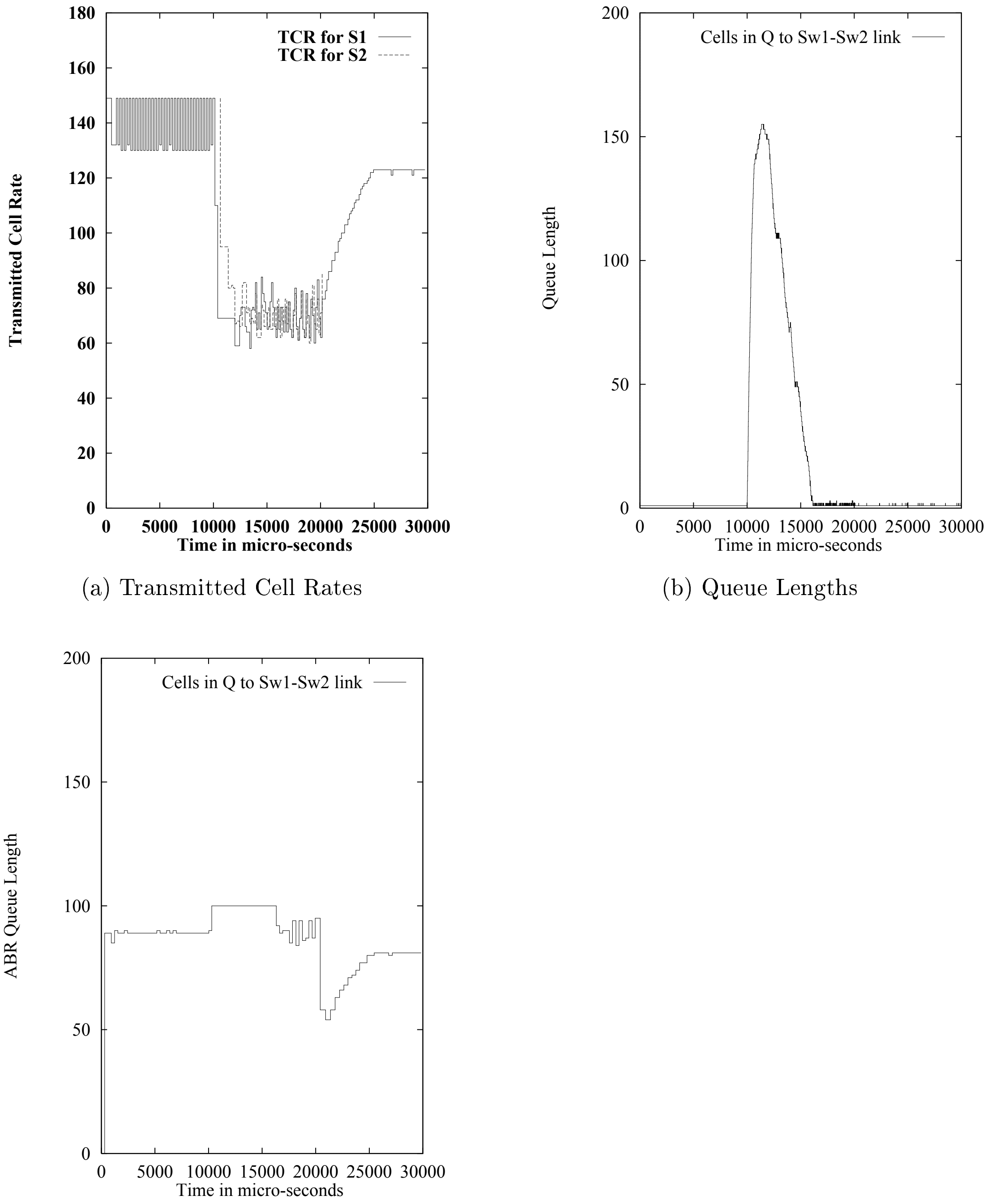


Figure 25: The decrease function for the aggressive fairness option

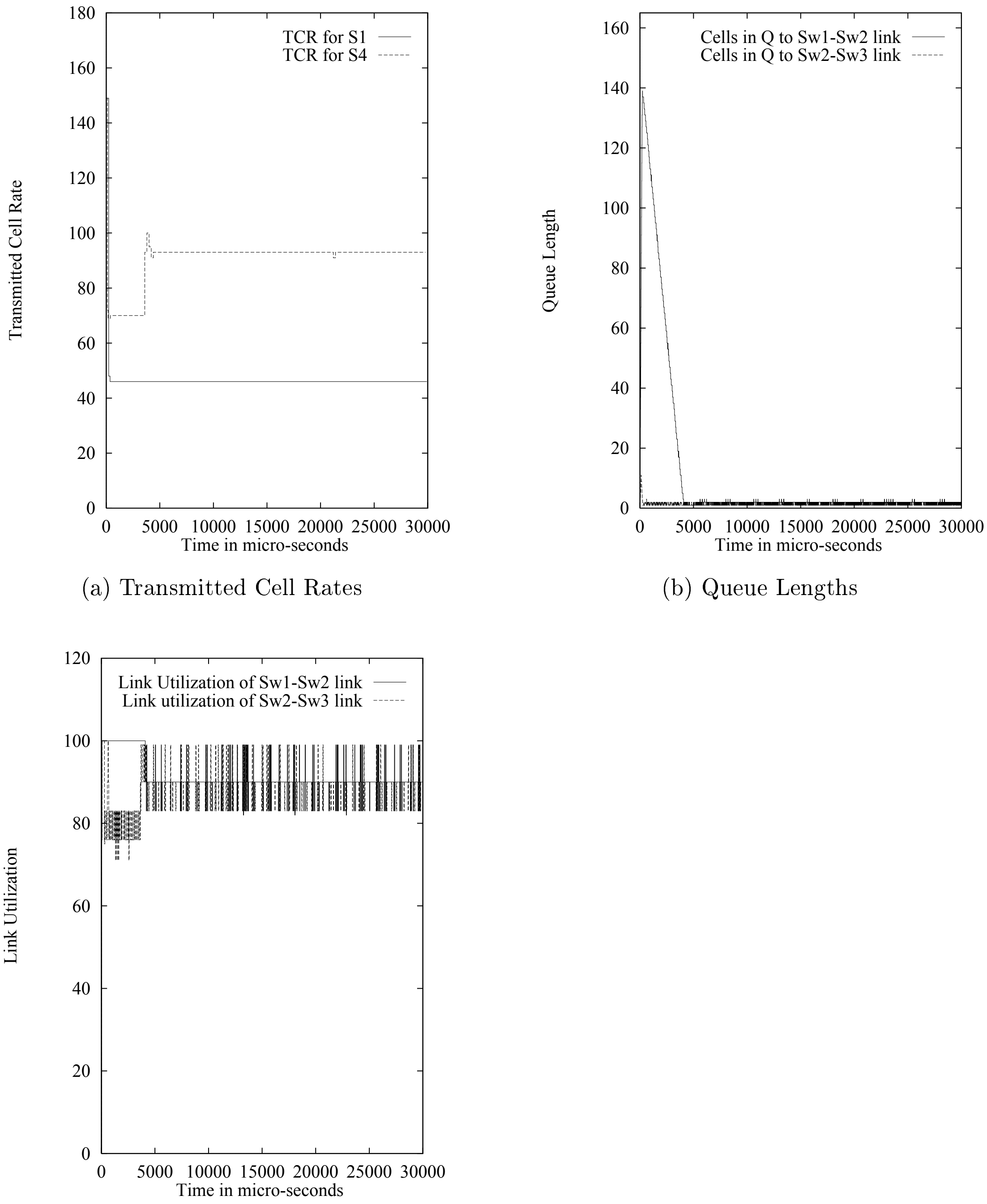


(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 26: Simulation results for the experiment with transients and aggressive fairness option

