

\*\*\*\*\*  
ATM Forum Document Number: ATM Forum/95-1346

\*\*\*\*\*  
Title: ERICA+: Extensions to the ERICA Switch Algorithm

\*\*\*\*\*  
Abstract:

Several enhancements to ERICA are described. The resulting scheme, ERICA+ provides congestion avoidance with 100% utilization. Even at 100% utilization, the queue lengths are bounded. The scheme gives better control of end-to-end delay compared to other schemes and keeps link utilization of expensive links high despite idle periods in the input load. The scheme responds well to VBR traffic by estimating the ABR capacity and employing a scheduling strategy between ABR and VBR. Simulation results on a set of representative configurations will be presented.

\*\*\*\*\*  
Source:

Raj Jain, Shiv Kalyanaraman, Rohit Goyal, Sonia Fahmy, and Fang Lu  
The Ohio State University

Raj Jain is now at Washington University in Saint Louis, [jain@cse.wustl.edu](mailto:jain@cse.wustl.edu) <http://www.cse.wustl.edu/~jain/>

The presentation of this contribution at ATM Forum is sponsored by NASA.

\*\*\*\*\*  
Date: October 1995

\*\*\*\*\*  
Distribution: ATM Forum Technical Working Group Members  
(Traffic Management)

\*\*\*\*\*  
Notice: This contribution has been prepared to assist the ATM Forum. It is offered to the Forum as a basis for discussion and is not a binding proposal on the part of any of the contributing organizations. The statements are subject to change in form and content after further study. Specifically, the contributors reserve the right to add to, amend or modify the statements contained herein.

\*\*\*\*\*  
INTRODUCTION

The ERICA+ scheme is an enhancement to the ERICA scheme [1,2] with changes in the following areas:

1. Estimation of available bandwidth when CBR and VBR take precedence over ABR.
2. Implementation of a scheduling policy for VBR and ABR cells.
3. Full (100%) utilization of link bandwidth with controlled queue lengths.
4. Quick reduction of queues during transient overloads.

5. Uses available bandwidth for ABR even if the bandwidth becomes available just for a short duration.
6. Scalable for wide range of link speeds and for varying available ABR capacities at these speeds.

BRIEF OVERVIEW OF ERICA:

Explicit Rate Indication for Congestion Avoidance (ERICA) is a simple switch scheme that allocates bandwidth fairly with a fast response. The scheme consists of using a target utilization of, say, 95%. The target rate is then set at:

$$\text{Target Rate} = \text{Target Utilization} * \text{Link Rate}$$

The overload is measured with respect to the target rate (and not link rate):

$$\text{Overload} = \text{Input rate} / \text{Target Rate}$$

In addition to the input rate, the switches also measure the number of active VCs and compute the fair share:

$$\text{Fair Share} = \text{Target Rate} / \text{Number of active VCs}$$

For each VC, its share is computed based on the overload factor and the VC's current cell rate:

$$\text{A VC's share} = \text{VC's current cell rate} / \text{Overload}$$

The VC is given the maximum of its share as computed above or the fair share.

$$\text{ER for VC} = \max(\text{Fair share}, \text{VC's share})$$

The explicit rate (ER) in the RM cell is reduced if ER for VC as computed above is less:

$$\text{ER in Cell} = \min(\text{ER in Cell}, \text{ER for the VC})$$

This simple algorithm has several desirable properties including fast response time, low queue length, and simplicity. Unlike many other switch algorithms, ERICA is not affected by "ACR Retention." A VC does not have to use all of its allocated cell rate. Source rule 5 (implemented or not) does not affect its performance.

EXTENSION 1: VBR TRAFFIC

If a switch provides both ABR and VBR service (and probably CBR as well), VBR takes precedence. The available bandwidth for ABR is not link rate but what ever is left over after VBR. This is handled by the following modification to the ERICA:

The switches measure VBR input rate and compute ABR capacity:

$$\text{ABR Capacity} = \text{Target rate} - \text{VBR input rate}$$

The fair share is measured based on the dynamic estimate of the ABR capacity:

$$\text{Fair share} = \text{ABR Capacity} / \text{Number of active ABR VCs}$$

The switches also measure the ABR input rate and compute overload factor:

$$\text{Overload factor} = \text{ABR input rate} / \text{ABR capacity}$$

This overload factor is used for computing each ABR VC's share. The rest of the computation is identical to that presented earlier for ERICA. That is,

$$\text{VC's share} = \text{VC's current cell rate} / \text{Overload factor}$$

$$\text{ER for VC} = \max(\text{Fair share}, \text{VC's share})$$

EXTENSION 2: SCHEDULING OF ABR AND VBR

Since the switches provide multiple classes of service, they maintain multiple queues. The key question is how cells in these different queues are serviced. In this section, we describe a scheduling policy which allows the implementor (or user) to allocate "soft" percentages of link capacity for various classes. These allocations are soft in the sense that if one class does not use its allocation, it is automatically passed on to the other class(es).

For example, in the case of a simple two class (VBR and ABR) system, an implementor could decide to give VBR a maximum of 90% and ABR a minimum of 10% bandwidth. If total VBR load is only 20%, ABR gets the remaining 80%. On the other hand if VBR input rate is 110% and ABR input rate is 15%, VBR gets only 90% and ABR gets 10%. If VBR and ABR are 110% and 5%, VBR gets 95% and ABR gets 5%.

