CHAPTER 6

THE ERICA AND ERICA+ SCHEMES

The ERICA scheme is built upon the ideas of the OSU scheme (described in chapter 5. The key limitations of the OSU scheme were the incompatibility with current ATM Forum Traffic Management 4.0 standards [35], and the long time taken to converge to steady state (transient response) from arbitrary initial conditions in complex configurations.

The ERICA and ERICA+ schemes overcome the limitations of the OSU scheme, while keeping the attractive features. Further, they are optimistic algorithms which allocate rates to optimize for both the transient performance, as well as the steady state performance. Since real networks are in a transient state most of the time (sources starting and stopping, ABR capacity varying constantly), we believe that a scheme deployed in real-world switches need to perform well under both transient and steady state conditions.

This chapter is organized as follows. Section 6.1 describes the basic ERICA algorithm. Modifications of this basic algorithm are then presented one by one. The simulation results and performance evaluation are described in section 6.22, while the pseudocode for the algorithm can be found in appendix C.

6.1 The Basic ERICA Algorithm

The switch periodically monitors the load on each link and determines a load factor, z, the available capacity, and the number of currently active virtual connections or VCs (N). The load factor is calculated as the ratio of the measured input rate at the port to the target capacity of the output link.

$$z \leftarrow \frac{\text{ABR Input Rate}}{\text{ABR Capacity}}$$

where ABR Capacity←Target Utilization (U) × Link Bandwidth.

The Input Rate is measured over an interval called the switch averaging interval. The above steps are executed at the end of the switch averaging interval.

Target utilization (U) is a parameter which is set to a fraction (close to, but less than 100 %) of the available capacity. Typical values of target utilization are 0.9 and 0.95.

The load factor, z, is an indicator of the congestion level of the link. High overload values are undesirable because they indicate excessive congestion; so are low overload values which indicate link underutilization. The optimal operating point is at an overload value equal to one. The goal of the switch is to maintain the network at unit overload.

The fair share of each VC, $FairShare$, is also computed as follows:

$$\text{FairShare} \leftarrow \frac{\text{ABR Capacity}}{\text{Number of Active Sources}}$$

The switch allows each source sending at a rate below the $FairShare$ to rise to $FairShare$ every time it sends a feedback to the source. If the source does not use all of its $FairShare$, then the switch fairly allocates the remaining capacity to the sources which can use it. For this purpose, the switch calculates the quantity:

$$\text{VCShare} \leftarrow \frac{CCR}{z}$$

If all VCs changed their rate to their $VCShare$ values then, in the next cycle, the switch would experience unit overload (z equals one). Hence $VCShare$ aims at bringing the system to an efficient operating point, which may not necessarily be fair, and $FairShare$ allocation aims at ensuring fairness, possibly leading to overload (inefficient operation). A combination of these two quantities is used to rapidly reach optimal operation as follows:

$$\text{ER Calculated} \leftarrow \text{Max (FairShare, VCShare)}$$

Sources are allowed to send at a rate of at least $FairShare$ within the first round-trip. This ensures minimum fairness between sources. If the $VCShare$ value is greater than the $FairShare$ value, the source is allowed to send at $VCShare$, so that the link is not underutilized. This step also allows an unconstrained source to proceed towards its max-min rate. The previous step is one of the key innovations of the ERICA scheme because it improves fairness at every step, even under overload conditions.

The calculated ER value cannot be greater than the ABR Capacity which has been measured earlier. Hence, we have:

$$\text{ER Calculated} \leftarrow \text{Min (ER Calculated, ABR Capacity)}$$

To ensure that the bottleneck ER reaches the source, each switch computes the minimum of the ER it has calculated as above and the ER value in the RM cell. This value is inserted in the ER field of the RM cell:
ER in RM Cell←Min(ER in RM cell, ER Calculated).

A flow chart of the basic algorithm is presented in figure C.1 (see appendix C). The flow chart shows steps to be taken on three possible events: at the end of an averaging interval, on receiving a cell (data or RM), and on receving a backward RM cell. These steps have been numbered for reference in further modifications of the basic scheme.

6.2 Achieving Max-Min Fairness

Assuming that the measurements do not suffer from high variance, the above algorithm is sufficient to converge to efficient operation in all cases and to the max-min fair allocations in most cases. The convergence from transient conditions to the desired operating point is rapid, often taking less than a round trip time.

However, we have discovered cases in which the basic algorithm does not converge to max-min fair allocations. This happens if all of the following three conditions are met:

1. The load factor z becomes one

2. There are some sources which are bottlenecked elsewhere upstream

3. CCR for all remaining sources is greater than the $FairShare$

If this happens, then the system remains in its current state, because the term CCR/z is greater than $FairShare$ for the non-bottlenecked sources. This final state may or may not be fair in the max-min sense.

To achieve max-min fairness, the basic ERICA algorithm is extended by remembering the highest allocation made during one averaging interval and ensuring that all eligible sources can also get this high allocation. To do this, we add a variable $MaxAllocPrevious$ which stores the maximum allocation given in the previous interval, and another variable $MaxAllocCurrent$ which accumulates the maximum allocation given during the current switch averaging interval. The step 9 of the basic algorithm is replaced by the flow chart shown in figure C.2 (see appendix C).

Basically, for $z > 1 + \delta$, where δ is a small fraction, we use the basic ERICA algorithm and allocate the source Max (FairShare, VCShare). But, for $z \leq 1 + \delta$, we attempt to make all the rate allocations equal. We calculate the ER as Max (FairShare, VCShare, MaxAllocPrevious).

The key point is that the $VCShare$ is only used to achieve efficiency. The fairness can be achieved only by giving the contending sources equal rates. Our solution attempts to give the sources equal allocations during underload and then divide the (equal) CCRs by the same z during the subsequent overload to bring them to their max-min fair shares. The system is considered to be in a state of overload when its load factor, z, is greater than $1 + \delta$. The aim of introducing the quantity δ is to force the allocation of equal rates when the overload is fluctuating around unity, thus avoiding unnecessary rate oscillations. The next subsection examines one further modification to the ERICA algorithm.

6.3 Fairshare First to Avoid Transient Overloads

The inter-RM cell time determines how frequently a source receives feedback. It is also a factor in determining the transient response time when load conditions change. With the basic ERICA scheme, it is possible that a source which receives feedback first can keep getting rate increase indications, purely because it sends more RM cells before competing sources can receive feedback. This results in unnecessary spikes (sudden increases) in rates and queues with the basic ERICA scheme.

The problem arises when the Backward RM (BRM) cells from different sources arrive asynchronously at the switch. Consider a LAN configuration of two sources (A and B), initially sending at low rates. When the BRM arrives, the switch calculates the feedback for the current overload. Without loss of generality, assume that the BRM of source A is encountered before that of source B. Now it is possible that the BRM changes the rate of source A and the new overload due to the higher rate of A is experienced at the switch before the BRM from the source B reaches the switch. The transient overload experienced at the switch may still be below unity, and the ACR of source A is increased further (BRMs for source A are available since source A sends more RM cells at higher rates). This effect is observed as an undesired spike in the ACR graphs and sudden queue spikes when the source B gets its fair share.

This problem can be solved by incorporating the following change to the ERICA algorithm. When the calculated ER is greater than the fair share value, and the source is increasing from a CCR below $FairShare$, we limit its increase to $FairShare$. Alternatively, the switch could decide not to give new feedback to this source for one measurement interval. The following computation is added to the switch algorithm.

After "ER Calculated" is computed:

```
IF ((CCR $<$ FairShare) AND (ER Calculated $\geq$ FairShare)) THEN
    ER Calculated $\leftarrow$FairShare
```

We can also disable feedback to this source for one measurement interval.

"ER in RM Cell" is then computed as before.

6.4 Forward CCR Used for Reverse Direction Feedback

Earlier schemes [51] provided their feedback to the RM cells going in the forward direction. This ensured that the CCR in the RM cell was correlated to the load level measured by the switch during that interval. However, the time taken by the forward going RM cell to travel back to the source was long and this slowed down the response of the system.

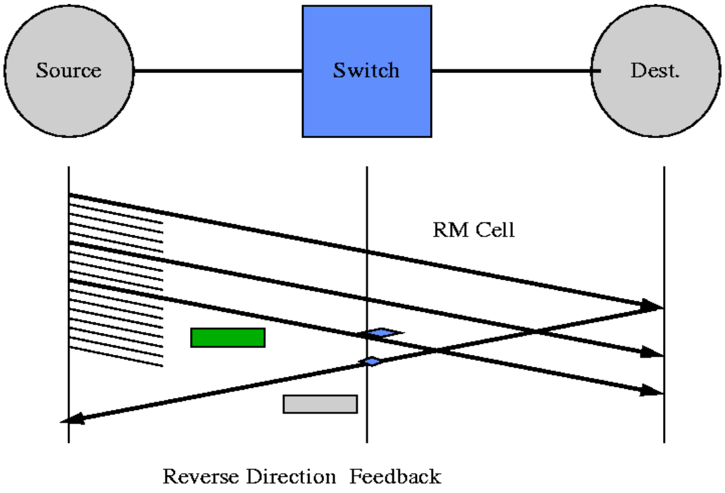


Figure 6.1: Reverse direction feedback

Switches can indicate their feedback to the sources in the reverse path of the RM cell. The backward going RM (BRM) cell takes less time to reach the source than the forward going RM (FRM) cell which has to reach the destination first. Thus, the system responds faster to changes in the load level. However, the CCR carried by the BRM cell no longer reflects the load level in the system. To maintain the most current CCR value, the switch copies the CCR field from FRM cells, and uses this information to compute the ER value to be inserted in the BRM cells. This ensures that the latest CCR information is used in the ER calculation and that the feedback path is as short as possible. Figure 6.1 shows that the first RM cell carries (in its backward path), the feedback calculated from the information in the most recent FRM cell. The CCR table update and read operations still preserve the O(1) time complexity of the algorithm.

6.5 Single Feedback in a Switch Interval

The switch measures the overload, the number of active sources and the ABR capacity periodically (at the end of every switch averaging interval). The source also sends RM cells periodically. These RM cells may contain different rates in their CCR fields. If the switch encounters more than one RM cell from the same VC during the same switch interval, then it uses the same value of overload for computing feedback in both cases. For example, if two RM cells from the same VC carried different CCR values, then the feedback in one of them will not accurately reflect the overload. As a result, the switch feedback will be erroneous and may result in unwanted rate oscillations. The switch thus needs to give only one feedback value per VC in a single switch interval.

The above example illustrates a fundamental principle in control theory, which says that the system is unstable when the control is faster than feedback. But the system is unresponsive if the control is slower than feedback. Ideally, the control rate should be matched to the feedback rate. In our system, the delay between successive feedbacks should not be greater than the delay between successive measurements (controls).

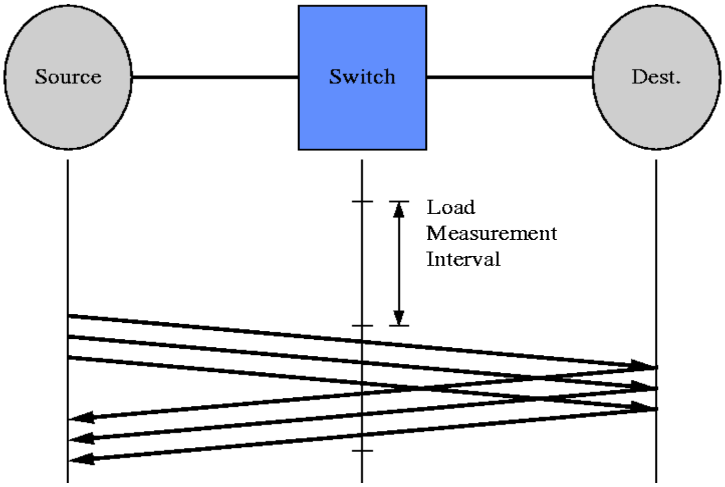


Figure 6.2: Independence of source and switch intervals

The switch provides only one feedback value during each switch interval irrespective of the number of RM cells it encounters. The switch calculates the ER only once per interval, and the ER value obtained is stored. It inserts the same ER value in all the RM cells it sees during this interval. In figure 6.2, the switch interval is greater than the RM cell distance. The ER calculated in the interval marked *Load Measurement Interval* is maintained in a table and set in all the RM cells passing through the switch during the next interval.

6.6 Per-VC CCR Measurement Option

The CCR of a source is obtained from the CCR field of the forward going RM cell. The latest CCR value is used in the ERICA computation. It is assumed that the CCR is correlated with the load factor measured. When the CCR is low, the frequency of forward RM cells becomes very low. Hence, the switch may not have a new CCR estimate though a number of averaging intervals have elapsed. Moreover, the CCR value may not be an accurate measure of the rate of the VC if the VC is bottlenecked at the source, and is not able to use its ACR allocation. Note that if a VC is bottlenecked on another link, the CCR is set to the bottleneck allocation within one round-trip.

A possible solution to the problems of inaccurate CCR estimates is to measure the CCR of every VC during the same averaging interval as the load factor. This requires the switch to count the number of cells received per VC during every averaging interval and update the estimate as follows:

At the *end of an switch averaging interval:*

```
FOR ALL VCs DO
    CCR[VC] ←NumberOfCells[VC]/IntervalLength
    NumberOfCells[VC] ←0
END
```

When a *cell is received:*

```
NumberOfCells[VC] ←NumberOfCells[VC] + 1
```

Initialization:

```
FOR ALL VCs DO NumberOfCells[VC] ←0
```

When an *FRM cell is received,* do not copy CCR field from FRM into CCR[VC].

Note that using this method, the switch ignores the CCR field of the RM cell. The per-VC CCR computation can have a maximum error of (one cell/averaging interval) in the rate estimate. Hence the error is minimized if the averaging interval is larger.

The effect of the per VC CCR measurement can be explained as follows. The basic ERICA uses the formula: ER Calculated←Max (FairShare, VCShare).

The measured CCR estimate is always less than or equal to the estimate obtained from the RM cell CCR field. If the other quantities remain constant, the term "VCShare" decreases. Thus the ER calculated will decrease whenever the first term dominates. This change results in a more conservative feedback, and hence shorter queues at the switches.

6.7 ABR Operation with VBR and CBR in the Background

The discussion so far assumed that the entire link was being shared by ABR sources. Normally, ATM links will be used by constant bit rate (CBR) and variable bit rate (VBR) traffic along with ABR traffic. In fact, CBR and VBR have a higher priority. Only the capacity left unused by VBR and CBR is given out to ABR sources. For such links, we need to measure the CBR and VBR usage along with the input rate. The ABR capacity is then calculated as follows:

$$\text{ABR Capacity} \leftarrow \text{Target Utilization} \times \text{Link Bandwidth} - \text{VBR Usage} - \text{CBR Usage}$$

The rest of ERICA algorithm remains unchanged. Notice that the target utilization is applied to the entire link bandwidth and not the the left over capacity. That is,

$$\text{ABR Capacity} \neq \text{Target Utilization} \times \{\text{Link Bandwidth} - \text{VBR Usage} - \text{CBR Usage}\}$$

There are two implications of this choice. First, (1-Target Utilization) ×(Link Bandwidth) is available to drain the queues, which is much more than what would be available otherwise. Second, the sum of VBR and CBR usage must be less than

(Target Utilization)×(Link Bandwidth). Thus, the VBR and CBR allocation should be limited to below the target utilization.

6.8 Bi-directional Counting of Bursty Sources

A bursty source sends data in bursts during its active periods, and remains idle during other periods. It is possible that the BRM cell of a bursty source could be traveling in the reverse direction, but no cells of this source are traveling in the forward direction. A possible enhancement to the counting algorithm is to also count a source as active whenever a BRM of this source is encountered in the reverse direction. We refer to this as the "bidirectional counting of active VCs".

One problem with this technique is that the reverse queues may be small and the feedback may be given before the $FairShare$ is updated, taking into consideration the existence of the new source. Hence, when feedback is given, we check to see if the source has been counted in the earlier interval and if the $FairShare$ has been updated based upon the existence of the source. If the source had not been counted, we update the number of active sources and the $FairShare$ before giving the feedback. This option is called "the immediate fairshare update option" in the flow chart of figure C.3 (see appendix C).

We could also reset the CCR of such a source to zero after updating the $FairShare$ value, so that the source is not allocated more than the $FairShare$ value. The motivation behind this strategy is that the source may be idle, but its CCR is unchanged because no new FRMs are encountered. When the per-VC CCR measurement is used, this option is not necessary, because the switch measures the CCRs periodically. The setting of CCR to zero is a conservative strategy which avoids large queues due to bursty or ACR retaining sources. A drawback of this strategy is that in certain configurations, the link may not be fully utilized if the entire traffic is bursty. This is because all the bursty sources are asked to send at $FairShare$, which may not be the optimal value if some sources are bottlenecked elsewhere. This option can also be enabled and disabled based upon a certain queue threshold.

6.9 Averaging of the Number of Sources

Another technique to overcome the problem of underestimating the number of active sources is to use exponential averaging to decay the contribution of each VC to the number of active sources count. The main motivation behind this idea is that if a source is inactive during the current interval, but was recently active, it should still contribute to the number of active sources. This is because this source might be sending its data in bursts, and just happened to be idle during the current interval.

Flow charts of figures C.4 and C.5 show this technique (see appendix C).

The $DecayFactor$ used in decaying the contribution of each VC is a value between zero and one, and is usually selected to be a large fraction, say 0.9. The larger the value of the $DecayFactor$, the larger the contribution of the sources active in prior intervals, and the less sensitive the scheme is to measurement errors. Setting the $DecayFactor$ to a smaller fraction makes the scheme adapt faster to sources which become idle, but makes the scheme more sensitive to the averaging interval length.

6.10 Boundary Cases

Two boundary conditions are introduced in the calculations at the end of the averaging interval. First, the estimated number of active sources should never be less

ABR Capacity	Input Rate	Overload	Fairshare	CCR/Overload	Feedback
Zero	Non-zero	Infinity	Zero	Zero	Zero
Non-zero	Zero	Infinity	C/N	Zero	C/N
Non-zero	Non-zero	I/C	C/N	CCR×C/I	Max (CCR×C/I, C/N)
Zero	Zero	Infinity	Zero	Zero	Zero

Table 6.1: Boundary Cases

than one. If the calculated number of sources is less than one, the variable is set to one. Second, the load factor becomes infinity when the ABR capacity is measured to be zero, and the load factor becomes zero when the input rate is measured to be zero. The corresponding allocations are described in Table 6.1.

6.11 Averaging of the Load Factor

In cases where no input cells are seen in an interval, or when the ABR capacity changes suddenly (possibly due to a VBR source going away), the overload measured in successive intervals may be considerably different. This leads to considerably different feedbacks in successive intervals. An optional enhancement to smoothen this variance is by averaging the load factor. This effectively increases the length of the averaging interval over which the load factor is measured.

One way to accomplish this is shown in the flow chart of figure C.6 (see appendix C).

The method described above has the following drawbacks. First, the average is reset everytime z becomes infinity. The entire history accumulated in the average prior to the interval where the load is to be infinity is lost.

For example, suppose the overload is measured in successive intervals as: 2, 1, Infinity, 3, Infinity, 0.5. The method previously described forgets the history in the fourth interval, and restarts at the new value 3. Similarly in the sixth interval, it restarts at the value 0.5. Note that this introduces dependencies between the boundary cases and the average value of the load factor.

The second problem with this method is that the exponential average does not give a good indication of the average value of quantities which are not additive. In our case, the load factor is not an additive quantity. However, the number of ABR cells received or output is additive.