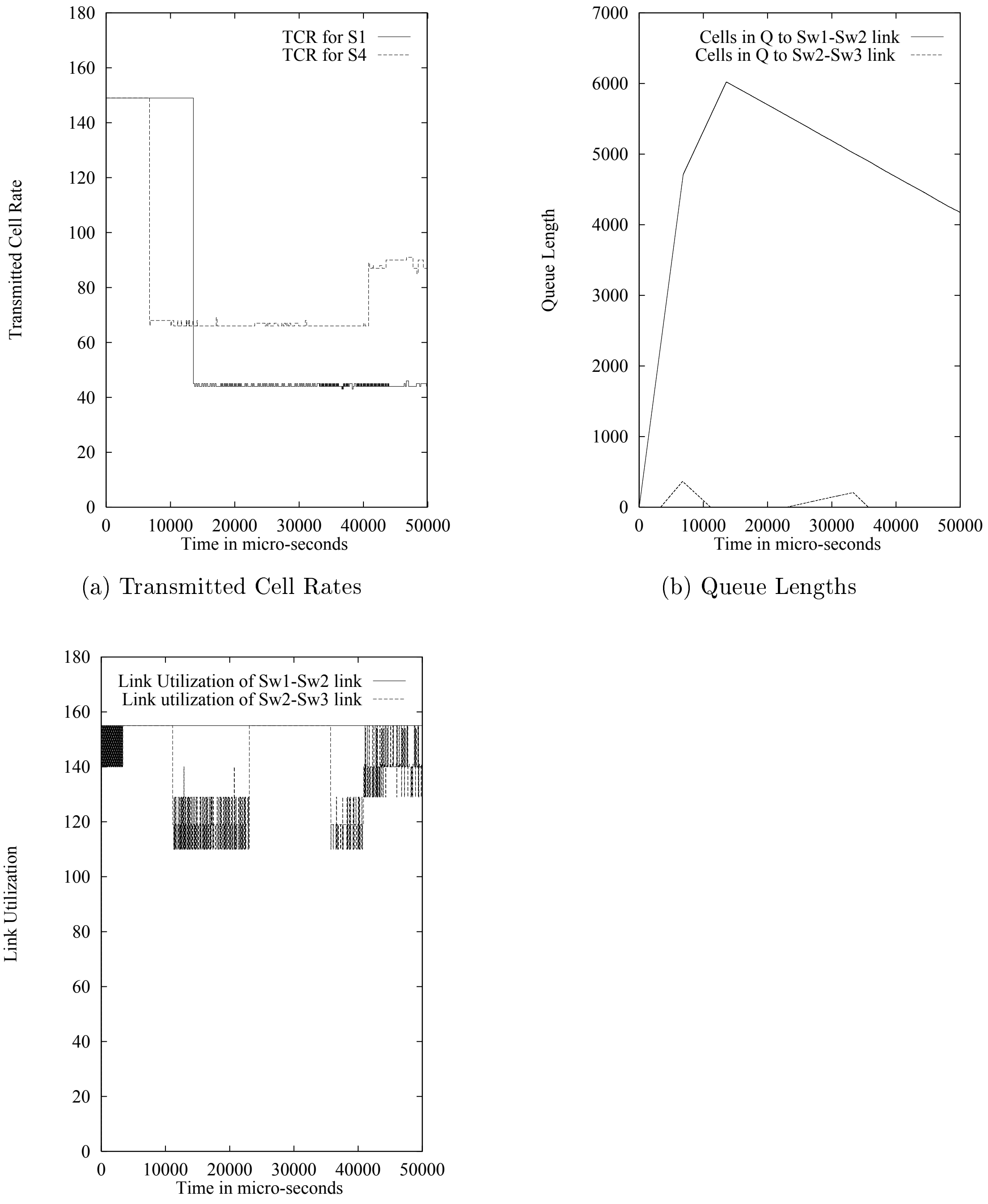


(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 27: Simulation results for the upstream bottleneck configuration with the precise fair share computation options.



