THE VIRTUAL BANDWIDTH BASED ER MARKING ALGORITHMS FOR FLOW CONTROL IN ATM NETWORKS

Tao Yang, Ping Wang, and Wengang Zhai
Ascend Communications, Inc., One Robbins Road, Westford, MA 01886
{tyang,pwang,zhai}@ascend.com

Abstract ABR service is designed for a wide range of applications that do not require bounded delay and loss ratio, but rather prefer low loss ratio and high throughput with only a minimum cell rate guaranteed. In this paper, we introduce the concept of *virtual bandwidth* and discuss its application to ABR flow control. We develop a new explicit rate marking algorithm that adopts a *traffic-driven* measurement-based approach to track the available bandwidth and the virtual bandwidth concept to achieve both fairness and high utilisation. This algorithm exhibits O(1)-computational complexity, O(1)-storage complexity, fast responsiveness, and quick convergence. It does not require special information from end systems nor *any information* from other switches, such as bottleneck link and bottleneck connection indication. Simulation results show that the proposed scheme adapts well to the distributed dynamic network environment and converges to max-min fairness allocation quick.

Keywords: ATM Networks, Available Bit Rate Service, Flow Control, ER Marking, Virtual Bandwidth, Traffic-Driven.

1. INTRODUCTION

ATM networks offer the ABR service as one of its five service classes. The ABR service does not require bounded delay and loss ratio for a given connection, and guarantees only a minimum cell rate (MCR) specified by the user. To support ABR service, each switch in the network adopts a closed-loop control mechanism to control the source rate of each connection according to availability of bandwidth. The objective is to achieve high bandwidth utilization, fairness, and low cell loss ratio.

In a typical rate-based ABR flow control model, sources adapt their rates to network conditions. Information about the network condition is conveyed to the sources through *resource management* (RM) cells, periodically generated by sources and turned around by the corresponding destinations.

D. H. K. Tsang et al. (eds.), *Broadband Communications*

RM cells travelling from a source to destination are called forward RM (FRM) cells and those travelling the opposite direction are called backward RM (BRM) cells. Each RM cell contains several fields including the source minimum cell rate (MCR), current cell rate (CCR), and the explicit rate (ER). When a RM cell traverses along its path, its fields can be updated by a switch to reflect its congestion state. When the RM cell returns to the source, the values of these fields are used by the source to adjust its rate.

The rule by which a switch updates the content of a RM cell and/or a control bit in the header of a data cell is called the *switch behavior*. The ATM Forum requires that each switch implement at least one of three algorithms: explicit forward congestion indicator (EFCI) marking, relative rate (RR) marking, and explicit rate (ER) marking. For details of these marking algorithms, the reader is referred to [20]. Amongst the three, ER marking is the most effective and is also the most challenging and complicated one. In this paper, we will study only ER marking.

There are several ER marking algorithms proposed in the literature (see [1, 4, 18] for related surveys). These algorithms can be classified into two categories: *direct marking* and *progressive marking*. Examples of direct marking algorithms include [2, 5, 6, 11, 12, 15, 24, 25, 26]. This approach allows the link to achieve both fairness and high utilization. A typical implementation requires a certain form of per-connection accounting and needs extra processing time to perform necessary bookkeeping. At high speed, these requirements can be very expensive. Examples of progressive marking algorithms include [3, 7, 8, 9, 10, 13, 14, 16, 17, 19, 21, 22, 23, 27]. The advantage of progressive ER marking algorithms is simplicity and lower implementation complexity. But it is generally considered as being slow to converge to max-min fairness allocation.

Therefore, there is a compelling demand to develop a new ER scheme with the following objectives:

- *O(1)-Processing Complexity*. This implies that the processing time required to do rate allocation should be independent of the number of ABR connections traversing the link.
- *O(1)-Storage Complexity*. The O(1)-storage complexity rules out the costly per-connection accounting, found in most direct marking schemes [1,12,15,26] and some progressive marking algorithms [16].
- *Fast Responsiveness*. The responsiveness of the new algorithm should be comparable to that of direct marking algorithms.
- *Max-min Fairness*. In steady state, the new algorithm should converge to the max-min fair allocation of the chosen fairness criterion.
- *High Bandwidth Utilization*. Capable of achieving high utilization when not all connections are bottlenecked elsewhere.
- *Robustness*. Congestion indicators based on only '*Local Bound*' information, which is insensitive to either end system behavior, other switch behavior, or the network topology. This is essential for the

switch being scalable and successfully performing in the future heterogeneous networking environment.