The load factor is a ratio of the input rate and the ABR capacity. The correct way to average a ratio is to find the ratio of the average (or the sum) of the numerators and divide it by the average (or the sum) of the denominators. That is, the average of $x_1/y_1, x_2/y_2, \ldots, x_n/y_n$ is $(x_1 + x_2 + \ldots + x_n)/(y_1 + y_2 + \ldots + y_n)$.

To average load factor, we need to average the input rate (numerator) and the ABR capacity (denominator) separately. However, the input rate and the ABR capacity are themselves ratios of cells over time. The input rate is the ratio of number of cells input and the averaging interval. If the input rates are $x_1/T_1, x_2/T_2, \ldots, x_n/T_n$, the average input rate is $((x_1 + x_2 + \ldots + x_n)/n)/((T_1 + T_2 + \ldots + T_n)/n)$. Here, x_i's are the number of ABR cells input in averaging interval i of length T_i. Similarly the average ABR capacity is $((y_1 + y_2 + \ldots + y_n)/n)/((T_1 + T_2 + \ldots + T_n)/n)$, where y_i's are the maximum number of ABR cells that can be output in averaging interval i of length T_i.

The load factor is the ratio of these two averages. Observe that each of the quantities added is not a ratio, but a number. Exponential averaging is an extension of arithmetic averaging used above. Averages such as $(x_1 + x_2 + \ldots x_n)/n$ can be replaced by the exponential average of the variable x_i.

The flow chart of figure C.7 describes this averaging method.

Observe that the load factor thus calculated is never zero or infinity unless the input rate or ABR capacity are always zero. If the input rate or the ABR capacity is measured to be zero in any particular interval, the boundary cases for overload are not invoked. The load level increases or decreases to finite values.

6.12 Time and Count Based Averaging

The load factor, available ABR capacity and the number of active sources need to be measured periodically. There is a need for an interval at the end of which the switch renews these quantities for each output port. The length of this interval determines the accuracy and the variation of the measured quantities. As mentioned before, longer intervals provide lower variation but result in slower updating of information. Alternatively, shorter intervals allow fast response but introduce greater variation in the response. This section proposes alternative intervals for averaging the quantities.

The averaging interval can be set as the time required to receive a fixed number of ABR cells (M) at the switch in the forward direction. While this definition is sufficient to correctly measure the load factor and the ABR capacity at the switch, it is not sufficient to measure the number of active VCs (N) or the CCR per VC accurately. This is because the quantities N and CCR depend upon the fact that at least one cell from the VC is encountered in the averaging interval. Moreover, when the rates are low, the time to receive M cells may be large. Hence the feedback in the reverse direction may be delayed.

An alternative way of averaging the quantities is by a fixed time interval, T. This ensures that any source sending at a rate greater than (one cell/T) will be encountered in the averaging interval. This interval is independent of the number of sources, but is dependent upon the minimum rate of the source. In addition to this, if the aggregate input rate is low, the fixed-time interval is smaller than the fixed-cells interval. However, when there is an overload, the fixed-cells interval provides faster response.

One way of combining these two kinds of intervals is to use the minimum of the fixed-cell interval and the fixed-time interval. This combination ensures quick response for both overload and underload conditions. But it still suffers from the disadvantages of a fixed-cell interval, where N and per-VC CCR cannot be measured accurately [84].

Another strategy for overcoming this limitation is to measure N and per-VC CCR over a fixed-time interval, and the capacity and load factor over the minimum of the fixed-cell and fixed-time interval. The time intervals can be different as long as some correlation exists between the quantities measured over the different intervals. Typically, the intervals to measure CCR and N would be larger to get more stable estimates.

6.13 Selection of ERICA Parameters

Most congestion control schemes provide the network administrator with a number of parameters that can be set to adapt the behavior of the schemes to their needs. A good scheme must provide a small number of parameters that offer the desired level of control. These parameters should be relatively insensitive to minor changes in network characteristics.

ERICA provides a few parameters which are easy to set because the tradeoffs between their values are well understood. Our simulation results have shown that slight mistuning of parameters does not significantly degrade the performance of the scheme. Two parameters are provided: the target Utilization (U) and the switch measurement interval.

6.13.1 Target Utilization U

The target utilization determines the link utilization during steady state conditions. If the input rate is greater than Target Utilization × Link Capacity, then the switch asks sources to decrease their rates to bring the total input rate to the desired fraction. If queues are present in the switch due to transient overloads, then $(1 - U) \times$ Link Capacity is used to drain the queues.

Excessively high values of target utilization are undesirable because they lead to long queues and packet loss, while low target utilization values lead to link underutilization. The effectiveness of the value of target utilization depends on the feedback delay of the network. Transient overloads can potentially result in longer queues for networks with longer feedback delays. Due to this, smaller target utilization values are more desirable for networks with long propagation delays.

Our simulation results have determined that ideal values of target utilization are 0.95 and 0.9 for LANs and WANs respectively. Smaller values improve the performance of the scheme when the traffic is expected to be highly bursty.

6.13.2 Switch Averaging Interval AI

The switch averaging or measurement interval determines the accuracy of feedback. This interval is used to measure the load level, link capacity and the number of active VCs for an outgoing link. The length of the measurement interval establishes a tradeoff between accuracy and steady state performance. This tradeoff has been briefly discussed in section 6.5.

ERICA measures the required quantities over an averaging interval and uses the measured quantities to calculate the feedback in the next averaging interval. Averaging helps smooth out the variation in the measurements. However, the length of the averaging interval limits the amount of variation which can be eliminated. It also determines how quickly the feedback can be given to the sources, because ERICA gives at most one feedback per source per averaging interval. Longer intervals produce better averages, but slow down the rate of feedback. Shorter intervals may result in more variation in measurements, and may consistently underestimate the measured quantities.

The load factor and available capacity are random variables whose variance depends on the length of the averaging interval. In practice, the interval required to measure the number of active sources is sufficient for the measurement of the load factor and available capacity. Both of these averaged quantities are fairly accurate, with an error margin of (one cell/averaging interval). Setting the target utilization below 100% helps drain queues due to errors in measurement of all the quantities. Whenever the scheme faces tradeoffs due to high errors in measurement, the degree of freedom is to reduce the target utilization parameter, sacrificing some steady state utilization for convergence.

6.14 ERICA+: Queue Length as a Secondary Metric

ERICA+ is a further modification of ERICA. In this and the following section, we describe the goals, target operating point, the algorithm, and parameter settings for ERICA+.

ERICA depends upon the measurement of metrics such as the overload factor and the number of active ABR sources. If there is a high error in the measurement, and the target utilization is set to very high values, ERICA may diverge, i.e., the queues may become unbounded, and the capacity allocated to drain the queues becomes insufficient. The solution in such cases is to set the target utilization to a smaller value, allowing more bandwidth to drain queues. However, steady state utilization (utilization when there is no overload) is reduced because it depends upon the target utilization parameter.

A simple enhancement to ERICA is to have a queue threshold, and reduce the target utilization if the queue exceeds the threshold. Once the target utilization is low, the queues are drained out quickly. Hence, this enhancement maintains high utilization when the queues are small, and drains out queues quickly when they become large. Essentially, we are using the queue length as a secondary metric (input rate is the primary metric).

In ERICA, we have not considered the queue length or queue delay as a possible metric. In fact, we rejected it because it gives no indication of the correct rates of the sources. In ERICA+, we maintain that the correct rate assignments depend upon the aggregate input rate, rather than the queue length. However, we recognize two facts about queues: a) non-zero queues imply 100% utilization, and, b) a system with very long queues is far away from the intended operating point. Hence, in ERICA+, if the input rates are low and the queues are long, we recognize the need to reserve more capacity to drain the queues and allocate rates conservatively till the queues are controlled. Further, keeping in line with the design principles of ERICA, we use continuous functions of the queue length, rather than discontinuous functions. Since feedback to sources is likely to be regular (as long as queues remain), the allocations due to a continuous function in successive averaging intervals track the behavior of the queue and reflect it in the rate allocations.

6.15 ERICA+: 100% Utilization and Quick Drain of Queues

ERICA achieves high utilization in the steady state, but utilization is limited by the target utilization parameter. For expensive links, it is desirable to keep the steady state utilization at 100%. This is because a link being able to service 5% more cells can translate into 5% more revenue. The way to get 100% utilization in steady state, and quick draining of queues is to vary the target ABR rate dynamically. During steady state, the target ABR rate is 100% while it is lower during transient overloads. Higher overloads result in even lower target rates (thereby draining the queues faster). In other words:

Target rate = function (queue length, link rate, VBR rate)

The "function" above has to be a decreasing function of the queue length.

Note that ERICA has a fixed target utilization, which means that the drain rate is independent of the queue size.

6.16 ERICA+: Maintain a "Pocket" of Queues

The ABR capacity varies dynamically, due to the presence of higher priority classes (CBR and VBR). Hence, if the higher priority classes are absent for a short interval (which may be smaller than the feedback delay), the remaining capacity is not utilized. In such situations, it is useful to have a "pocket" full of ABR cells which use the available capacity while the RM cells are taking the "good news" to the sources and asking them to increase their rates.

One way to achieve this effect is to control the queues to a "target queue length." In the steady state, the link is 100% utilized, and the queue length is equal to the target queue length, which is the "pocket" of queues we desire. If the queue length falls below this value, the sources are encouraged to increase their rate and vice versa. In other words:
Target rate = function (queue length, target queue length, link rate, VBR rate)

6.17 ERICA+: Scalability to Various Link Speeds

The above function is not scalable to various link speeds because the queue length measured in cells translates to different drain times for different transmission speeds. For example, a queue length of 5 at a T1 link may be considered large while a queue length of 50 at an OC-3 link may be considered small. This point is significant due to the varying nature of ABR capacity, especially in the presence of VBR sources.

To achieve scalability, we need to measure all queue lengths in units of time rather than cells. However, the queue is the only directly measurable quantity at the switch. The queueing delay is then estimated using the measured ABR capacity value. The above function for target rate becomes:

Target rate = function (queue delay, target queue delay, link rate, VBR rate)

In the following sections, we define and describe a sample function to calculate the target rate.

6.18 ERICA+: Target Operating Point

ERICA+ uses a new target operating point which is in the middle of the "knee" and the "cliff," as shown in figure 3.2. The new target operating point has 100% utilization and a fixed non-zero queueing delay. This point differs from the *knee* point (congestion avoidance: 100% throughput, minimum delay) in that it has a fixed non-zero delay goal. This is due to non-zero queueing delay at the operating point. Note that the utilization remains 100% as long as the queue is non-zero. The utilization remains at 100% even if there are short transient underloads in the input load, or the output capacity increases (appearing as an underload in the input load).

We note that non-zero queue values in steady state imply that the system is in an unstable equilibrium. Queues grow immediately during transient overloads. In contrast, ERICA could allow small load increases (5 to 10%) without queue length increases.

The challenge of ERICA+ is to maintain the unstable equilibrium of non-zero queues and 100% utilization. Specifically, when the queueing delay drops below the target value, $T0$, ERICA+ increases allocation of VCs to reach the optimum delay. Similarly, when the queueing delay increases beyond $T0$, the allocation to VCs is reduced and the additional capacity is used for queue drain in the next cycle. When the queueing delay is $T0$, 100% of the ABR capacity is allocated to the VCs. ERICA+, hence, introduces a new parameter, $T0$, in place of the target utilization parameter of ERICA.

6.19 The ERICA+ Scheme

As previously mentioned, the ERICA+ scheme is a modification of the ERICA scheme. In addition to the suggested scheduling method between VBR and ABR classes, the following are the changes to ERICA.

1. The link utilization is no longer targeted at a constant *Target Utilization* as in ERICA. Instead, the total ABR capacity is measured given the link capacity and the VBR bandwidth used in that interval: $Total\ ABR\,Capacity + VBR\,Capacity = Link\ Capacity$

2. The target ABR capacity is a fraction of the total ABR capacity

$$Target\ ABR\,Capacity \leftarrow f(T_q) \times Total\ ABR\,Capacity$$

This function must satisfy the following constraints:

1. It must have a value greater than or equal to 1 when the queueing delay, T_q is 0 (zero queues). This allows the queues to increase and T_q can go up to $T0$, the threshold value. A simple choice is to keep the value equal to one. The queue increases due to the slight errors in measurement. Another alternative is to have a linear function, with a small slope. Note that, we should not use an aggressive increase function. Since queueing delay is a highly variant quantity, a small variation in delay values may cause large changes in rate allocations, and hence lead to instability.

2. It must have a value less than 1 when the queueing delay, T_q is greater than $T0$. This forces the queues to decrease and T_q can go down to $T0$. Since queue increases are due to traffic bursts, a more aggressive control policy is required for this case compared to the former case where we project a higher capacity than available. Since we project a lower capacity than what is available, the remaining capacity is used to drain the queues.

3. If the queues grow unboundedly, then we would like the function to go to zero. Since zero, or very low, ABR capacity is unacceptable, we place a cutoff on the capacity allocated to queue drain. The cutoff is characterized by a parameter, called the queue drain limit factor (QDLF). A value of 0.5 for the QDLF parameter is sufficient in practice.

4. When the queueing delay, T_q is $T0$ we want f $(T_q) = 1$.

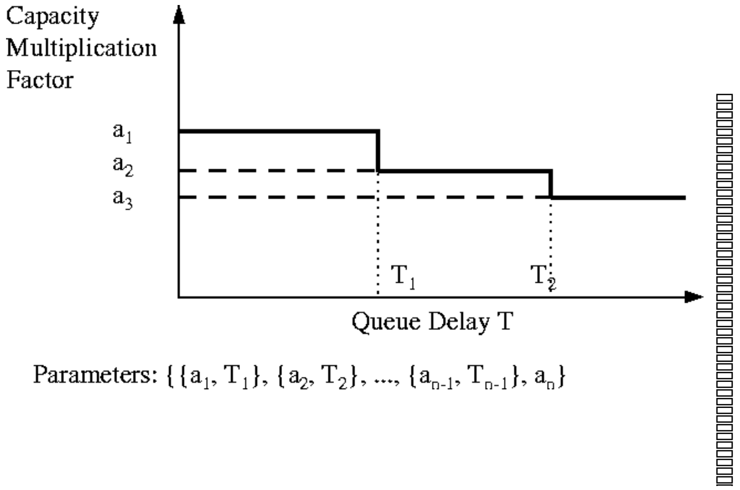


Figure 6.3: Step functions for ERICA+

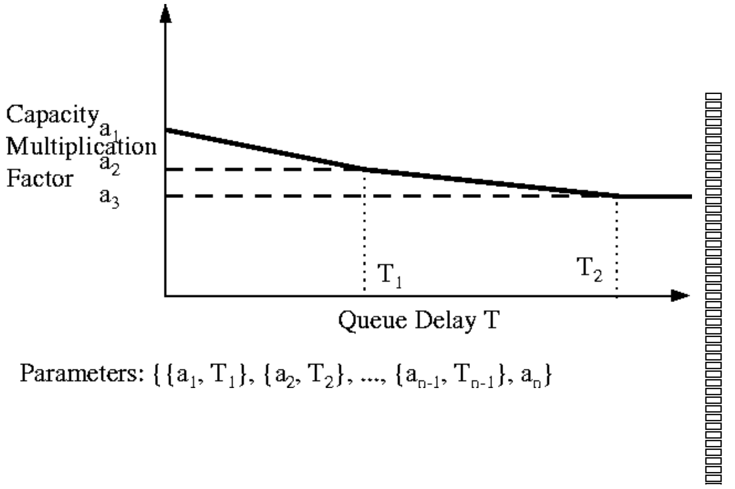


Figure 6.4: Linear functions for ERICA+

A step function which reduces the capacity in steps (down to the cutoff value) as the queueing delay exceeds thresholds is a possible choice. This is shown in figure 6.3. Linear segments as shown in figure 6.4 can be used in place of step functions. Hysteresis thresholds (figure 6.5) can be used in place of using a single threshold to increase and decrease the capacity. Hysteresis implies that we use one threshold to increase the capacity and another to decrease the capacity. However, these functions require the use of multiple thresholds (multiple parameters). Further, the thresholds are points of discontinuity, i.e., the feedback given to the source will be very different if the system is on the opposite sides of the threshold. Since queueing delay is a highly variant quantity, the thresholds and experience is required to choose these different parameters.

However, it is possible to have a function with just 2 parameters, one for the two ranges: (0, Q0) and (Q0, infinity) respectively. The rectangular hyperbolic and the negative exponential functions are good choices to provide the aggressive control

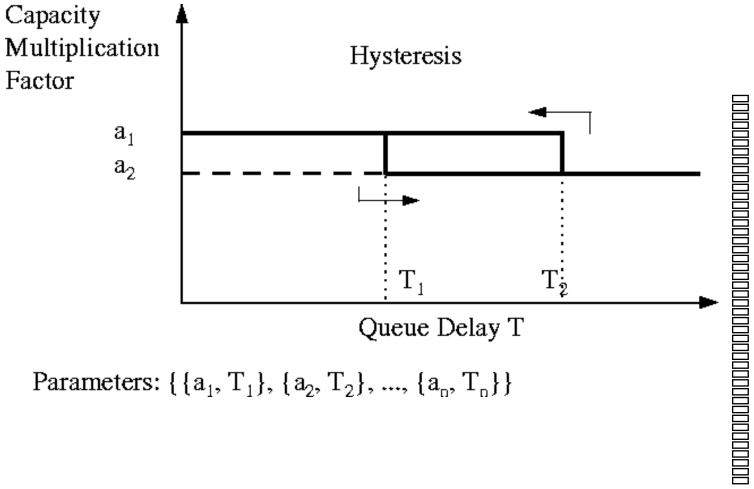


Figure 6.5: Hysteresis functions for ERICA+

required when the queues grow. We choose the former which is the simpler of the two.

Since the portion $T < T0$ requires milder control, we can have a different hyperbola for that region. This requires an extra parameter for this region. The queue control scheme uses a time (queueing delay) as a threshold value. Hence, depending upon the available capacity at the moment, this value $T0$ translates into a queue length Q0, as follows:

$$Q0 = Total\ ABR\, Capacity \times T0$$

In the following discussion, we will refer to Q0 and queues alone, but Q0 is a variable dependent upon available capacity. The fixed parameter is $T0$. The queue control function, as shown in figure 6.6, is:

$$f(T_q) = \frac{a \times Q0}{(a-1) \times q + Q0}\ for\, q > Q0$$

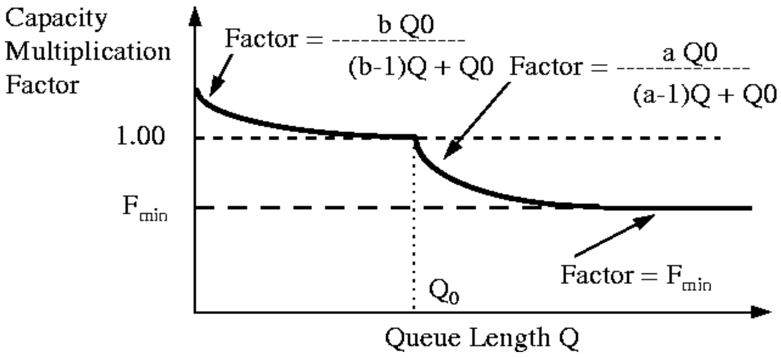

Capacity Multiplication Factor
Factor = b Q0 / (b-1)Q + Q0
Factor = a Q0 / (a-1)Q + Q0
1.00
F_{min}
Factor = F_{min}
Q_0
Queue Length Q

Figure 6.6: The queue control function in ERICA+

and

$$f(T_q) = \frac{b \times Q0}{(b-1) \times q + Q0} \ for\ 0 \leq q \leq Q0$$

Note that $f(T_q)$ is a number between 1 and 0 in the range Q0 to infinity and between b and 1 in the range 0 to Q0. Both curves intersect at Q0, where the value is 1. These are simple rectangular hyperbolas which assume a value 1 at Q0. This function is lower bounded by the queue drain limit factor (QDLF):

$$f(T_q) = Max(QDLF, \frac{a \times Q0}{(a-1) \times q + Q0}) \ for\ q > Q0$$

6.20 Effect of Variation on ERICA+

ERICA+ calculates the target ABR capacity, which is the product of $f(T_q)$ and the ABR capacity. Both these quantities are variant quantities (random variables), and the product of two random variables (say, A and B) results in a random variable which has more variance than either A or B. Feedback becomes less reliable as the variance increases.