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 28: Simulation results for the upstream bottleneck configuration with the precise fair share computation options.

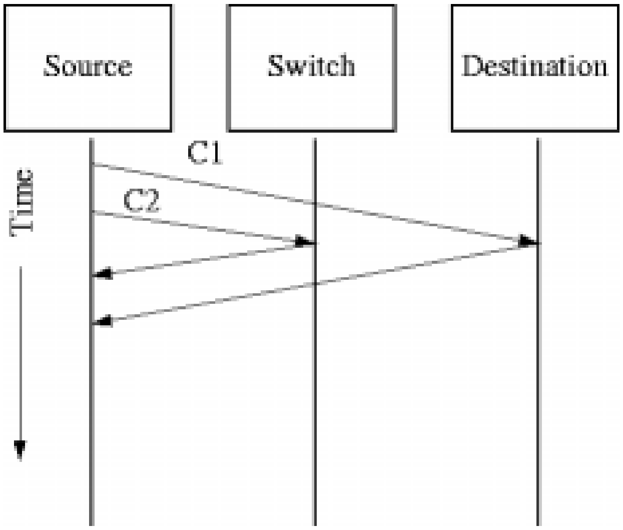
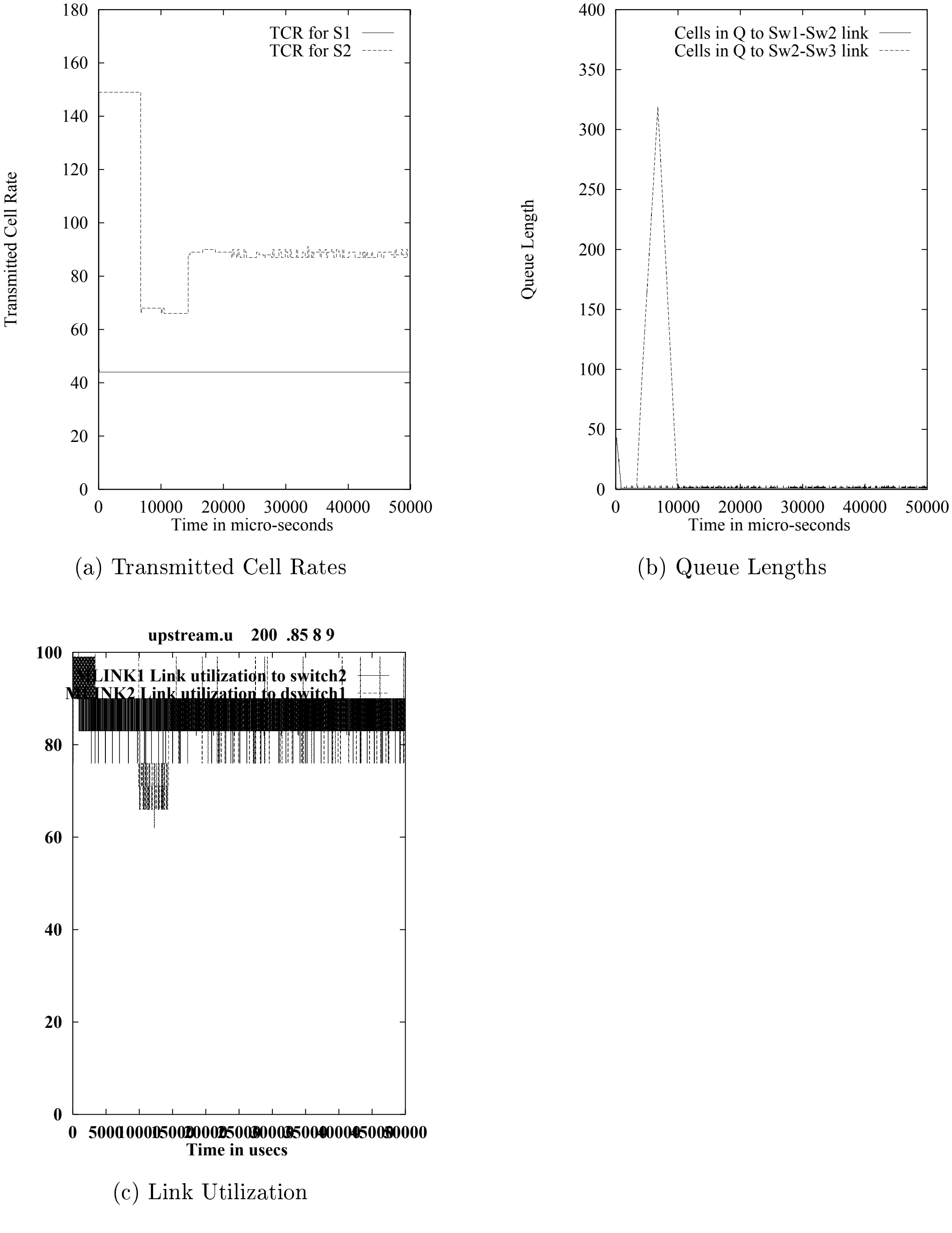