Notice no class is starved and no bandwidth is wasted.

The idea can be easily extended to any number of classes. The pseudocode for a two-class system is given below. In the pseudocode the following variables are used:

afrac = Minimum Fraction desired for ABR  
vfrac = Maximum Fraction desired for VBR  
(afrac ABR cells are transmitted for every vfrac VBR cells)

acredit = Current credit for ABR traffic  
vcredit = Current credit for VBR traffic  
(In general, the traffic with higher credit is serviced next.)

aqueue = Number cells in the ABR queue  
vqueue = Number cells in the VBR queue

account = Number of ABR cells served  
vcount = Number of VBR cells served

int vqueue, aqueue, vcount=0, account=0;  
float vfrac, afrac=1-vfrac;  
float vcredit = vfrac, acredit = afrac;

```
while (aqueue > 0 || vqueue > 0) {
    if (vcredit >= acredit) {
        if (vqueue > 0) {
            schedule VBR cell
            vqueue--; vcount++;
            if (aqueue > 0) {
                vcredit--;
                vcredit+=vfrac;
                acredit+=afrac;
            }
        } else if (aqueue > 0) {
            schedule ABR cell
            aqueue--;
            account++;
        }
    } else {
        if (aqueue > 0) {
            schedule ABR cell
            aqueue--; account++;
            if (vqueue > 0) {
                acredit--;
                acredit+=afrac;
                vcredit+=vfrac;
            }
        }
    }
}
```

```

    } else if (vqueue > 0) {
        schedule VBR cell
        vqueue--;
        vcount++;
    }
}
}
}

```

#### EXTENSION 3: 100% LINK UTILIZATION

So far, the common belief is that one has to have a target utilization less 100% to achieve congestion avoidance. (We may be partially responsible for this belief with our work on the OSU scheme). However, we have developed a method to achieve congestion avoidance while keeping the link fully utilized. This is particularly important for long distance public carrier links which are rather expensive and being able to service 5% more cells (translate that to 5% more revenue) can be a significant issue.

In ERICA, when one selects the target utilization of 95%, the remaining 5% is used for draining the queue when the input rate is more than the capacity. Thus, the main difference between a system with 95% target utilization and another with 85% target utilization is that in the latter case, the queues (if they ever build up) due to overload are drained out quickly. Lower target utilization leads to faster response to overloads.

The main problem with lower target utilization is that under steady state (when there is no overload), a larger fraction of the capacity is left unused. Thus, there is a tradeoff between steady state and transient operation.

One way to get the best of both worlds is to vary the target rate dynamically. During steady state target 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 =fn(queue length, link rate, VBR rate)

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

#### EXTENSION 4: QUICK DRAIN OF QUEUES DURING OVERLOAD

This is actually a side effect of the above design with dynamic target rate. For example, if the queues are large, the target rate may be 80% of the ABR capacity.

The switch will attempt to bring the ABR input rate to 80% of the ABR capacity. The remaining 20% is used to drain the current queue. If the queues become larger, the target rate will be set lower, to say, 70%. The switch will then used 30% to drain the queues. Notice that with a fixed target utilization of, say, 95%, only 5% of the capacity is used for draining the current queue regardless of the queue size.

#### EXTENSION 5: ABR CAPACITY IS USED EVEN IF IT BECOMES AVAILABLE FOR SHORT DURATION.

This is achieved by setting a "target queue length."

If the queue length falls below this value, the sources are encouraged to increase their rate and vice versa.

In ERICA, the steady state queue length is close to 1. While this gives minimum delay, we found that in a link shared by ABR and VBR, a queue length of 1 is too small. Whenever, the VBR goes away, it takes some time to get ABR cells from the source. In these cases, it is better to keep a bit larger ABR queue in the switch. These queued cells use the available capacity while the RM cells are taking the "good news" to the source and asking them to increase their rate.

We achieve this effect by using the "target queue length" as one of the parameters of "fn" used for the target rate. In other words:  
Target rate =fn(queue length, target queue length, Link rate, VBR rate)

#### EXTENSION 6: SCALABILITY TO VARIOUS SPEEDS

We achieve scalability by measuring all queue lengths in units of time rather than cells. This is more scalable since a network consists of links of various speeds. A queue length of 5 at a T1 link can be considered undesirable while a queue length of 50 at an OC-3 link may be considered OK. Thus,  
Target Rate =fn(queue delay, target queue delay, Link Rate, VBR Rate)

The above point is specially significant due to varying ABR capacity. This is especially true in the presence of VBR sources. However, the queue is only directly measurable quantity at the switch. The queuing delay is then estimated using the measured ABR capacity values.

We have done extensive simulations of these extensions. These simulation results will be presented at the Forum.

Note: Patent applications on ERICA and ERICA+ are pending [3]. The inventions are available for licensing on a non-exclusive and nondiscriminatory basis and at reasonable licensing fees.

#### REFERENCES:

- [1] AF-TM 95-0178R1: A Sample Switch Algorithm, February 1995.
- [2] AF-TM 95-0467: Simulation Results for ERICA Switch Algorithm with VBR+ABR traffic, April 1995.
- [3] AF-Plen 95-0978, Patent Disclosures, August 1995.