For example, overload depends upon the ABR capacity and is used in the formula to achieve max-min fairness. Since the ERICA+ algorithm changes the ABR capacity depending upon the queue lengths, this formula needs to tolerate minor changes in load factor. In fact, the formula applies hysteresis to eliminate the variation due to the load factor. Since techniques like hysteresis and averaging can tolerate only a small amount of variation, we need to reduce the variance in the target ABR capacity.

We examine the ABR capacity term first. ABR capacity is estimated over the averaging interval of ERICA. A simple estimation process can entail counting the VBR cells sent, calculating the VBR capacity, and subtracting it from the link capacity. This process may have an error of one VBR cell divided by the averaging interval length. The error can be minimized by choosing longer averaging intervals.

However, the measured ABR capacity has less variance than instantaneous queue lengths. This is because averages of samples have less variance than the samples themselves, and ABR capacity is averaged over an interval, whereas queue length is not. The quantity $Q0 = T0 \times ABRCapacity$ has the same variance as that of the measured ABR capacity.

We now examine the function, $f(T_q)$. This function is bounded below by $QDLF$ and above by b. Hence, its values lie in the range (QDLF,b) or, in practice, in the range (0.5, 1.05). Further, it has variance because it depends upon the queue length, q and the quantity $Q0$. Since the function includes a ratio of $Q0$ and q, it has higher variance than both quantities.

One way to reduce the variance is to use an averaged value of queue length (q), instead of the instantaneous queue length. A simple average is the mean of the queue lengths at the beginning and the end of a measurement interval. This is sufficient for small averaging intervals. If the averaging interval is long, a better average can be obtained by sampling the queue lengths during the interval and taking the average of the samples. Sampling of queues can be done in the background.

Another way to reduce variation is to specify a constant $Q0$. This can be specified instead of specifying $T0$ if a target delay in the range of $[\frac{Q0}{MinimumABRcapacity}, \frac{Q0}{MaximumABRcapacity}]$ is acceptable.

6.21 Selection of ERICA+ Parameters

The queue control function in ERICA+ has four parameters: $T0$, a, b, and $QDLF$. In this section, we explain how to choose values for the parameters and discuss techniques to reduce variation in the output of the function.

The function $f(T_q)$ has three segments: (1) a hyperbola characterized by the parameter b (called the b-hyperbola) between queueing delay of zero and $T0$, (2) an a-hyperbola from a queueing delay of $T0$ till $f(T_q)$ equals $QDLF$, (3) $QDLF$. Hence, the range of the function $f(T_q)$ is $[QDLF, b]$.

6.21.1 Parameters a and b

a and b are the intercepts of the a-hyperbola and b-hyperbola, i.e., the value of $f(T_q)$ when $q = 0$. b determines how much excess capacity would be allocated when the queueing delay is zero. a and b also determine the slope of the hyperbola, or, in other words, the rate at which $f(T_q)$ drops as a function of queueing delay. Larger values of a and b make the scheme very sensitive to the queueing delay, whereas, smaller values increase the time required to reach the desired operating point.

The parameter b is typically smaller than a. b determines the amount of overallocation required to reach the target delay $T0$ quickly in the steady state. Any small over-allocation above 100% of ABR capacity is sufficient for this purpose. The parameter a primarily determines how quickly the function $f(T_q)$ drops as a function of queueing delay. a should not be very different from b because, this can result in widely different allocations when the delay slightly differs from $T0$. At the same time, a should be high enough control the queues quickly.

Through simulation, we found that the values 1.15 and 1.05 for a and b respectively work well for all the workloads we have experimented with. Hence, at zero queues, we over-allocate up to 5% excess capacity to get the queues up to $Q0$. Higher values of b would allow sources to overload to a higher extent. This can aggravate transient overloads and result in higher queue spikes. Using a value of 1 for b is also acceptable, but the "pocket" of queues builds up very slowly in this case. A value of 1 for b is preferable when the variance is high. Further, these parameters values for a and b are relatively independent of $T0$ or $QDLF$. Given these values for a and b, the function depends primarily on the choice of $T0$ and $QDLF$ as discussed below.

6.21.2 Target Queueing Delay $T0$

When the function $f(T_q)$ is one of the two hyperbolas, its slope ($\frac{df}{dq}$) is inversely proportional to the parameter $T0$. For a constant value of a, larger $T0$ reduces the slope of the function, and hence its effectiveness. The queueing delay required to reduce the ABR capacity by a fixed fraction is directly proportional to $T0$. It is also directly proportional to the ABR capacity. Hence, if the ABR capacity is high (as is the case in OC-3 and higher speed networks), the queues need to build up to a large value before the drain capacity is sufficient. Hence, the maximum value of $T0$ depends upon and how fast the transient queues need to be cleared.

The maximum value of $T0$ also depends on the buffer size at the switch, and must be set to allow the control of the queues before the buffer limit is reached. One strategy is to keep the buffer size at least the sum of the feedback delay and $8 \times T0$ (assuming a = 1.15 and $QDLF = 0.5$, and ABR capacity is constant, and other factors like measurement interval length are negligible). One feedback delay is enough for the feedback to reach the sources and $8 \times T0$ is enough for the function to reach $QDLF$. For other values of $QDLF$, the recommended buffer size is:

$$\frac{(a - QDLF) \times T0}{[(a-1) \times QDLF]}$$

The maximum value of $T0$ can be calculated reversing the above formula, given the buffer size.

$$T0 = \frac{[(a-1) \times QDLF]}{(a - QDLF)}$$

A minimum value of $T0$ is also desired for stable operation. If $T0$ is very small, the function $f(T_q)$ can traverse the range $[QDLF, b]$ in a time $\frac{(a-QDLF)\times T0}{[(a-1)\times QDLF]}$, assuming that capacity is constant over this period of time. This time can be shorter than the feedback delay, and lead to undesired oscillations in rates and queues. This is because the function changes from b to $QDLF$ before feedback is effective. Such a behavior is undesired because, the scheme now is very sensitive to the changes in queue length. Recall that queue length is only a secondary metric, i.e., we want the input rate and not the queue length to be the primary metric of congestion. Further, the minimum $T0$ is at least the "pocket" of queues desired. For WANs, $T0$ is at least $\frac{[(a-1)\times QDLF]}{(a-QDLF)}$ of the feedback delay, which is 1/8, assuming $a = 1.15$, $QDLF = 0.5$. For LANs, we set $T0$ to at least one feedback delay, to reduce the sensitivity of the ABR capacity to small queue lengths. In cases of high variation and measurement errors, the "pocket" of queues may not be achievable. High throughput is the goal in this case, and $T0$ should be set close to the minimum value to allow queues to be quickly drained.

6.21.3 Queue Drain Limit Factor $QDLF$

$QDLF$ ensures that there is enough capacity to drain out the transient queues. We recommend a value of 0.5 for WAN switches and 0.8 for LAN switches.

WAN switches need to have greater drain capacity because of the longer feedback delays of its VCs and consequently longer response times to transient overloads. If the fluctuations in load or capacity are of a time-scale much smaller than the feedback delay, the rate allocations using a high target rate may not be sufficient. Transient queues may build up in such cases unless there is sufficient capacity allocated to drain the queues. An example of such high variation workload is TCP traffic combined with a VBR load which has an ON-OFF period of 1 ms, whereas the feedback delay is 10 ms.

However, for LAN switches which can receive feedback rapidly, and $T0$ is small, the function can move quickly through the range $[QDLF, b]$. Given these conditions, a large drain capacity is not required, since large queues never build up. For such configurations, $QDLF$ can have higher values like 0.8.

Since the $QDLF$ parameter defines the lower bound of the function $f(T_q)$, we should ensure that this value is reached only for large queue values. This can be achieved by choosing small values for a, or large values for $T0$. Since large values of $T0$ reduce the effectiveness of the function $f(T_q)$, the parameter a is chosen small. This is another factor in the choice of a. It turns out that the recommended value for a (1.15) is small enough for the $QDLF$ values recommended.

6.22 Performance Evaluation of the ERICA and ERICA+ Schemes

In this section, we shall describe the methodical performance evaluation of the ERICA scheme, and provides simple benchmarks to test the performance of different ATM switch algorithms. We use the principles discussed in chapter 3 to test ERICA for various configurations, source models and background traffic patterns.

We present the set of experiments conducted to ensure that ERICA meets all the requirements of switch algorithms. In the cases where the original algorithm failed to meet the requirements and an enhancement to the algorithm was deemed necessary, the performance of the basic algorithm is compared to the performance of the enhanced algorithm, and a discussion of why the enhancement was needed is presented.We prefer to use simple configurations when applicable because they are more instructive in finding problems [66]. The results are presented in the form of four graphs for each configuration:

1. Graph of allowed cell rate (ACR) in Mbps over time for each source

2. Graph of ABR queue lengths in cells over time at each switch

3. Graph of link utilization (as a percentage) over time for each link

4. Graph of number of cells received at the destination over time for each destination

We will examine the efficiency and delay requirements, the fairness of the scheme, its transient and steady state performance, and finally its adaptation to variable capacity and various source traffic models. The experiments will also be selected such that they have varying distances and number of connections to examine the scalability requirement.

6.22.1 Parameter Settings

Throughout our experiments, the following parameter values are used:

1. All links have a bandwidth of 155.52 Mbps.

2. All LAN links are 1 Km long and all WAN links are 1000 Km long.

3. All VCs are bidirectional.

4. The source parameter Rate Increase Factor (RIF) is set to one, to allow immediate use of the full Explicit Rate indicated in the returning RM cells at the source.

5. The source parameter Transient Buffer Exposure (TBE) is set to large values to prevent rate decreases due to the triggering of the source open-loop congestion control mechanism. This was done to isolate the rate reductions due to the switch congestion control from the rate reductions due to TBE.

6. The switch target utilization parameter was set at 95% for LAN simulations and at 90% for WAN simulations.

7. The switch averaging interval was set to the minimum of the time to receive 50 cells and 1 ms for LAN simulations, and to the minimum of the time to receive 100 cells and 1 ms for WAN simulations.

8. The ERICA+ parameters are set as follows. The parameters a and b (intercepts of the two hyperbolas in the queueing delay function) are set to 1.15 and 1.05

respectively. The target delay $T0$ is set at 100 microseconds for LAN simulations and 500 microseconds for WAN simulations, and the queue drain limit factor ($QDLF$) is set at 0.5.

9. All sources, including VBR sources are deterministic, i.e., their start/stop times and their transmission rates are known. The bursty traffic sources send data in bursts, where each burst starts after a request has been received from the client.

6.22.2 Efficiency

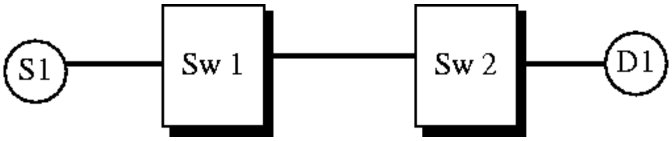


Figure 6.7: One source configuration

The very first test to verify efficient operation is to use a single source configuration as shown in figure 6.7. A scheme that does not work for this simple configuration is not worth further analysis. The source is active over the entire simulation period. Figure 6.11 illustrates that ERICA achieves the required efficiency, since the source rate rises to almost fully utilize the link. Observe that there are no rate oscillations in the steady state, and that utilization is at the target utilization goal (95%).

The same configuration has also been simulated to examine the efficiency of ERICA+. As seen in figure 6.12, the source rate rises to fully utilize the link (100%

utilization) with no oscillations and minimal queues. Please note that the simulation graphs, though introduced periodically, are at the end of the chapter.

6.22.3 Minimal Delay and Queue Lengths

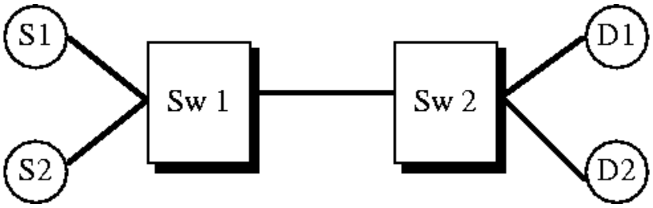


Figure 6.8: Two source configuration

To test for minimal delays and short queue lengths, we use a multiple source configuration. The simplest such configuration is the two source configuration, where two sources share a link as illustrated in figure 6.8. Each source must converge to almost half of the link rate (1/2 × Target Utilization), which is the max-min optimal allocation.

Figure 6.13 shows that the convergence is fast, the queue lengths are small (hence, the delay is minimal) and steady state performance is good. For ERICA+, the two sources rapidly converge to their optimal rates as seen in figure 6.14, and the queue length rises to reach the limit corresponding to its target delay parameter (100 microseconds corresponds to approximately 30 cells at 155.52 Mbps). There is a slight rate oscillation seen in figure 6.14(a) to allow the queues to reach the target value, but the steady state has no rate oscillations and 100% link utilization (figure 6.14(c)).

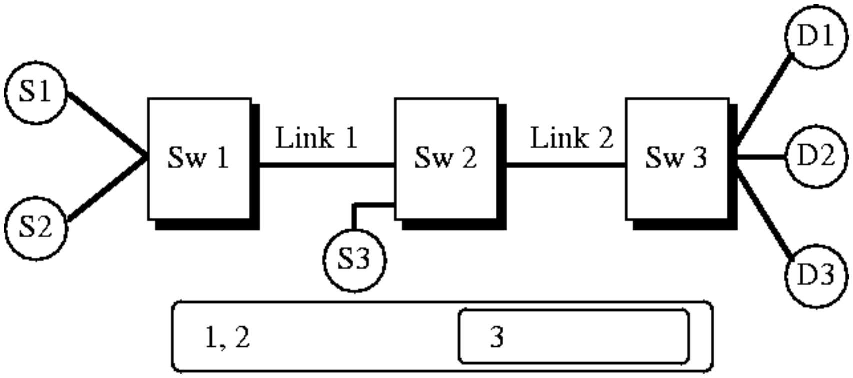


Figure 6.9: Parking lot configuration

6.22.4 Fairness

Two configurations are used for studying the fairness of the scheme: the parking lot configuration and the upstream configuration. The parking lot configuration and its name were 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 series. There are n VCs. The first VC starts from the first switch and goes through all the remaining switches. For the remaining VCs, the i^{th} VC starts at the $i - 1^{st}$ switch. The link between the last two switches is the bottleneck link. The max-min allocation for each VC is $1/n$ of the bandwidth. A 3-switch parking lot configuration is shown in figure 6.9. Figure 6.15 illustrates that ERICA achieves the desired max-min allocation.

Although the parking lot configuration had been believed to test fairness, we discovered that it is not sufficient to demonstrate max-min fairness, and a more stringent test is required. We had observed that the original ERICA algorithm does not converge to max-min fairness in certain situations. Such situations arise when the ERICA algorithm is executed in a state where some of the sources cannot fully utilize their allocated bandwidth on a link (for example, because they are bottlenecked on another link), and the rest of the sources contending for bandwidth have unequal CCR values, which are greater than the fair share value (first term in the maximum formula). The ERICA algorithm does not converge to max-min fairness in these situations because, after z converges to one, the second term in the maximum formula becomes $CCR_i/1 = CCR_i$, and the first term is constant. The maximum of the two terms for the contending sources is the second term, because there are sources that are not fully utilizing their allocated bandwidth. Hence, the sources do not change their rates.

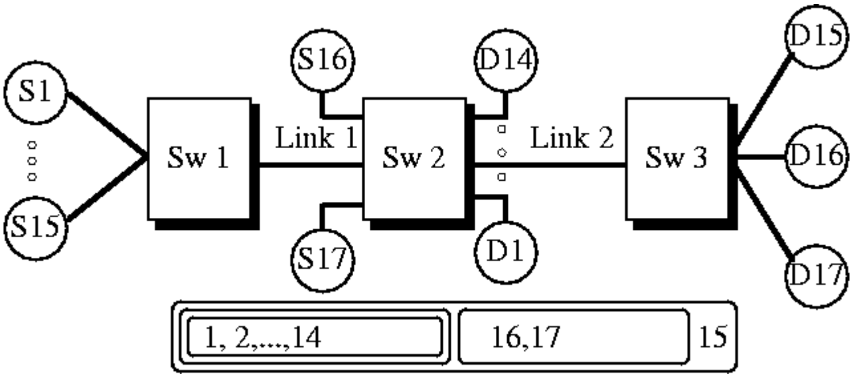


Figure 6.10: Upstream Configuration

An example of this situation can be illustrated by an upstream configuration (see figure 6.10). The upstream configuration consists of three switches and 17 VCs. The second link is shared by VC_{15}, VC_{16}, and VC_{17}. Because there are 15 VCs on the first link, VC_{15} is limited to a throughput of less than 1/15 the link rate. VC_{16} and VC_{17} should, therefore, each converge to a little less than 7/15 of the second link rate. The configuration is called an upstream configuration because the bottleneck link is the first link (upstream link). A WAN configuration (1000 Km links) is used in this situation to illustrate the scalability of ERICA to long distances.

Figure 6.17 shows that the original ERICA algorithm was unfair in this situation, and figure 6.18 shows that ERICA, after the modification discussed in section 6.2, is fair. As seen in figure 6.18, the modified algorithm converges to max-min allocations. Regardless of the initial load factor value, after a short transient period, all sources contending for bandwidth are allocated equal rates, and the two curves in figure 6.18(b) (number of cells received at the destination) have the same slope (compare with figure 6.17(b)). The transient response is slightly worse than the original ERICA algorithm due to the temporary over-allocation needed to equalize the shares, but the steady state performance is as good as with the original ERICA algorithm.

6.22.5 Transient and Steady State Performance

To test the transient response of the system, we use a modified two source configuration. The configuration is similar to the two source configuration because two sources share the same link, but one of the sources is only active from 10 ms to 20 ms while the other source is active throughout. Besides illustrating the transient response of the system, this configuration also illustrates the effect of the "fairshare first" algorithm discussed in section 6.3. That algorithm (see section 6.3) prevents a low rate VC to rise above $FairShare$. This VC takes an extra round trip compared to the basic ERICA because it first comes to $FairShare$ before rising further. The switch can use the extra round trip to give feedback to all the sources, measure a new load factor and reduce overloading sources. The modification reduces the maximum queues in transient situations.

Figures 6.19 and 6.20 illustrate the effect of the "fairshare first" modification on a transient configuration in a LAN. Figure 6.19 shows the transient performance of ERICA without the "fairshare first" modification, and figure 6.20 illustrates how ERICA with the "fairshare first" modification avoids transient overloads. It is clear that ERICA exhibits good transient response characteristics to changing load, and the modification mitigates sudden overloads, constraining the queue length when the second source starts transmission. The figure also shows that the steady state performance of the scheme is excellent, as there are minimal oscillations in the rates of the sources.