In this paper, we introduce the concept of virtual bandwidth and discuss how it can be applied to develop a simple progressive ER marking algorithm, called the Traffic-driven, Virtual-bandwidth based ER Marking Algorithm (T-VERMA), that aims to achieve the above objectives. In the next section, we introduce the concept of virtual bandwidth and an approach to achieve max-min fairness. In section 3, we describe how to apply the concept to do distributed rate allocation and present details of T-VERMA. In section 4, we present simulation results and compare T-VERMA with two algorithms. Finally, we conclude our work in section 5.

2. VIRTUAL BANDWIDTH

2.1 Network Model and Notations

We consider a network consisting of a set of links, $\mathbf{J} = \{1, 2, \ldots, M\}$, with C_j being the capacity (bandwidth) of link $j \in \mathbf{J}$. Let $\mathbf{I} = \{1, 2, \ldots, N\}$ be a set of connections competing for network resources (bandwidth) and MCR_i be the minimum cell rate to be guaranteed by the network for connection $i \in \mathbf{I}$. Define $\mathbf{J}(i)$ to be the set of *links* traversed by connection $i \in \mathbf{I}$ and $\mathbf{I}(j)$ to be the set of *connections* traversing link $j \in \mathbf{J}$. To guarantee MCR for each connection, we assume that

$$\sum_{i \in \mathbf{I}(j)} MCR_i \le C_j \tag{1}$$

holds for all $j \in \mathbf{J}$.

For $i \in \mathbf{I}(j)$, we denote by $R_{i,j}$ the *rate allocated* to connection i by link j, also called the *local rate allocation* for connection i at link j. The matrix $[R_{i,j}]$ is called the *allocation matrix*. $[R_{i,j}]$ is *feasible* if it satisfies:

$$R_{i,j} \ge MCR_i, \quad i \in \mathbf{I} \text{ and } j \in \mathbf{J}(i); \text{ and } \sum_{i \in \mathbf{I}(j)} R_{i,j} \le C_j, \quad j \in \mathbf{J}. \tag{2}$$

It is *work conserving* if it satisfies (2) *and* $\sum_{i \in \mathbf{I}(j)} R_{i,j} = C_j$ for any $j \in \mathbf{J}$. The allowed rate for connection $i \in \mathbf{I}$ is given by $R_i = \min\{R_{i,j} \mid j \in \mathbf{J}(i)\}$, also called the *global rate allocation*. The vector $(R_1, R_2, \cdots, R_N)$ is called the *allocation vector* derived from the allocation matrix $[R_{i,j}]$.

For a given network, it is possible that there are multiple feasible allocation matrices. The questions are: which one is optimal and how to find it? This is the focal point of all ER marking algorithms and their primary function is to compute the *right* local allocation. Here, the objectives are two folds: *fairness* and *high utilization*.

2.2 Fairness Criteria

From (1), we observe that the quantity $C_j - \sum_{i \in \mathbf{I}(j)} MCR_i$, or the *excessive bandwidth* at link $j \in \mathbf{J}$, is greater than or equal to zero. The fairness issue is on how to distribute the excessive bandwidth amongst all traversing connections. There are two well-known fairness criteria: *equal share* and *proportional share*. In equal share, the excessive bandwidth is equally shared by all traversing connections. In proportional share, it is shared in proportion to their respective minimum cell rates. Let $S_{i,j}$ be the fair share of connection $i \in \mathbf{I}(j)$ at link $j \in \mathbf{J}$. Then,

$$S_{i,j} = \begin{cases} MCR_i + (C_j - SUM_j)/|\mathbf{I}(j)|, & \text{for equal share criterion,} \\ MCR_i + (C_j - SUM_j) \times MCR_i / SUM_j, & \text{for proportional share criterion,} \end{cases} \quad (3)$$

where $SUM_j = \sum_{i \in \mathbf{I}(j)} MCR_i$ and $|\mathbf{A}|$ is the total number of elements in set $\mathbf{A}$. It can be shown that if the condition in (1) is satisfied then $S_{i,j} \geq MCR_i$ for any $j \in \mathbf{J}$ and $i \in \mathbf{I}(j)$. In this paper, we assume that *all* links in the network adopt either the equal share criterion or the proportional share criterion.