Figure 29: Space time diagram showing out-of-order feedback with BECN



(a) Transmitted Cell Rates

(b) Queue Lengths

(c) Link Utilization

Figure 30: Simulation results for the upstream configuration with the BECN option

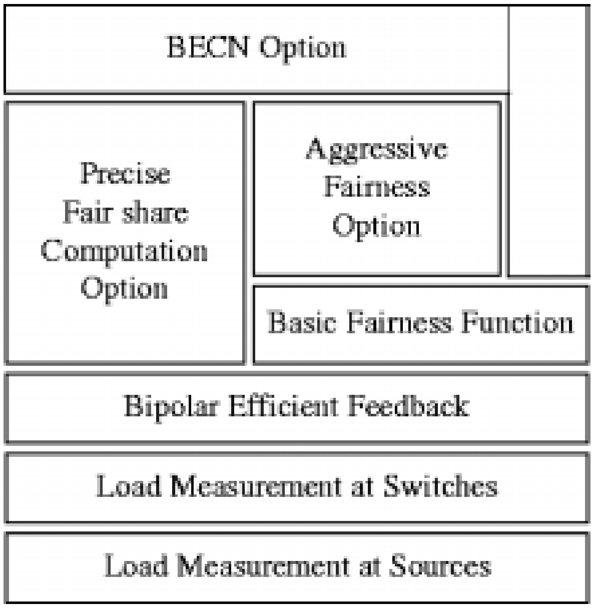

BECN Option
Precise Fair share Computation Option
Aggressive Fairness Option
Basic Fairness Function
Bipolar Efficient Feedback
Load Measurement at Switches
Load Measurement at Sources

Figure 31: A layered view of various components and options of the OSU scheme.