6.22.6 Adaptation to Variable ABR Capacity

Constant Bit Rate (CBR) and Variable Bit Rate (VBR) service classes have a higher priority than the ABR service. In cases of VBR traffic, the ABR capacity becomes a variable quantity.

The two source configuration in a wide area network is used to demonstrate the behavior of ERICA in the presence of VBR sources. A deterministic VBR source is used whose peak rate is 124.42 Mbps (80% of the link capacity). Figure 6.21 illustrates the behavior of ERICA on a WAN where the VBR source was active for alternating periods of 1 ms with 1 ms inactive periods in between (high frequency VBR), while figure 6.22 shows the performance with VBR on/off periods of 20 ms (low frequency VBR).

From the figures, it is clear that ERICA rapidly detects the change in the available ABR capacity and gives the appropriate feedback to the sources. When the VBR source is active, the ABR sources rapidly reduce their rates (figures 6.21(a) and 6.22(a)). The utilization is generally high; the utilization drops reflect the time taken for the feedback to reach the sources: the feedback delay (figures 6.21(c) and 6.22(c)). The spikes in the queue lengths seen in figures 6.21(b) and 6.22(b) also reflect the feedback delay, but the queues are rapidly drained. Observe that the number of cells received in both cases (figures 6.21(d) and 6.22(d)) is approximately equal, which shows that the performance is approximately the same. The throughput can be calculated from the graphs showing the number of cells received at the destination. It is clear that the throughput is high, indicating a high link utilization.

Figure 6.23 illustrates how ERICA+ adapts to high frequency VBR in the background, and figure 6.24 shows its performance with low frequency VBR. ERICA+ adapts rapidly to the changing background traffic, recomputing the available bandwidth and the rate allocations. The link utilization is higher than that with ERICA, and the queue lengths are constrained. The target queue goal is never reached due to the high variation, but the utilization goal is partially reached.

6.22.7 Adaptation to Various Source Traffic Models

In all the previous experiments, the ABR sources are assumed to be persistent sources, which means that they always have data to send, and can utilize their full capacity at all times. It is essential to examine the performance of the scheme with bursty sources which alternate between active periods when they utilize their full capacity, and idle periods when they do not send any data. In addition, situations where sources are bottlenecked are also of particular interest. The scheme should be able to rapidly react to the overload that can arise if the bottlenecked sources suddenly start utilizing their full capacity.

6.22.8 Bursty Traffic

Figures 6.25 through 6.29 illustrate the performance of ERICA in a wide area network two source configuration where one of sources is a persistent (greedy or infinite) source, while the other connection is a request-response type connection. The request-response connection consists of a source sending a request (of size 16 cells), and the destination responding with a burst of data. Two different burst sizes are used in our simulations: small bursts are 128 cells, and large bursts are 6144 cells. Upon the receipt of the response at the source, and after a certain period of time, the source sends another request for data, and the cycle is repeated (see chapter 7). The figures show the performance of the reverse (response) connection where a burst of data is sent in response to every request.

Figure 6.25 illustrates the performance of ERICA with small response burst sizes, figure 6.26 shows the effect of medium burst sizes, while figure 6.27 illustrates the effect of large burst sizes. As seen in the figures, ERICA can adapt to small and medium bursts of data, and the queue lengths are constrained. However, with a target utilization of 90%, ERICA does not have enough capacity to drain large bursts of data from the switch queues before the next burst is received. This problem can be solved by using smaller values for the target utilization parameter.

Figure 6.28 shows that bi-directional counting of the number of active sources (as discussed in section 6.8) limits the queue sizes for large bursts. This is because it counts the bursty source as active if its RM cells are traveling in the reverse direction, even though it might not be sending any data in the forward direction during its idle periods. This situation is called the "out-of-phase" effect, and is also a common problem with TCP sources. The problem affects the load measurement, as well as the measurement of the number of active VCs. As seen in figure6.28(b), the queue lengths are constrained, and the problem seen in figure 6.27(b) has been solved, even for a target utilization of 90%.

Another method to limit the queue sizes in this case is by averaging the number of active sources as discussed in section6.9. As previously explained, we should account for the presence of a source, even though it might be currently idle. The effect of averaging the value of the number of active sources is illustrated in figure 6.29. The bidirectional counting option is not used in this case. Figure 6.29(b) shows that the queue length is constrained.

Figure6.30, 6.31 and 6.32 illustrates the performance of ERICA+ for the same configuration without the averaging the number of sources option. The figure illustrates the effect of large burst sizes. It is clear that ERICA+ can adapt to bursty traffic better than ERICA, because it accounts for the time to drain the queues when estimating the available capacity. Even with large burst sizes, the queues built up when the bursty source is active can drain before the next burst arrives at the switch. The bidirectional counting and the averaging of number of sources options are not necessary in this case.

In cases of many sources running TCP on ABR in the presence of high frequency VBR background, ERICA+ sometimes fails to drain the queues when the averaging interval parameter is set to very small values. This phenomenon is explained in depth in chapter 8. Averaging the number of sources and averaging the load factor as explained earlier in this chapter can also alleviate this problem.

6.22.9 ACR Retention

ACR retention is the problem which occurs when sources are not able to fully use their rate allocations. For example, the input to the ATM end-system can be steady, but have a rate lower than its ABR allocation (allowed cell rate). Another example is an end-system which supports multiple VCs (to possibly different destinations) on a single outgoing link. A VC may not be able to use its ACR allocation because the outgoing link is running at capacity. In such situations, the switches reallocate the unused capacity to the other sources which are unconstrained. However, if the ACR retaining sources suddenly use their allocations, a potential overload situation exists.

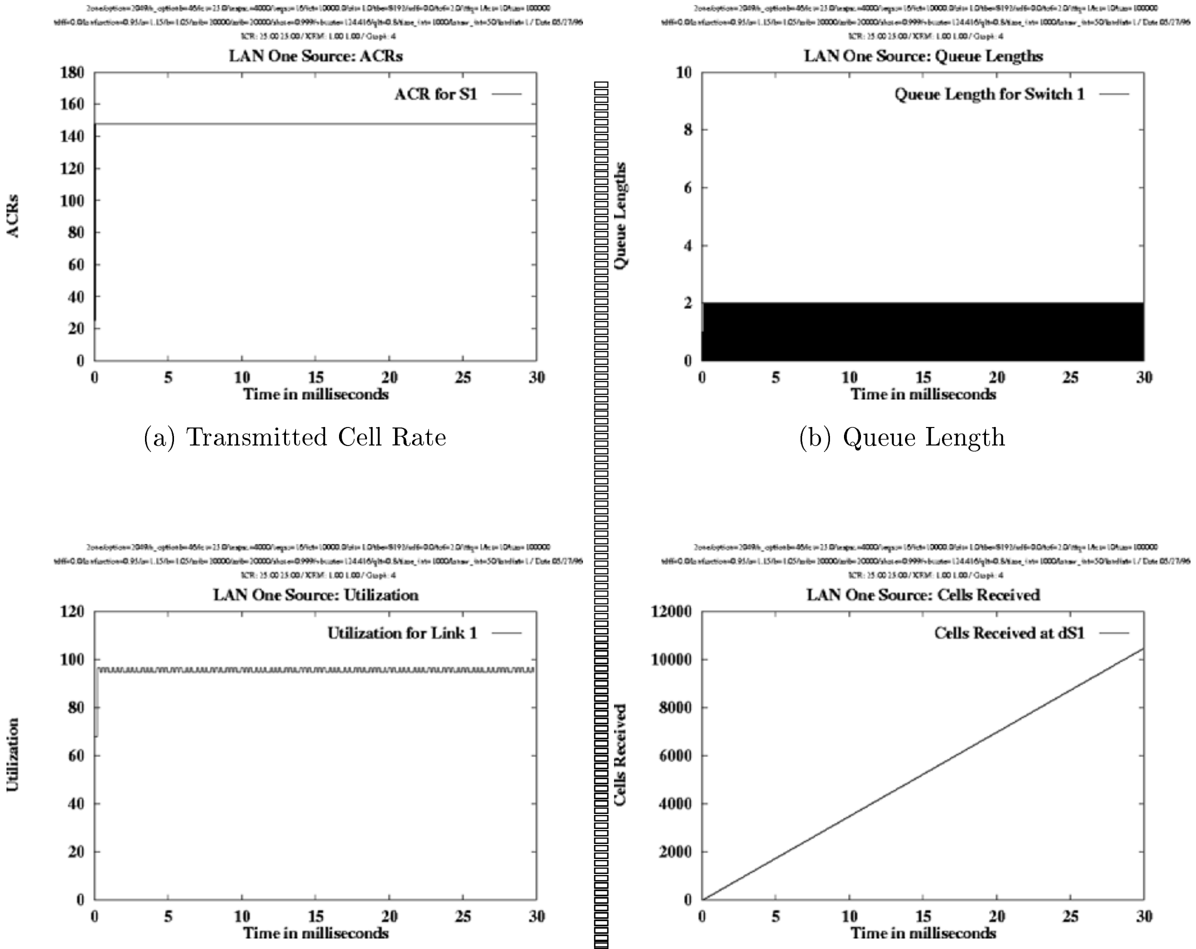
Figure 6.33 illustrates the performance of ERICA when there are ten VCs sharing a link. This larger number of connections has been selected to demonstrate the scalability of ERICA to more VCs, as well as to aggravate the problem of ACR retention. Initially, the ten sources are retaining their ACRs, and each cannot send at a rate of more than 10 Mbps. After 100 ms, all the sources suddenly start sending at their full allocations. ERICA rapidly detects the overload and gives the appropriate feedback asking sources to decrease their rates. All the ten sources stabilize at their optimal rates after that.

Figure 6.34 shows how the per-VC CCR measurement option can mitigate the overload situation arising when all the ACR retaining sources start transmission at their full capacities. The per-VC CCR measurement results in more conservative initial allocations, and hence smaller queues in this case.

6.23 Summary of the ERICA and ERICA+ Schemes

This chapter has examined the ERICA and ERICA+ schemes, explicit rate indication schemes for congestion avoidance in ATM networks, and explained several extensions and enhancements of the scheme. The scheme entails that the switches periodically monitor their load on each link and determine a load factor, the available capacity, and the number of currently active virtual channels. This information is used to calculate a fair and efficient allocation of the available bandwidth to all contending sources. The algorithm exhibits a fast transient response, and achives high utilization and short delays, in addition to adapting to high variation in the capacity and demand.

Based on the discussion of requirements of switch algorithms in chapter 3 we have examined how each of these requirements can be tested. Using these techniques, we presented a comprehensive performance evaluation of the ERICA switch algorithm and demonstrated the effect of several features and options of the algorithm. We have examined the efficiency and delay requirements, the fairness of the scheme, its transient and steady state performance, its scalability, and its adaptation to variable capacity and various source traffic models. Simulation results have illustrated that the algorithm gives optimal allocations, and rapidly adapts to load and capacity changes. The performance of the algorithm was examined for various configurations, source models and background traffic patterns.



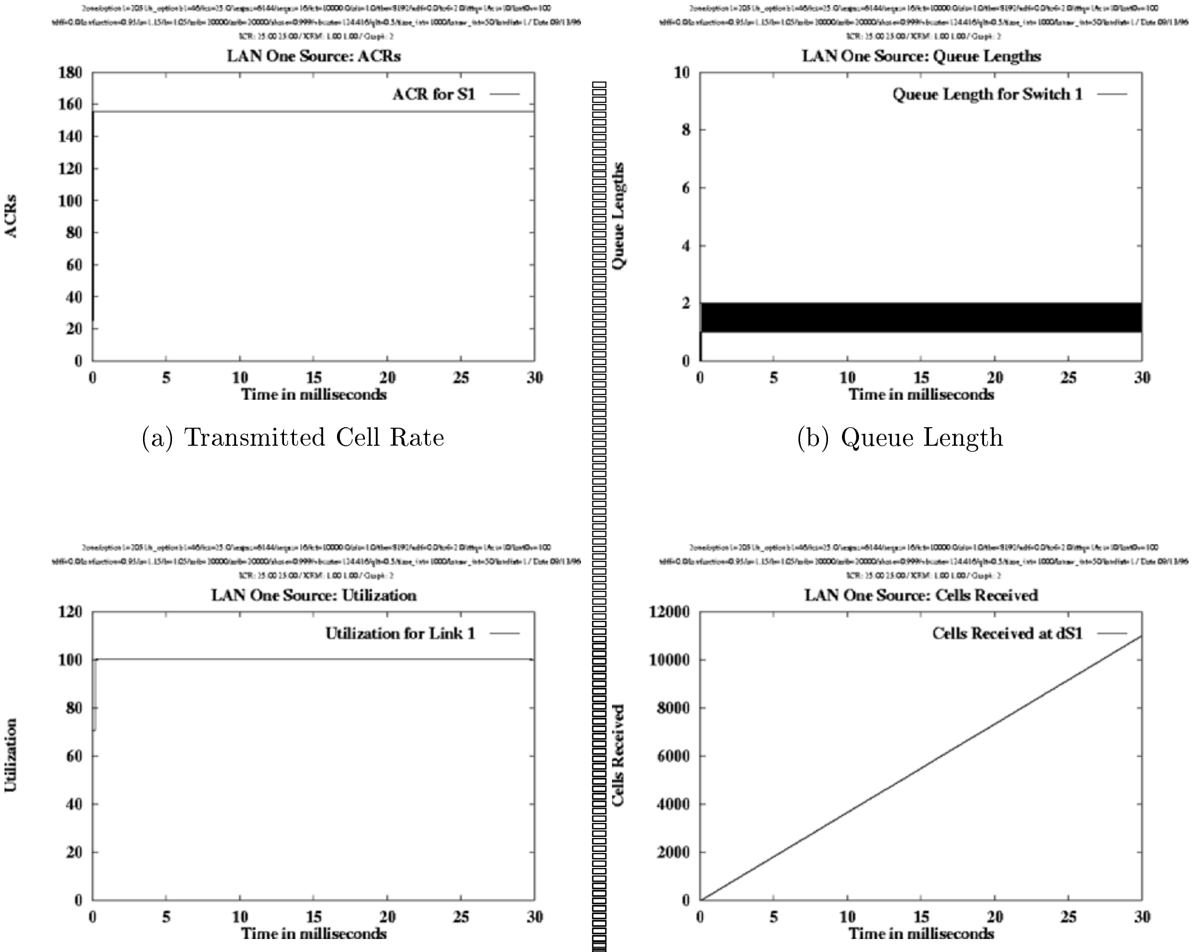
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.11: Results for a one source configuration in a LAN (ERICA)



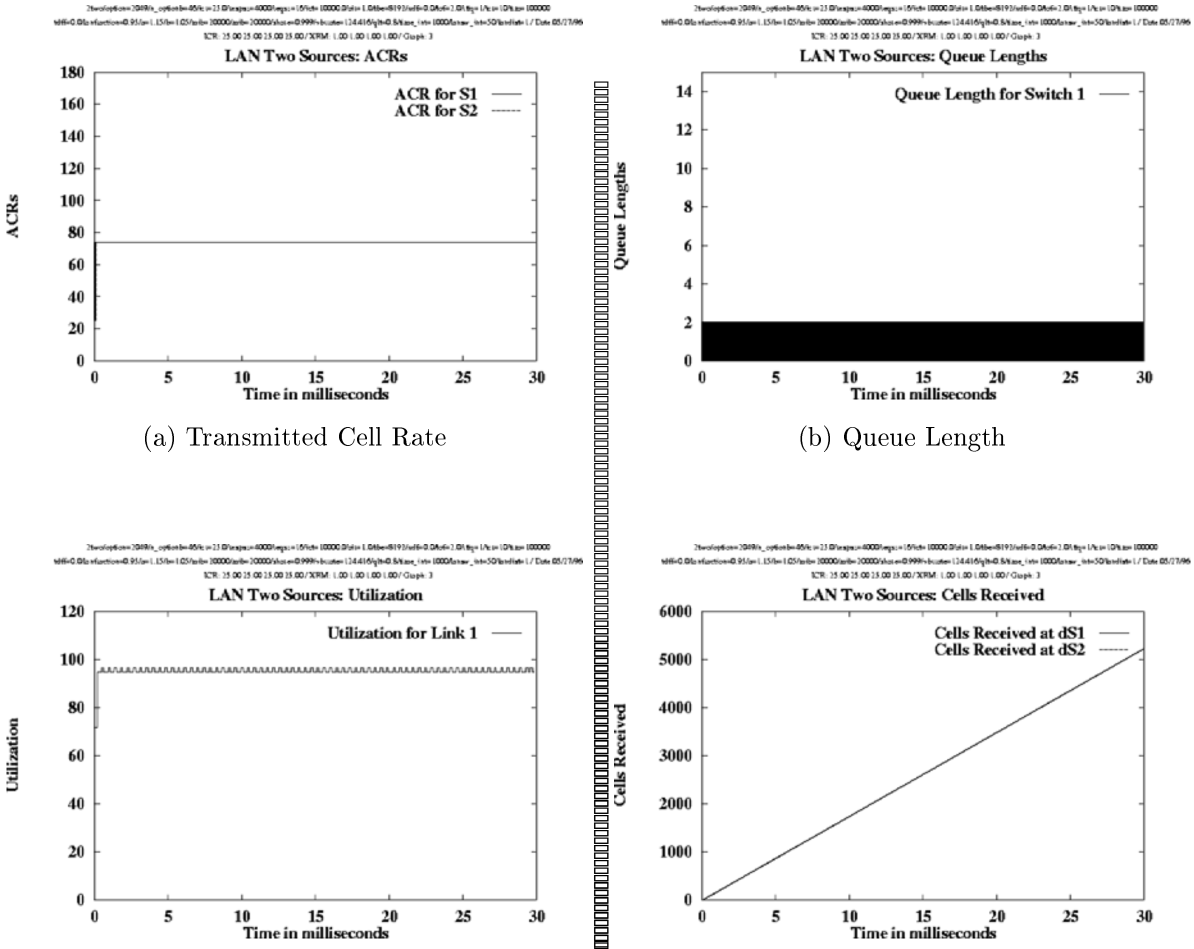
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.12: Results for a one source configuration in a LAN (ERICA+)