2.3 Bottleneck Links and Bottleneck Connections

One approach to rate allocation is to set $R_{i,j} = S_{i,j}$. This achieves fairness but may cause low link utilization. To improve, the concept of bottleneck link was introduced. For a given allocation matrix $[R_{i,j}]$, a link $j \in \mathbf{J}$ is said to be a *bottleneck link* if $\sum_{i \in \mathbf{I}(j)} R_i = C_j$ holds. A connection $i \in \mathbf{I}$ is said to be a *bottleneck connection* if it traverses at least one bottleneck link. It can be shown that if an allocation matrix $[R_{i,j}]$ is work conserving, then there exists at least one bottleneck link in the network and, for every link $j \in \mathbf{J}$, there exists at least one connection $i \in \mathbf{I}(j)$ that is bottlenecked.

2.4 Max-min Fair Allocation

An optimal allocation should be the one that maximizes the network resource utilization and maintains fairness amongst all connections. This is called the *max-min fair allocation*. A procedure has been proposed to compute the max-min fair allocation [5]. In Table 1, we rephrase this procedure for a more general scenario described in the previous subsections.

Table 1. Max-Min Fair Allocation – The Traditional Approach

1. At each link j, allocate each connection its fair share $S_{i,j}$;
2. At each bottleneck link j, mark each connection i and assign it the rate R_i;
3. Decrease capacities of all links by the total capacity consumed by the marked connections traversing these links.
4. Consider a reduced network with all link capacities adjusted as

above and with marked connections removed. Repeat the procedure until all connections are assigned their rates.

2.5 The Virtual Bandwidth

We see that, in the traditional procedure, each link simply removes traversing bottleneck connections, decreases its capacity by the total bandwidth consumed by these bottleneck connections, and then redistribute the remaining bandwidth among other (non-bottleneck) connections using the selected fairness criterion. Here, we describe a procedure that takes the *opposite direction* for computing the max-min fair allocation. Instead of decreasing the link capacity, we *increase* the link capacity to a level that is *above* its actual or real value, hence the term *virtual bandwidth.*

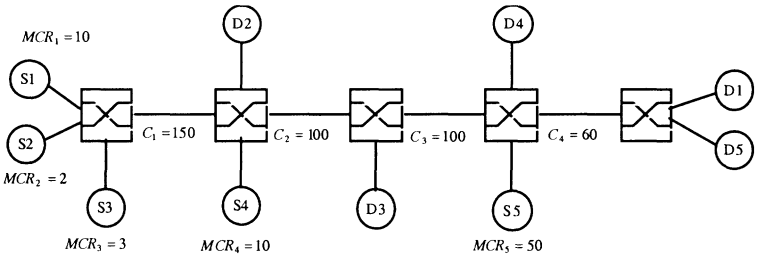

Figure 1. An Example.

To explain, let us consider Link 1 in Figure 1. According to proportional share criterion, it allocates 100, 20, and 30 to connections 1, 2, and 3, respectively. Suppose that connection 1 is bottlenecked elsewhere (link 4) in the network and is allowed to transmit only at the rate of 10. In the traditional procedure, the link would decrease the total link capacity of 150 by 10 and then redistribute the remaining 140 between connections 2 and 3. This gives 10 to connection 1, 56 to connection 2, and 84 to connection 3.

In the virtual bandwidth approach, we *increase* the link capacity to its virtual bandwidth, which is 420 in this case, and then redistribute the *total* virtual bandwidth of 420 among *all* connections, *including* the one that is bottlenecked elsewhere. This will give 280 to connection 1, 56 to connection 2, and 84 to connection 3. We notice that since connection 1 is bottlenecked elsewhere at the rate of 10, assigning a rate *greater* than its previous allocation *will not* cause over allocation.

In principle, the virtual bandwidth should be at such a capacity level that the sum of the local rates of all non-bottleneck connections plus the sum of the global rates of all bottleneck connections is equal to the real link capacity, hence achieving 100% link utilization. For the above simple example, let $\tilde{C}_1$ be the virtual bandwidth of Link 1. Then, $2\tilde{C}_1/15$ and $3\tilde{C}_1/15$ are the new allocations to connections 2 and 3 (both unbottlenecked), respectively. Hence, $\tilde{C}_1$ should satisfy

$$2\tilde{C}_1/15+3\tilde{C}_1/15+10=C_1=150. \quad (4)$$

Solving the above for $\tilde{C}_1$ yields $\tilde{C}_1=420$.