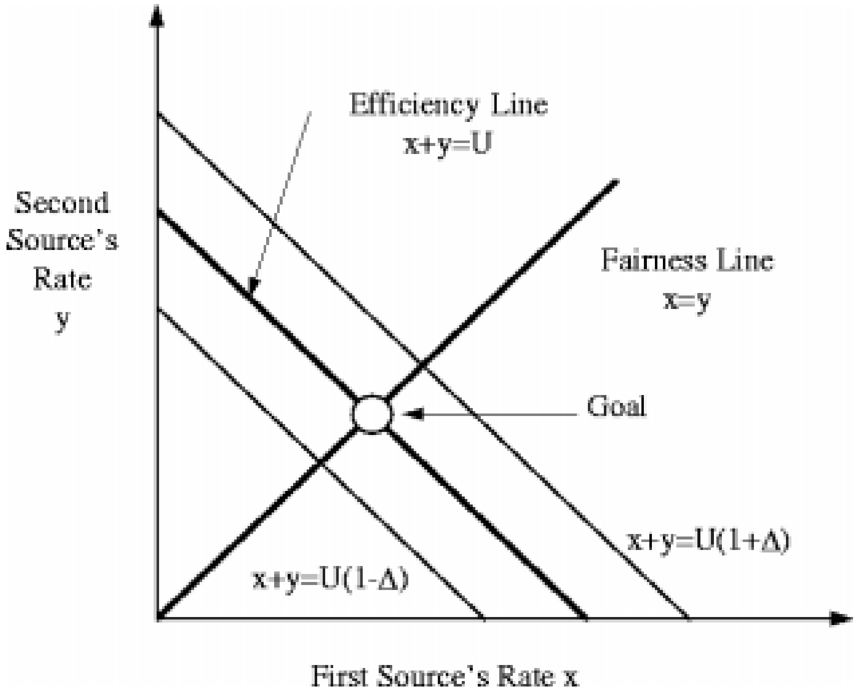


Figure 32: A geometric representation of efficiency and fairness for a link shared by two sources.

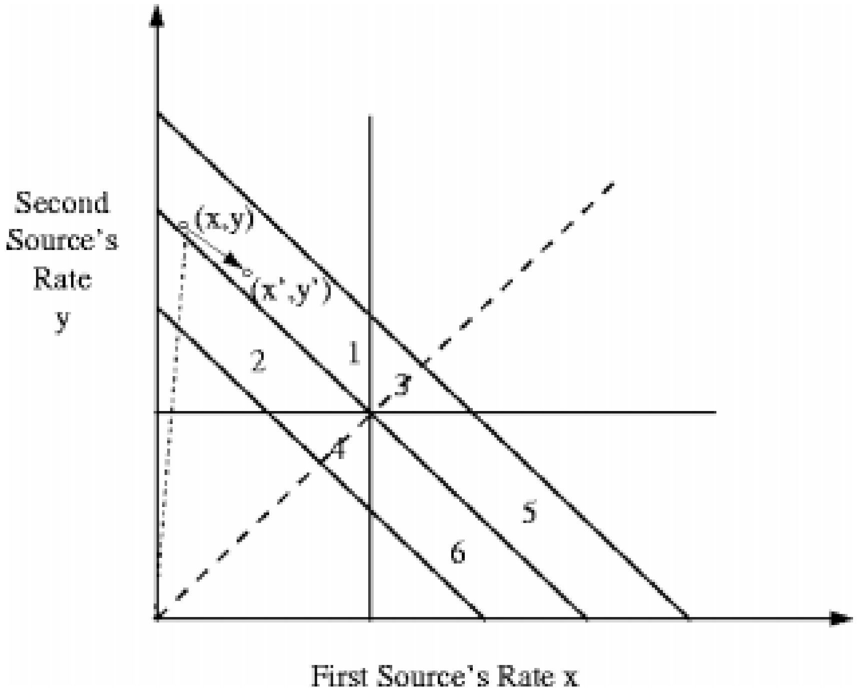


Figure 33: Subregions of the efficient operation zone.

Decisions:

1. Use control cell in place of marked cell or RM cell
2. Use fair share in place of advertised rate
3. Use desired rate in place of stamped rate
4. Use Transmitted Cell Rate (TCR), Offered Average Cell Rate (OCR)

Alphabet soup for the cell rates:

1. ACR=Actual/average/allowed cell rate (confusing)
2. DCR=Desired cell rate
3. ECR=Emitted cell rate
4. GCR=Granted cell rate
5. LCR=Link cell rate
6. MCR=Minimum cell rate
7. OCR=Offered average cell rate
8. PCR=Peak cell rate
9. SCR=Sustained cell rate
10. TCR=Transmitted cell rate

Action Items

1. Simulate dynamic capacity changes