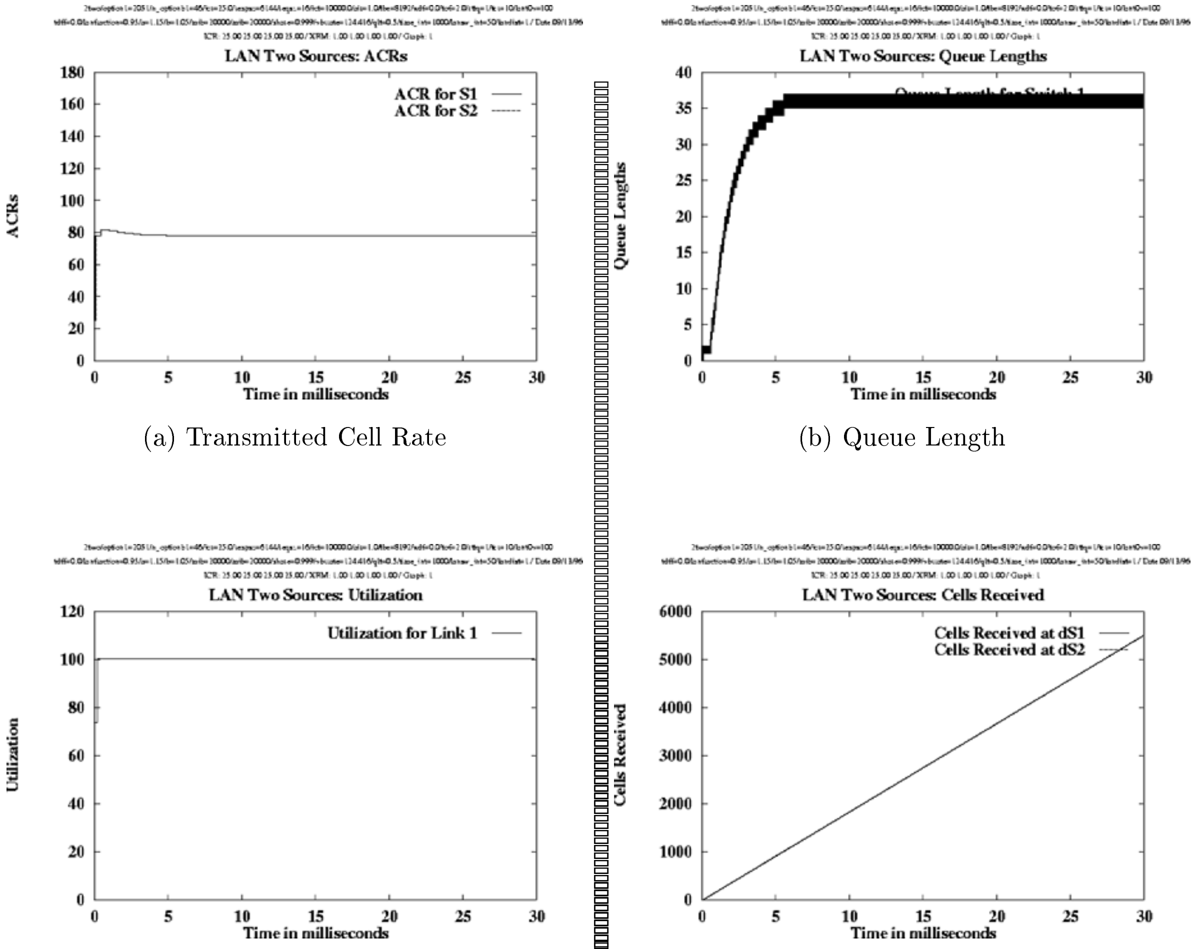
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.13: Results for a two sources configuration in a LAN (ERICA)



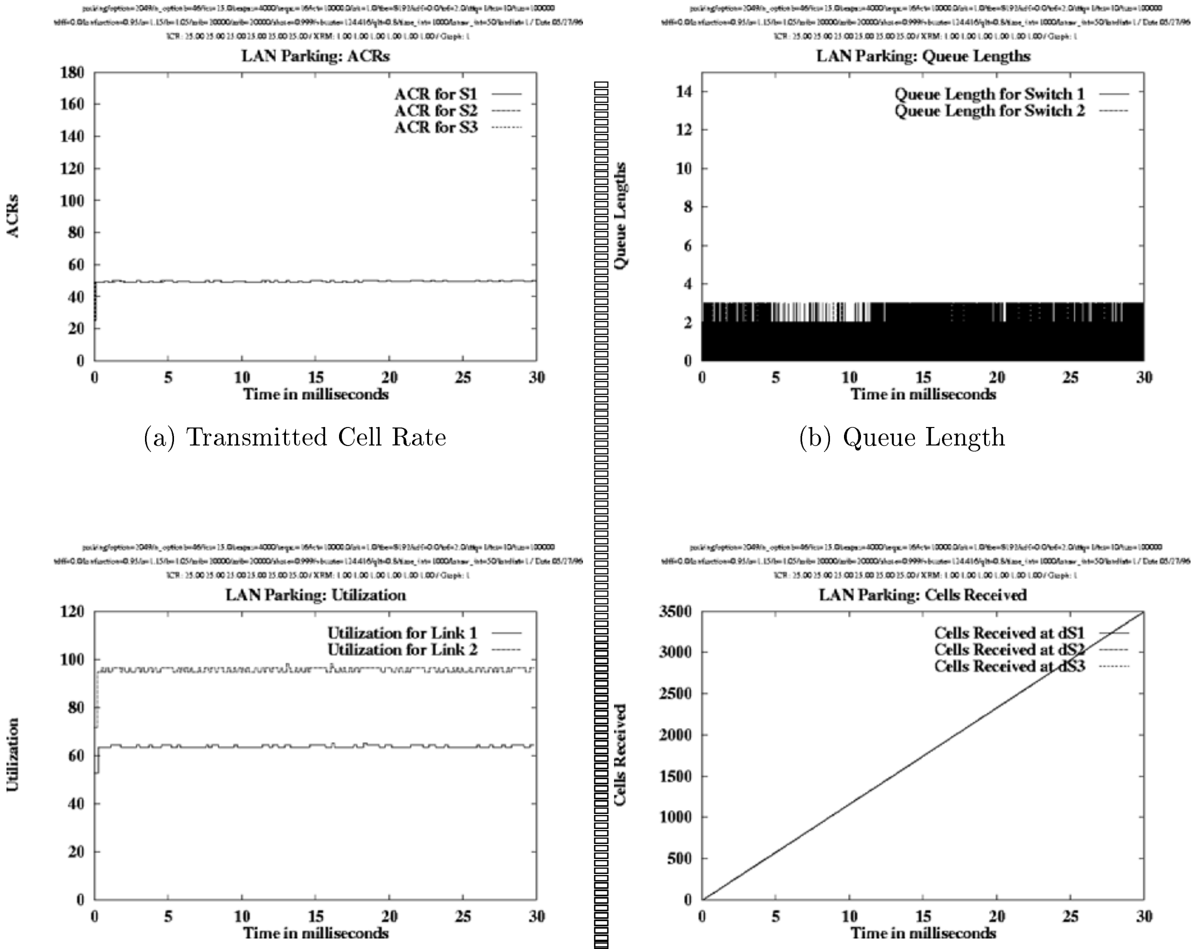
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.14: Results for a two sources configuration in a LAN (ERICA+)



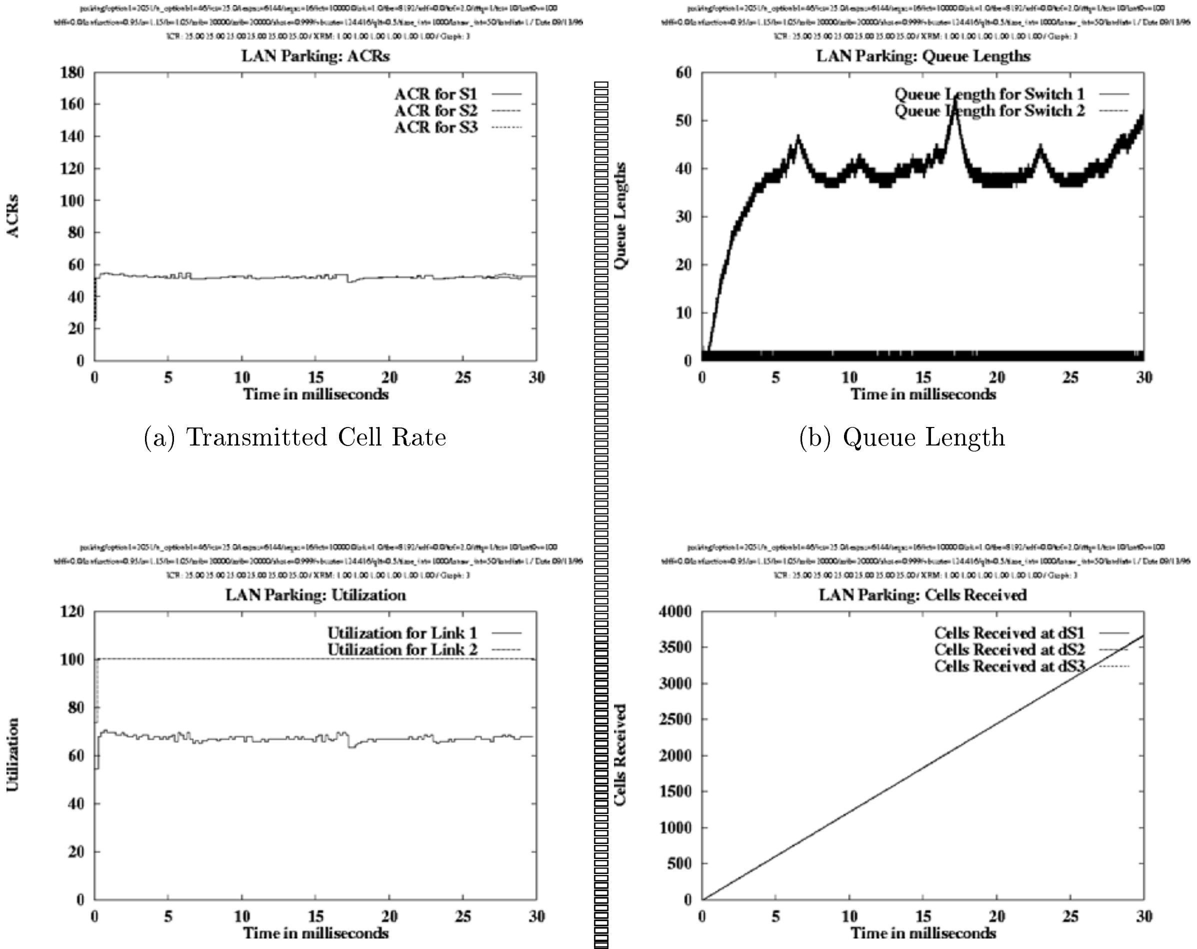
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.15: Results for a parking lot configuration in a LAN (ERICA)



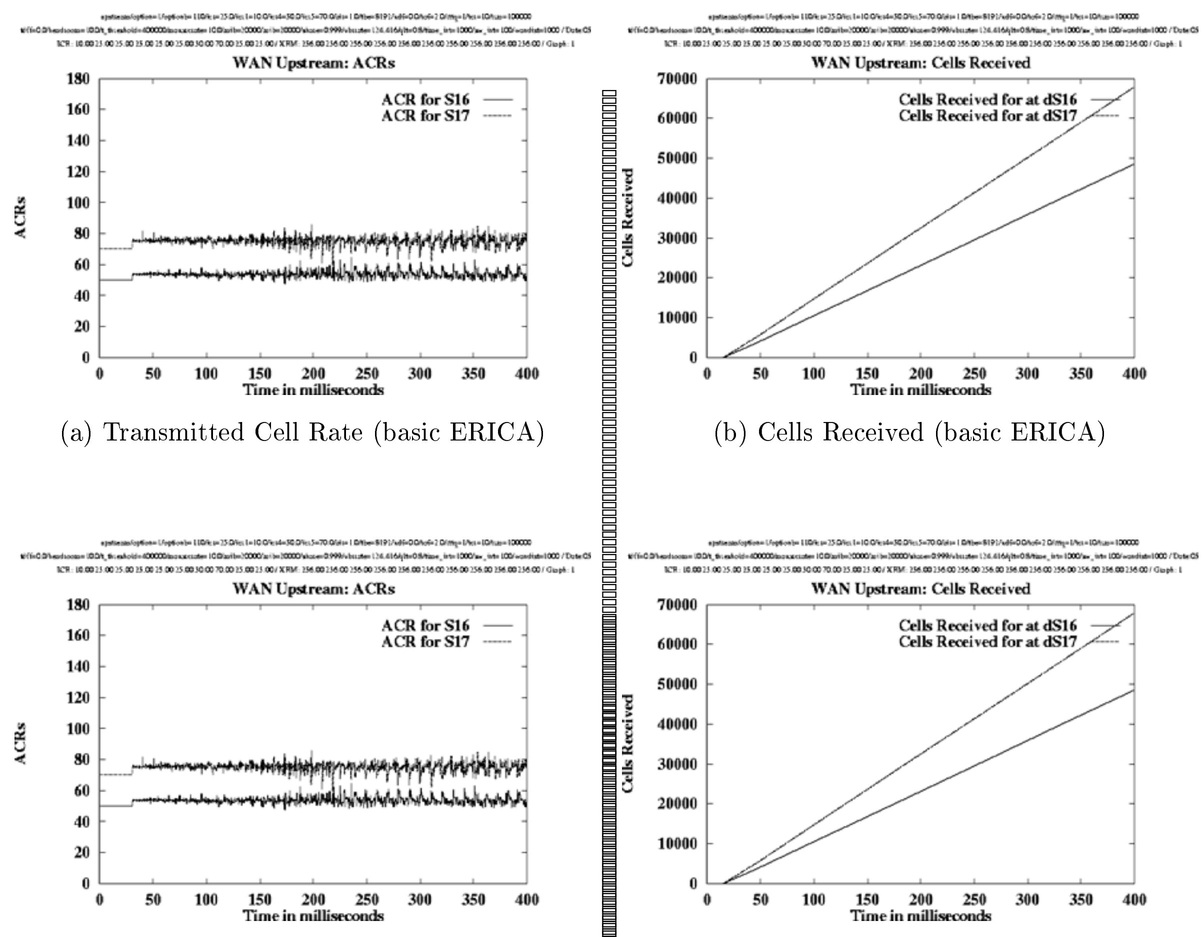
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.16: Results for a parking lot configuration in a LAN (ERICA+)



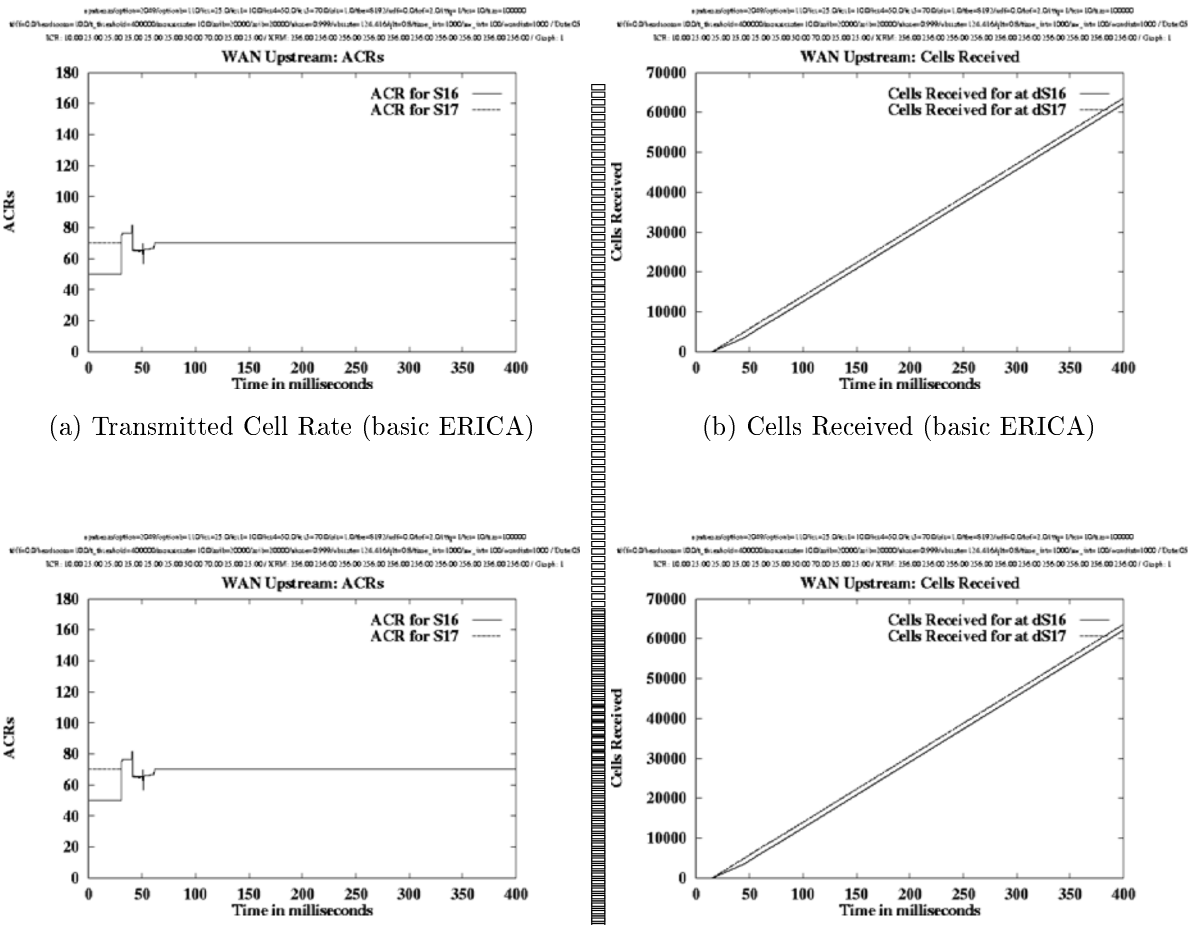
(a) Transmitted Cell Rate (basic ERICA)

(b) Cells Received (basic ERICA)

(c) Transmitted Cell Rate

(d) Cells Received

Figure 6.17: Results for an upstream configuration in a WAN (ERICA without the max-min fairness enhancement)



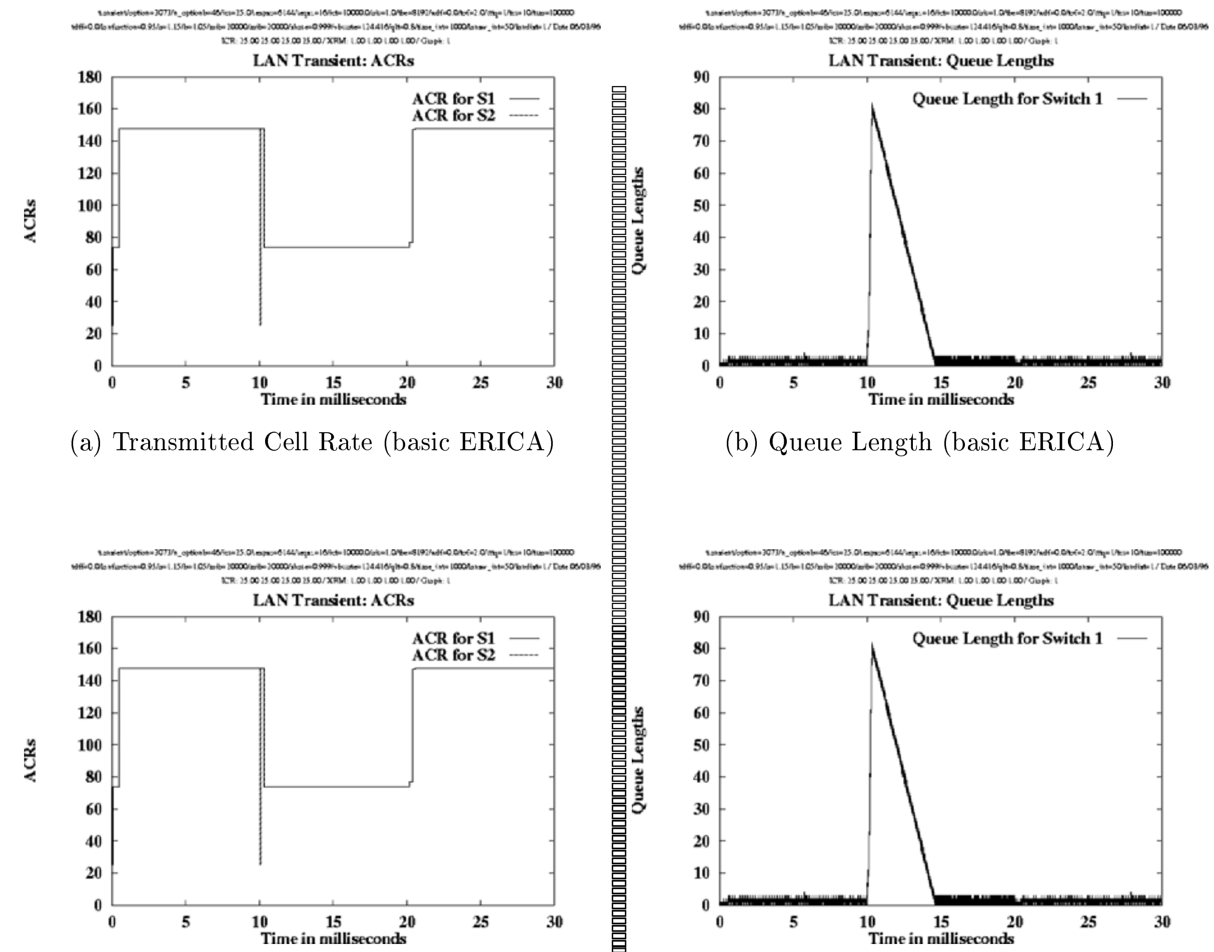
(a) Transmitted Cell Rate (basic ERICA)

(b) Cells Received (basic ERICA)

(c) Transmitted Cell Rate

(d) Cells Received

Figure 6.18: Results for an upstream configuration in a WAN (ERICA with the max-min fairness enhancement)



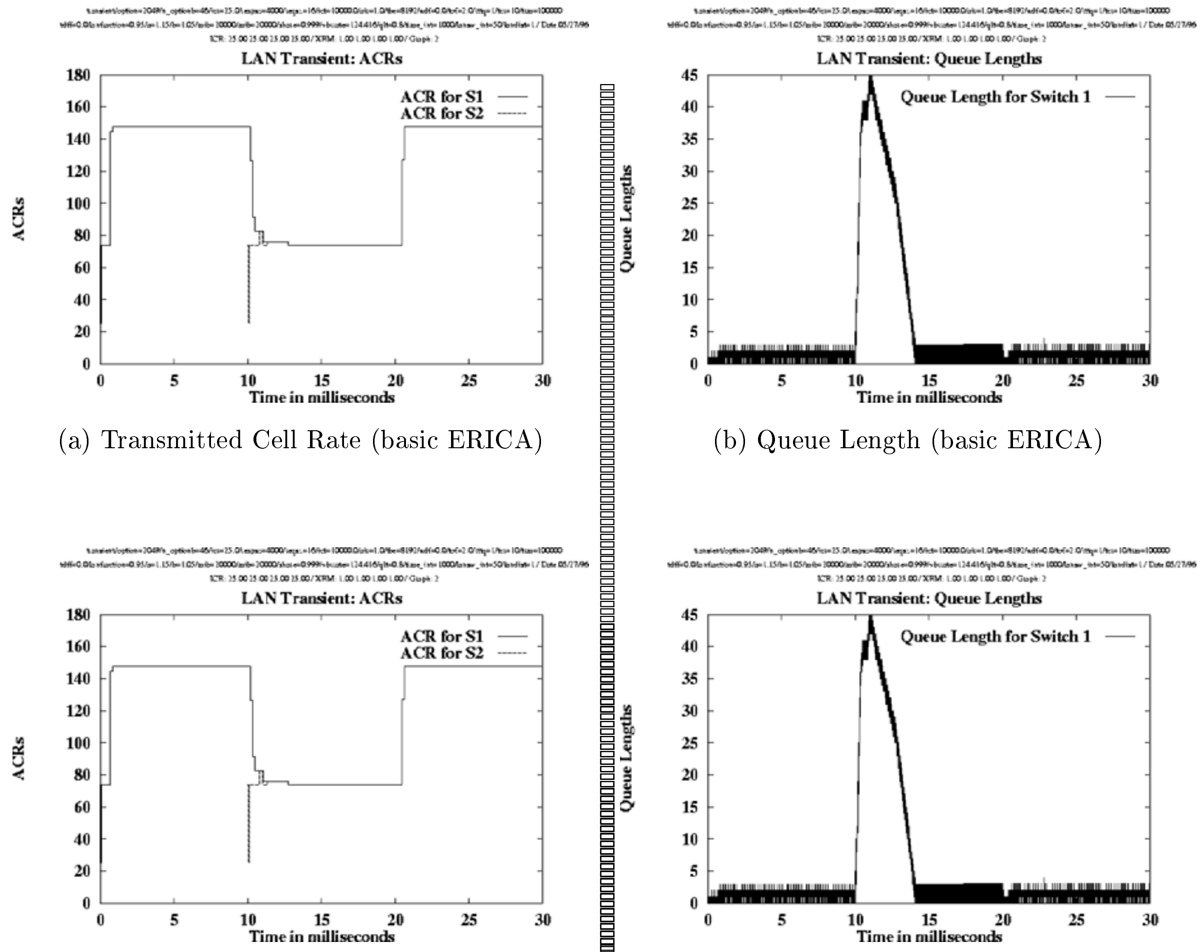
(a) Transmitted Cell Rate (basic ERICA)

(b) Queue Length (basic ERICA)

(c) Transmitted Cell Rate

(d) Queue Length

Figure 6.19: Results for a transient source configuration in a LAN (ERICA without the “fairshare first” enhancement)



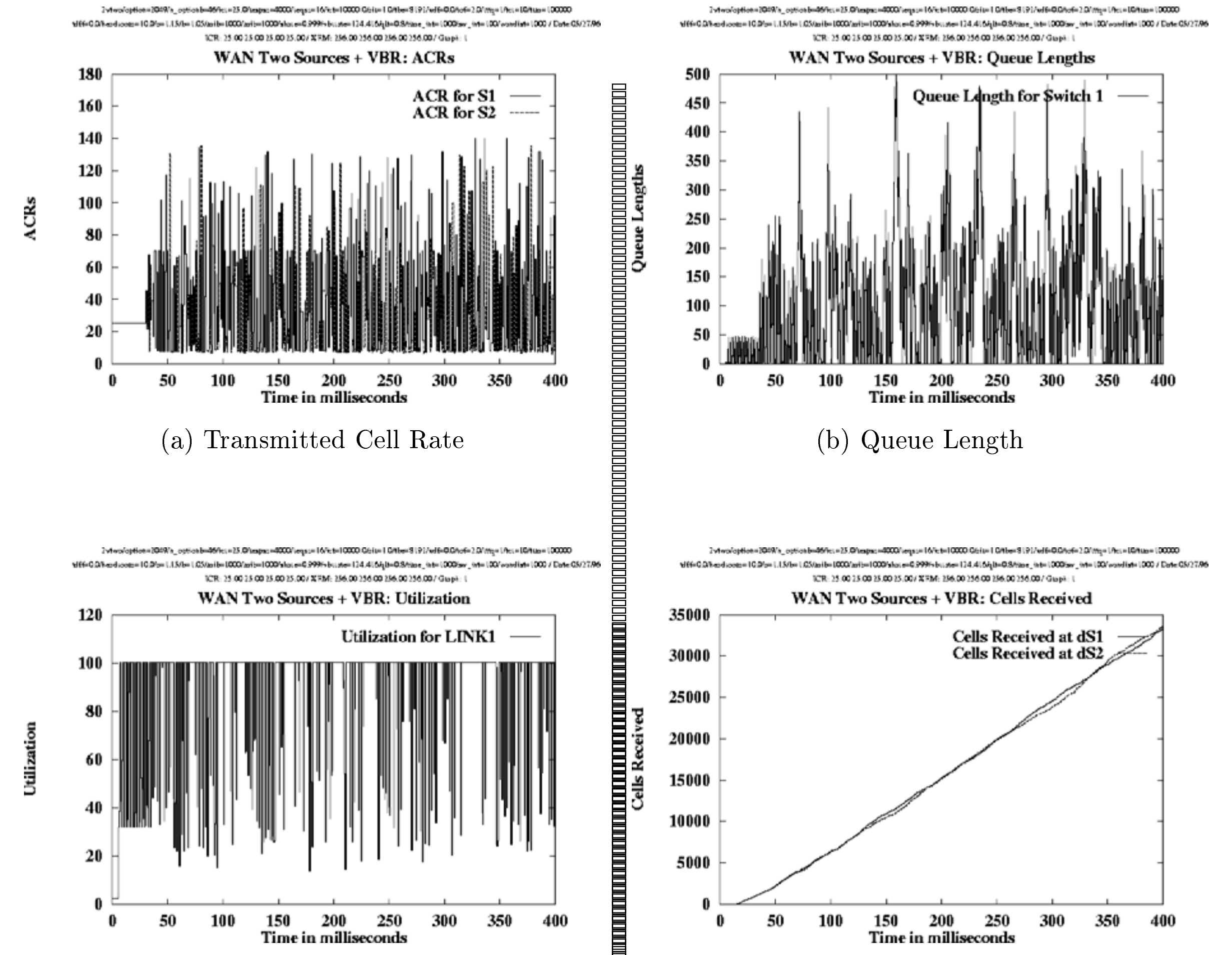
(a) Transmitted Cell Rate (basic ERICA)

(b) Queue Length (basic ERICA)

(c) Transmitted Cell Rate

(d) Queue Length

Figure 6.20: Results for a transient source configuration in a LAN (ERICA with the "fairshare first" enhancement)



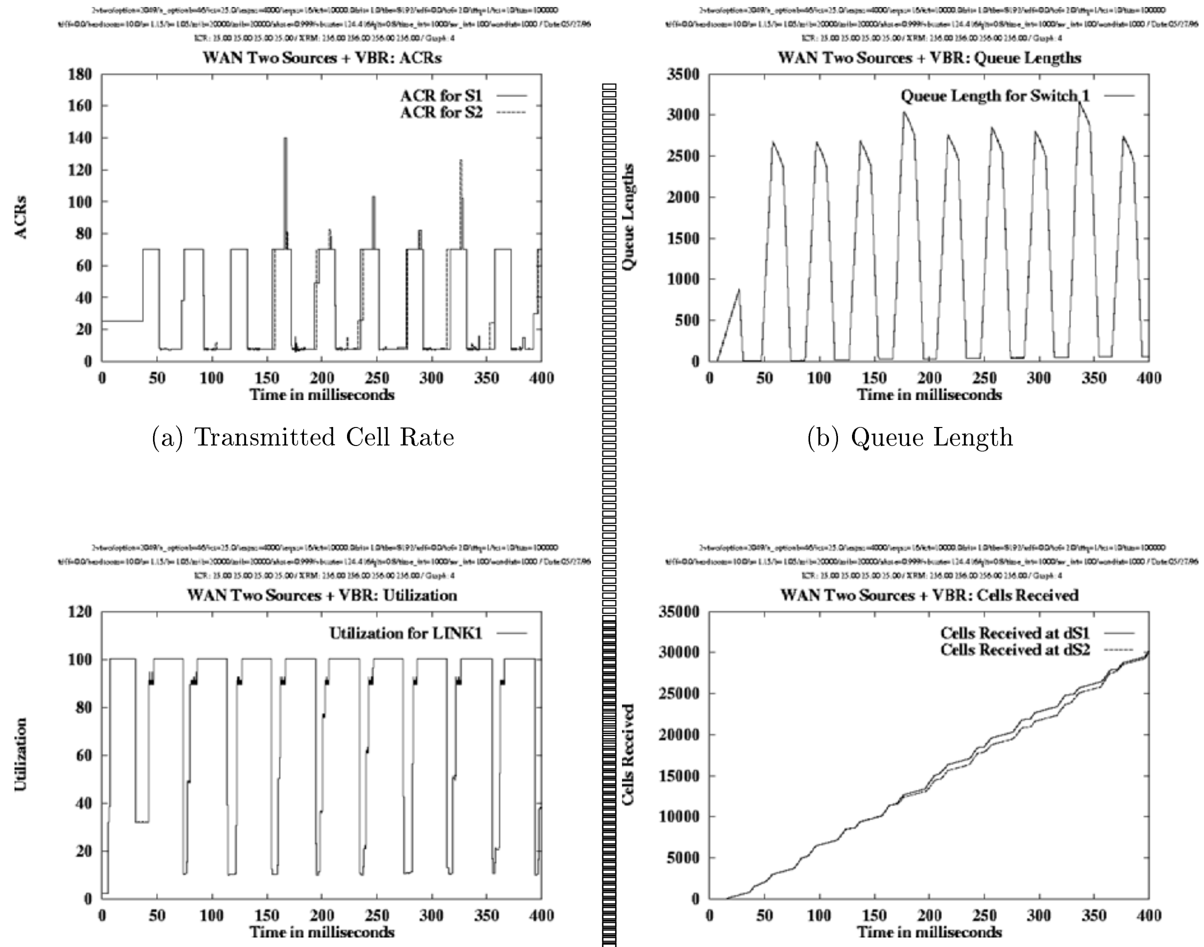
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.21: Results for a two source configuration with (1 ms on/1 ms off) VBR background in a WAN (ERICA)



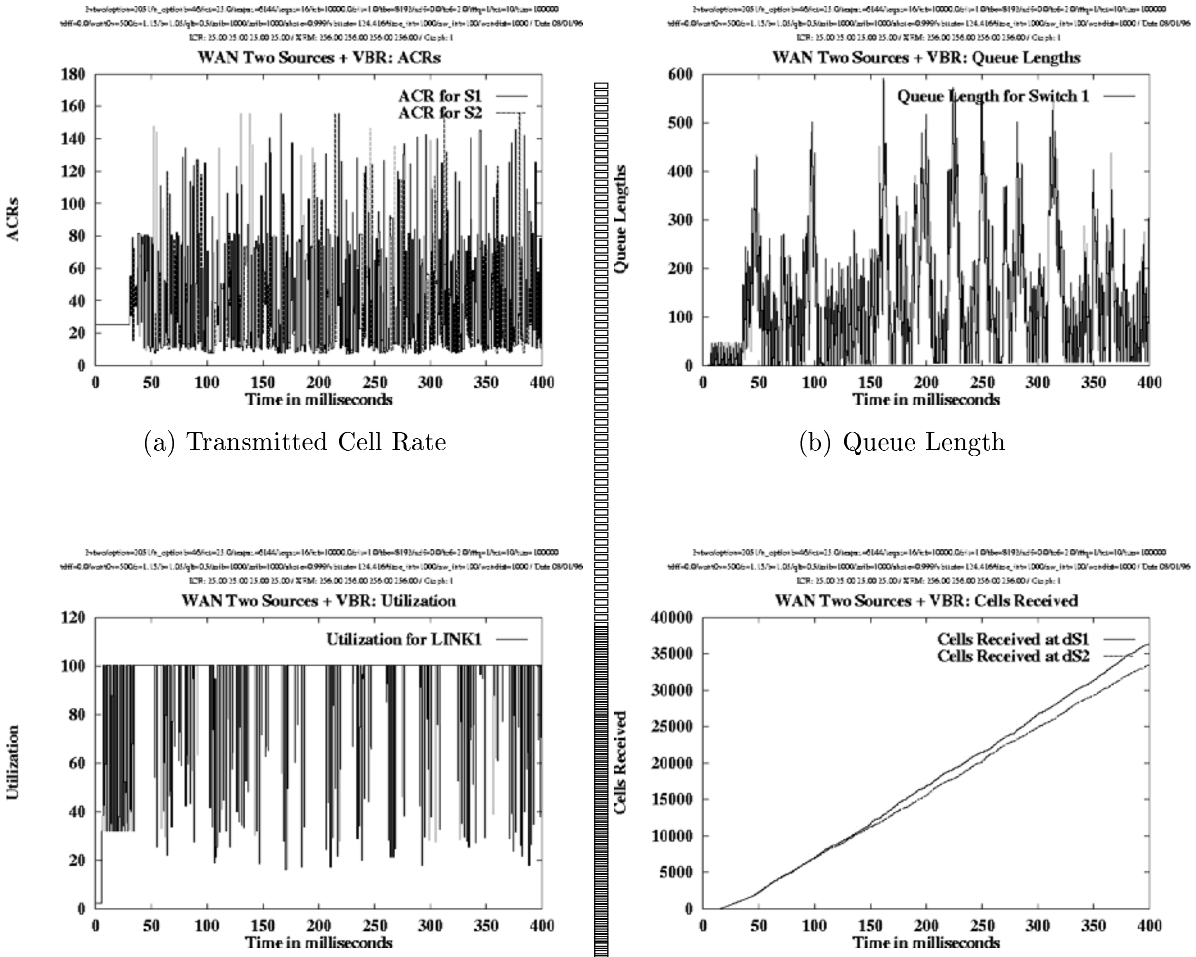
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.22: Results for a two source configuration with (20 ms on/20 ms off) VBR background in a WAN (ERICA)



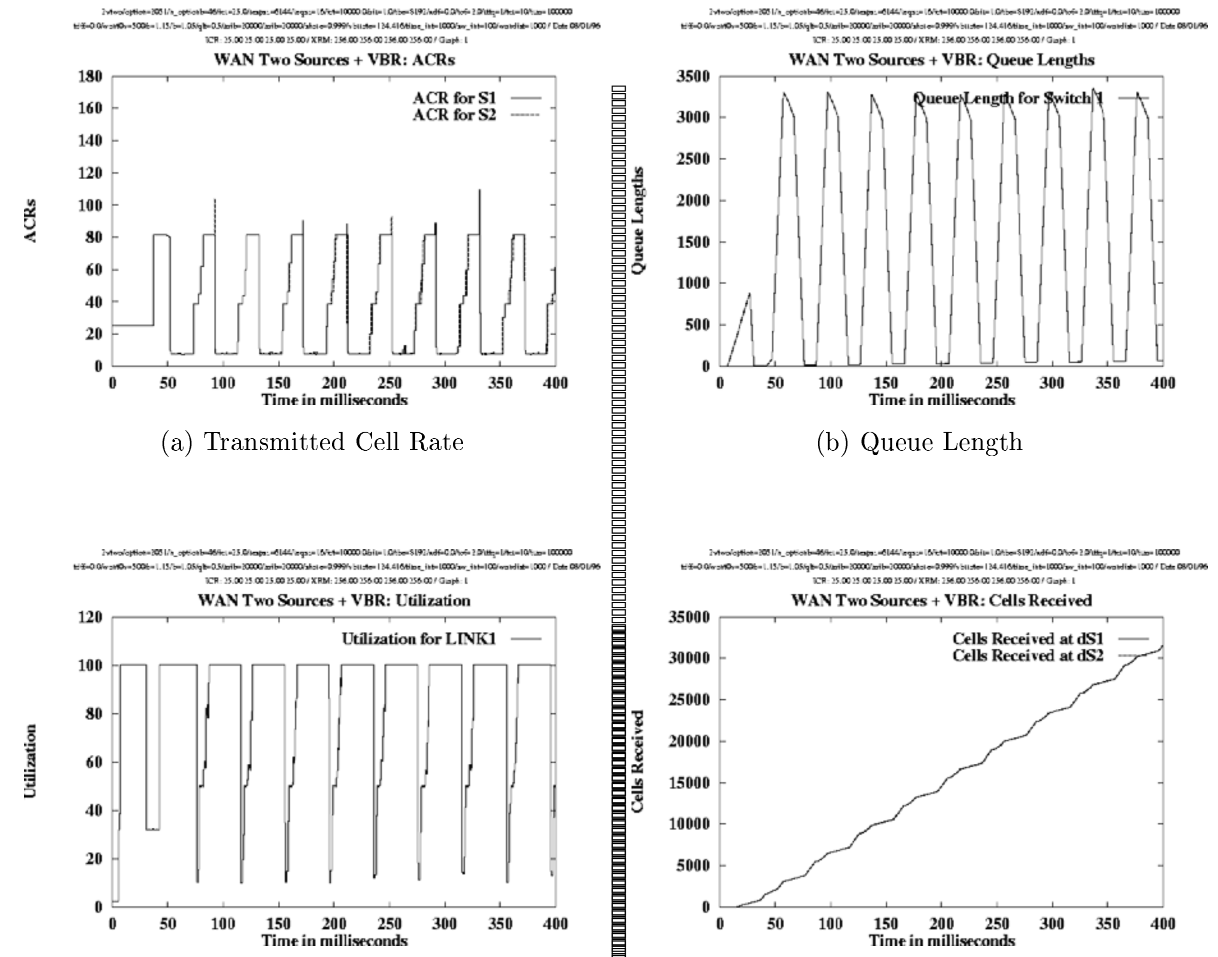
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.23: Results for a two source configuration with (1 ms on/1 ms off) VBR background in a WAN (ERICA+)



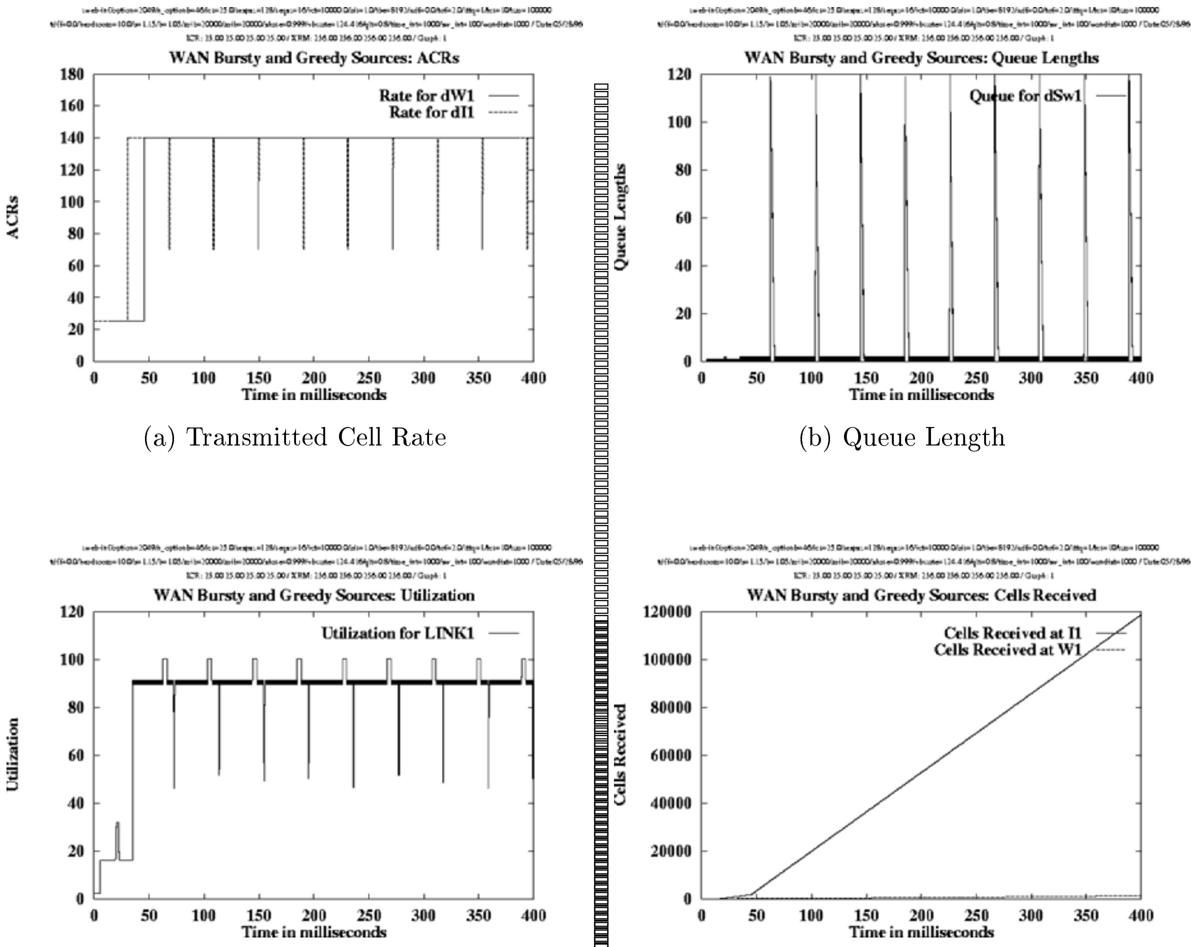
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.24: Results for a two source configuration with (20 ms on/20 ms off) VBR background in a WAN (ERICA+)



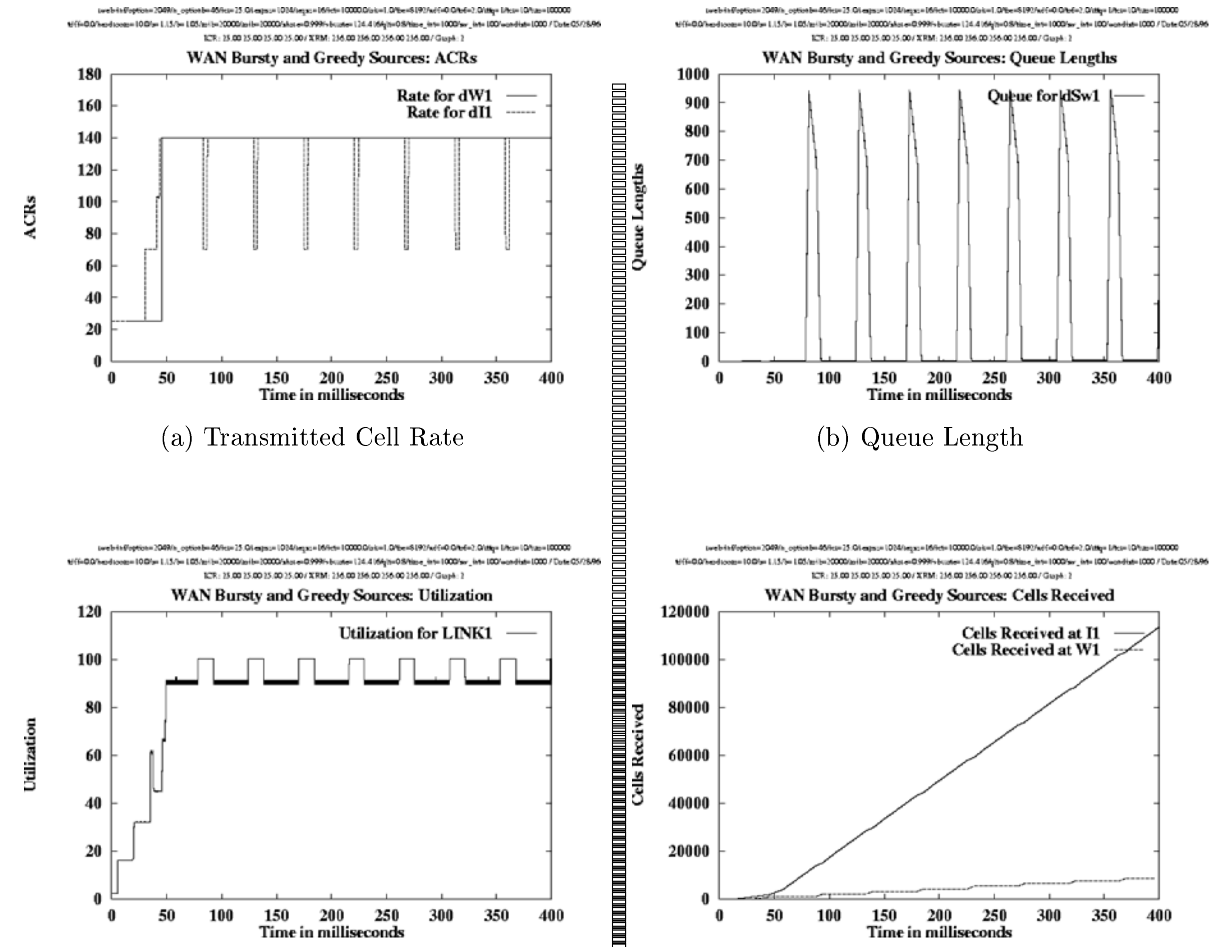
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.25: Results for one persistent source and one bursty source (small bursts) in a WAN (ERICA)



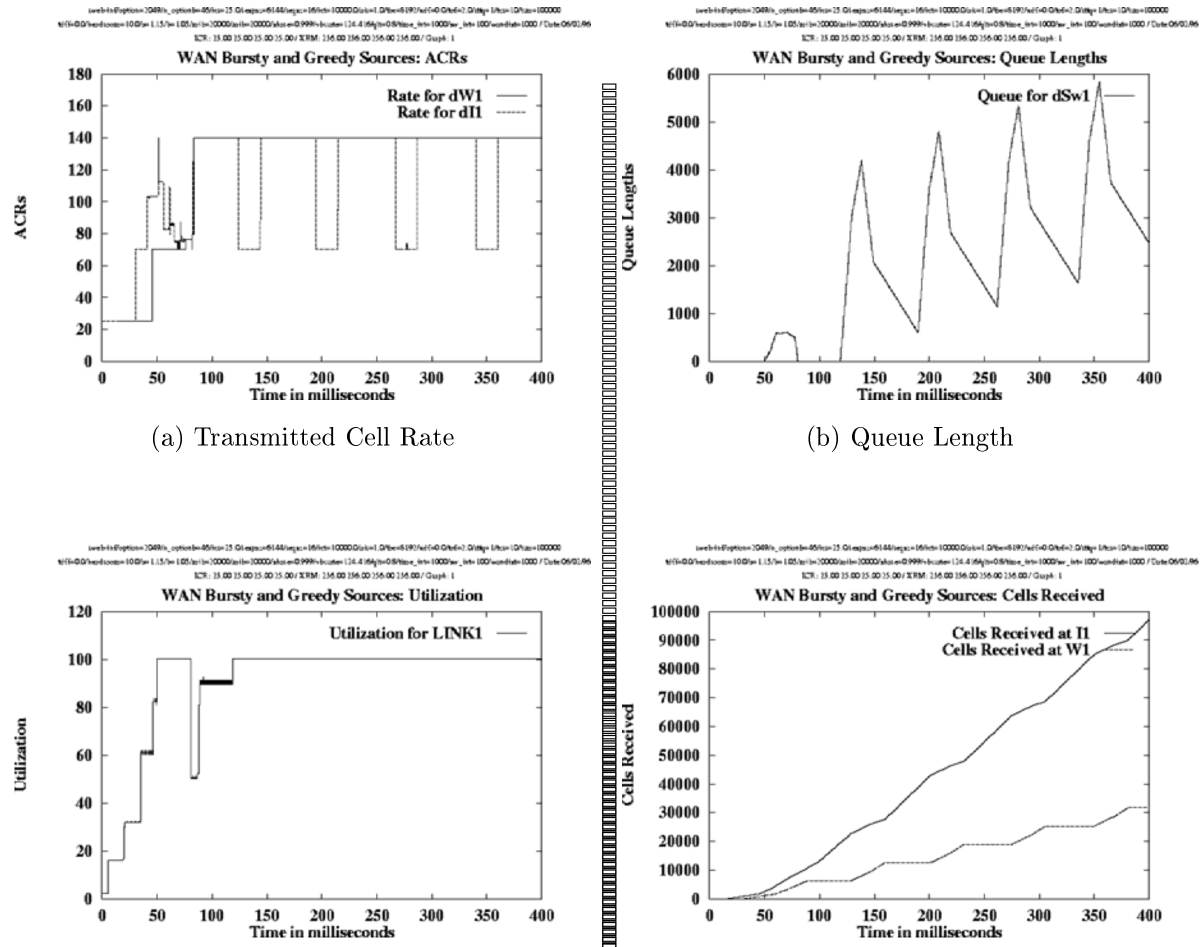
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.26: Results for one persistent source and one bursty source (medium bursts) in a WAN (ERICA)



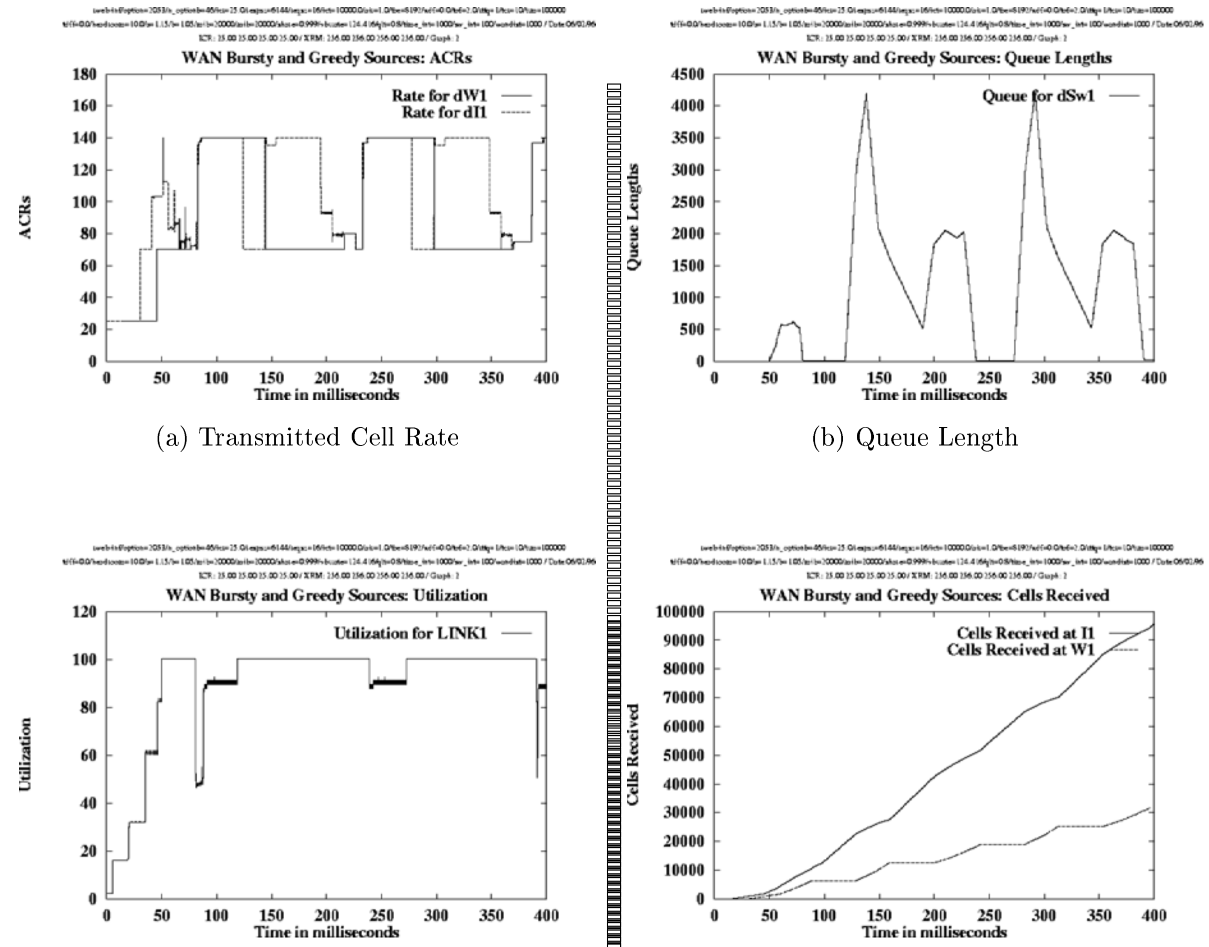
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.27: Results for one persistent source and one bursty source (large bursts) in a WAN (ERICA)



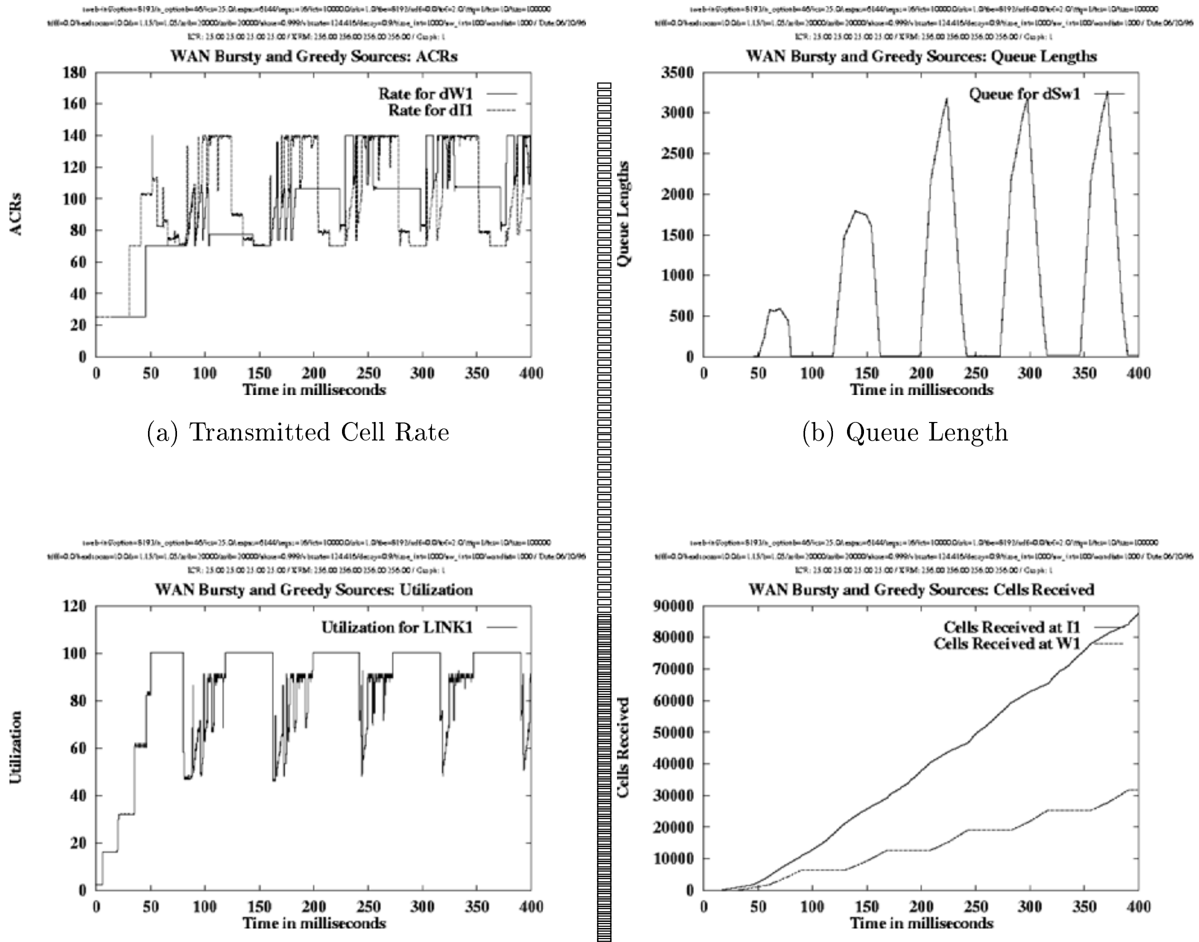
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.28: Results for one persistent source and one bursty source (large bursts) in a WAN (ERICA with bidirectional counting)



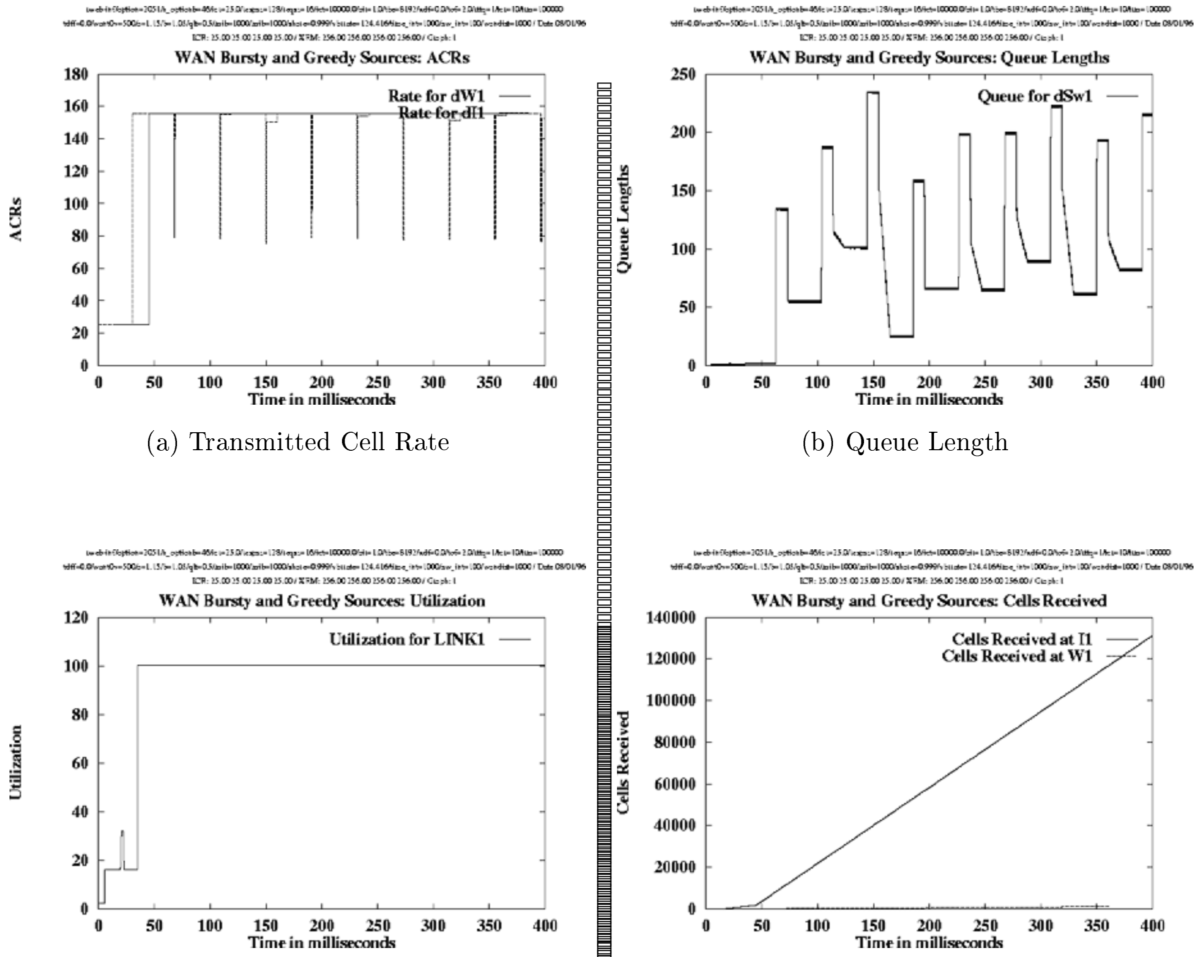
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.29: Results for one persistent source and one bursty source (large bursts) in a WAN (ERICA with averaging of number of sources)



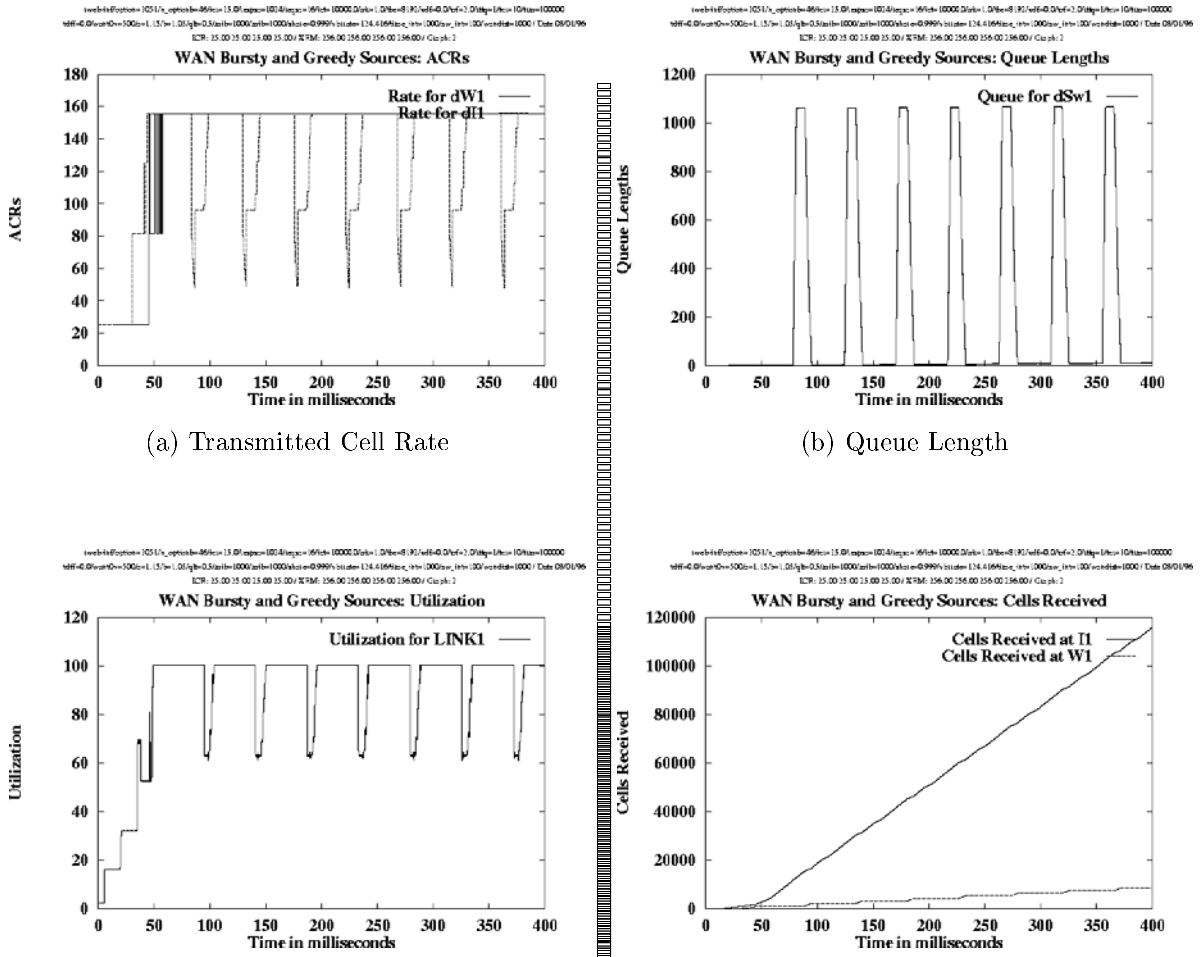
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.30: Results for one persistent source and one bursty source (small bursts) in a WAN (ERICA+)



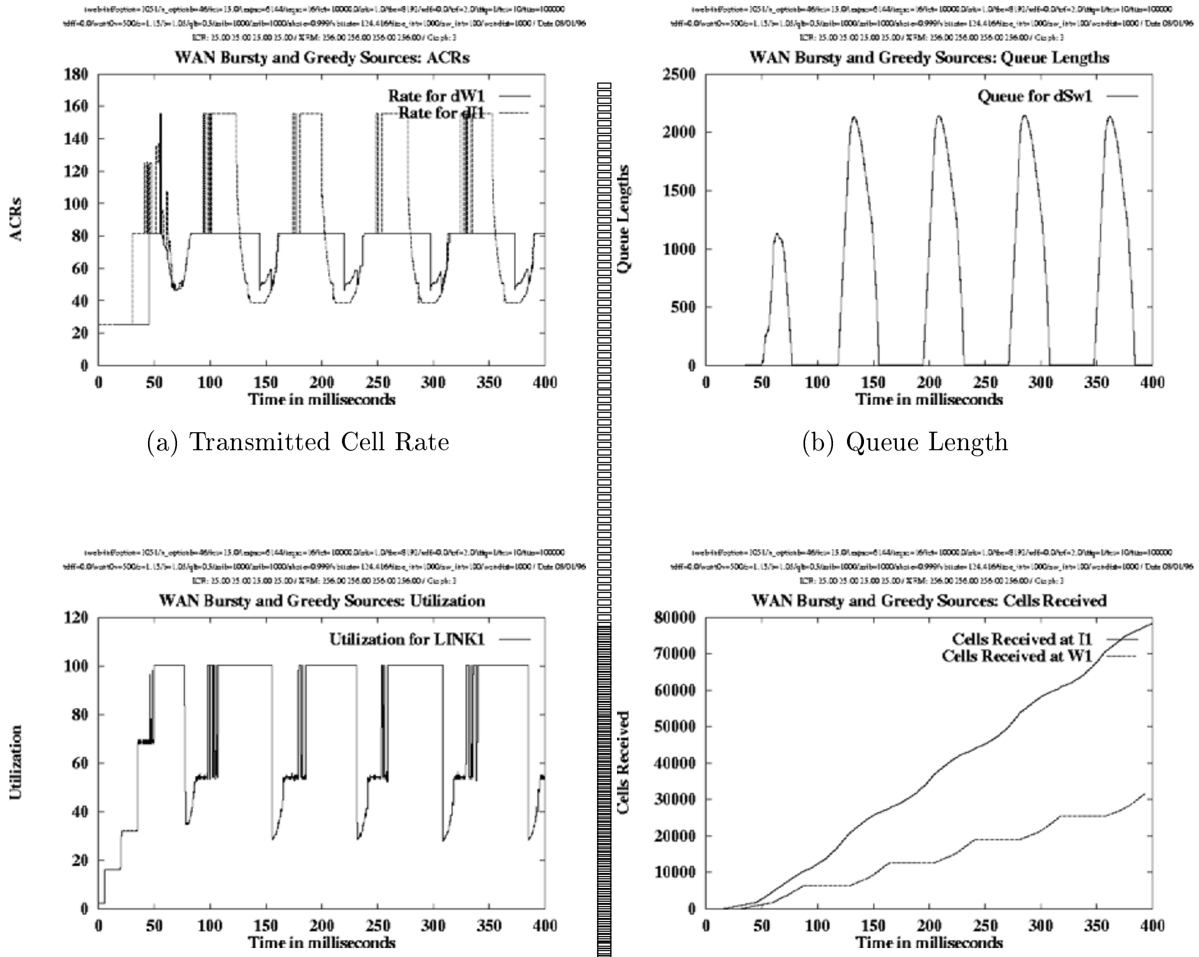
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.31: Results for one persistent source and one bursty source (medium bursts) in a WAN (ERICA+)



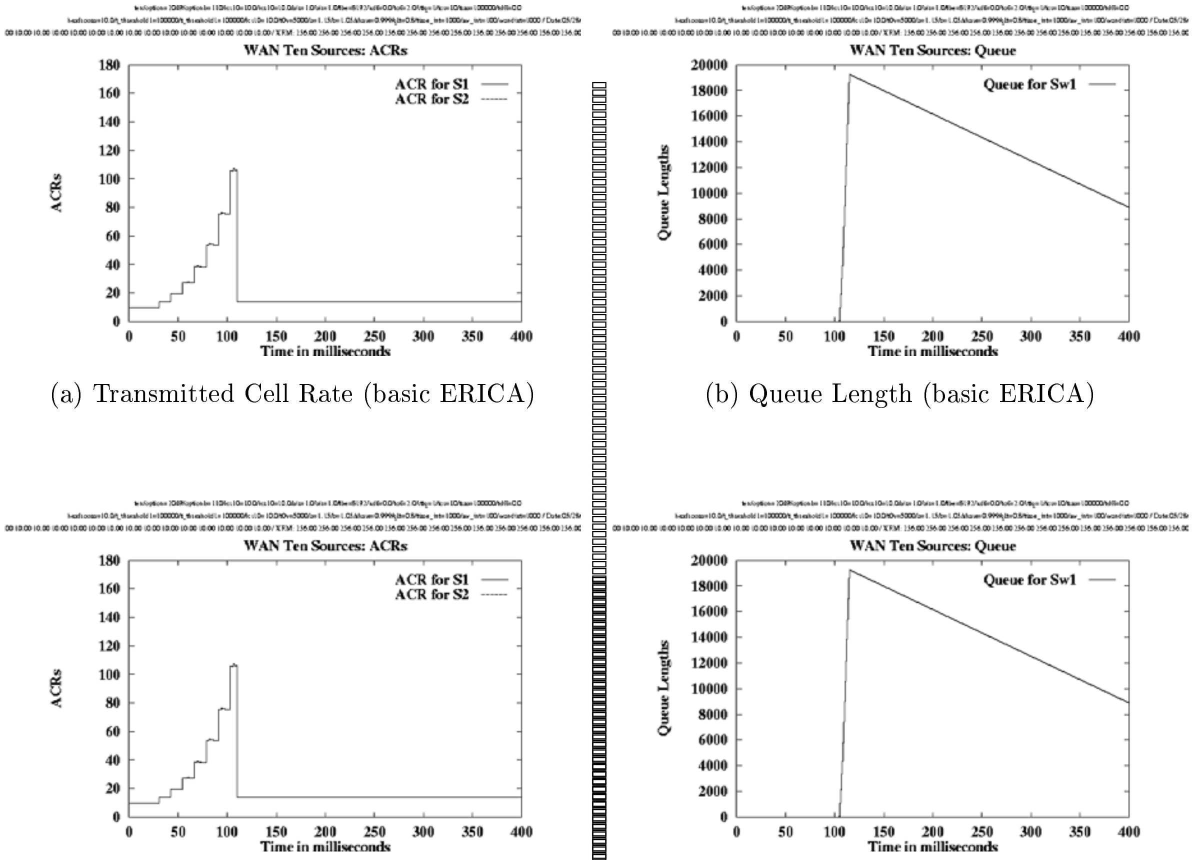
(a) Transmitted Cell Rate

(b) Queue Length

(c) Link Utilization

(d) Cells Received

Figure 6.32: Results for one persistent source and one bursty source (large bursts) in a WAN (ERICA+)



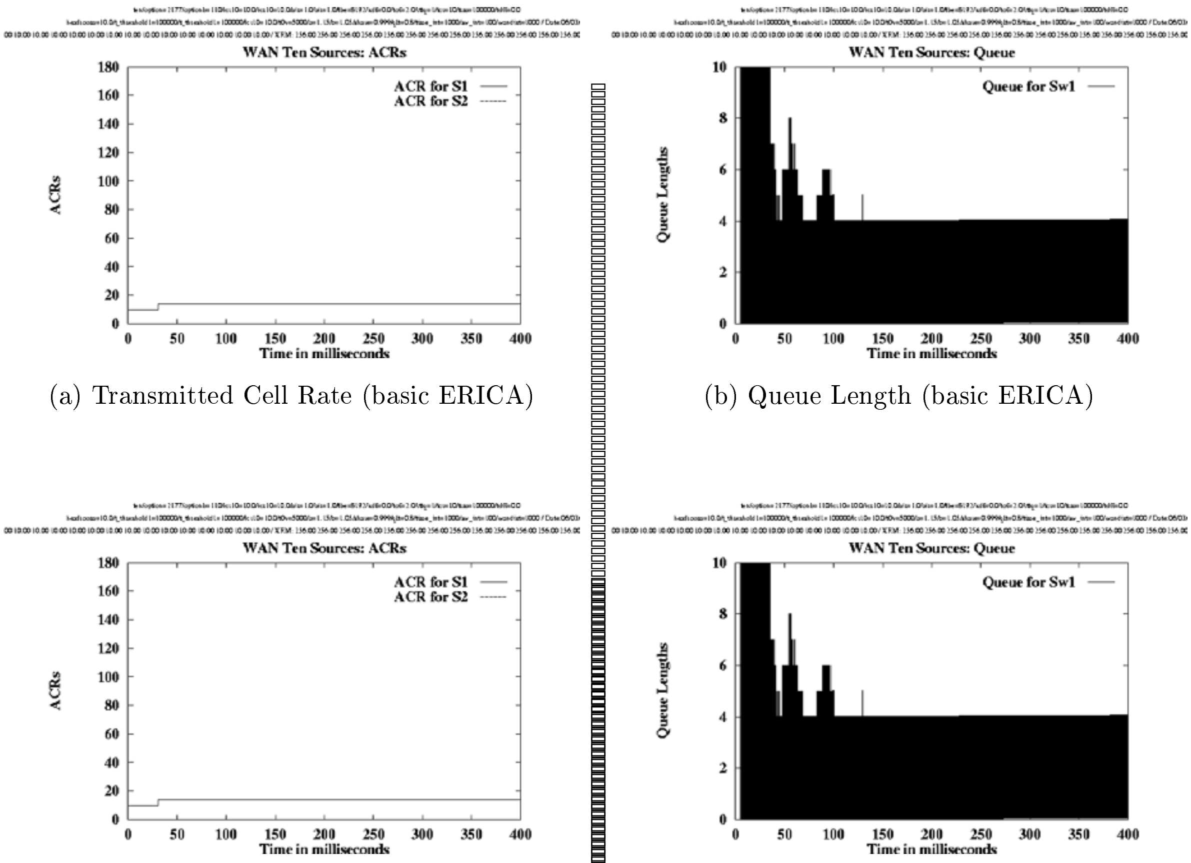
(a) Transmitted Cell Rate (basic ERICA)

(b) Queue Length (basic ERICA)

(c) Transmitted Cell Rate

(d) Queue Length

Figure 6.33: Results for ten ACR retaining sources in a WAN (ERICA)



(a) Transmitted Cell Rate (basic ERICA)

(b) Queue Length (basic ERICA)

(c) Transmitted Cell Rate

(d) Queue Length

Figure 6.34: Results for ten ACR retaining sources in a WAN (ERICA with per-VC CCR measurement)

BIBLIOGRAPHY

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

[2] Yehuda Afek, Yishay Mansour, and Zvi Ostfeld. Convergence Complexity of Optimistic Rate Based Flow Control Algorithms. In *28th Annual Symposium on Theory of Computing (STOC)*, pages 89–98, 1996.

[3] Yehuda Afek, Yishay Mansour, and Zvi Ostfeld. Phantom: A Simple and Effective Flow Control Scheme. In *Proceedings of the ACM SIGCOMM*, pages 169–182, August 1996.

[4] Anthony Alles. ATM Internetworking. White paper, Cisco Systems, http://www.cisco.com, May 1995.

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

[6] Ambalavanar Arulambalam, Xiaoqiang Chen, and Nirwan Ansari. Allocating Fair Rates for Available Bit Rate Service in ATM Networks. *IEEE Communications Magazine*, 34(11):92–100, November 1996.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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