As in the traditional method, computing the max-min fair allocation using virtual bandwidth is also done in an iterative manner. For $j\in \mathbf{J}$, let $\tilde{C}_j$ be the virtual bandwidth of link j and $\tilde{C}_j(k)$ be the virtual bandwidth computed at the k^{th} iteration. Let $R_{i,j}(k)$ and $R_i(k)$ be, respectively, the local and the global allocations for the k^{th} iteration. Also, let $\mathbf{I}_B(j,k)$ ($\mathbf{I}_U(j,k)$) be the set of connections traversing link j that are bottlenecked elsewhere (unbottlenecked anywhere) at the k^{th} iteration.

Given the results of the k^{th} iteration, the virtual bandwidth $\tilde{C}_j(k+1)$ is computed as follows. If all connections traversing link j are bottlenecked either elsewhere or at this link, then $\tilde{C}_j(k+1)=\tilde{C}_j(k)$. Otherwise, $\tilde{C}_j(k+1)$ should assume such a value that the sum of the local allocations $R_{i,j}(k+1)$ for non-bottleneck connections plus the sum of global allocations $R_i(k+1)$ of bottleneck connections should be equal to the real link capacity C_j, i.e.,

$$\sum_{i\in\mathbf{I}_U(j,k)} R_{i,j}(k+1)+\sum_{i\in\mathbf{I}_B(j,k)} R_i(k+1)=C_j. \quad (5)$$

Since

$$R_{i,j}(k+1)=\begin{cases} MCR_i+\left(\tilde{C}_j(k+1)-SUM_j\right)/|\mathbf{I}(j)|, & \text{for equal share,} \\ MCR_i+\left(\tilde{C}_j(k+1)-SUM_j\right)\times MCR_i/SUM_j, & \text{for proportion al share,} \end{cases} \quad (6)$$

for any $i\in\mathbf{I}(j)$, and $R_i(k+1)=R_i(k)$ for any $i\in\mathbf{I}_B(j,k)$, we can solve (5) and (6) simultaneously for $\tilde{C}_j(k+1)$ and obtain

$$\tilde{C}_j(k+1)=\begin{cases} \left(C_j-\sum_{i\in\mathbf{I}_U(j,k)} MCR_i-\sum_{i\in\mathbf{I}_B(j,k)} R_i(k)\right)\times N_j/|\mathbf{I}_U(j,k)|+SUM_j, & \text{equal share,} \\ \left(C_j-\sum_{i\in\mathbf{I}_B(j,k)} R_i(k)\right)\times SUM_j/\sum_{i\in\mathbf{I}_U(j,k)} MCR_i, & \text{prop. share.} \end{cases} \quad (7)$$

The iterative procedure for computing the max-min fair allocation using virtual bandwidth is outlined in Table 2.

Table 2. Max-Min Allocation: A Virtual Bandwidth Approach

1.	Set $k=0$ and $\tilde{C}_j(0)=C_j$ for all $j\in\mathbf{J}$.
2.	Compute $R_{i,j}(k)$ using formula (6) for all $j\in\mathbf{J}$ and all $i\in\mathbf{I}(j)$.
3.	For each link j, determine whether it is a bottleneck. If so, mark any unmarked traversing connections.
4.	For each link j, determine whether it has both marked and unmarked connections traversing it. If so, increase its current virtual bandwidth $\tilde{C}_j(k)$ to $\tilde{C}_j(k+1)$ using (7). Otherwise, set $\tilde{C}_j(k+1)=\tilde{C}_j(k)$.
5.	Terminate if all connections have been marked. Otherwise, increase k by one and go to step 2.

$[R_{i,j}(k)]$ may not be feasible with respect to the real capacity C_j. However, it is always feasible and work conserving with respective to the $\tilde{C}_j(k)$. Furthermore, if a connection is bottlenecked at iteration k, then it remains to be bottlenecked in iteration (k+1) and its global allocation will not be affected by the re-distributions of bandwidth occurred at various links in the network. These properties ensure that the iterative process will converge to the max-min fair allocation. Once $\tilde{C}_j$ has been obtained, then the local allocation $R_{i,j}$ for connection $i \in \mathbf{I}(j)$ can be computed as follows:

$$R_{i,j} = \begin{cases} MCR_i + \left(\tilde{C}_j - SUM_j\right) / |\mathbf{I}(j)|, & \text{for equal share criterion,} \\ MCR_i + \left(\tilde{C}_j - SUM_j\right) \times MCR_i / SUM_j, & \text{for proportional share criterion,} \end{cases} \quad (8)$$

regardless of whether connection $i \in \mathbf{I}(j)$ is bottlenecked or not.

3. VIRTUAL BANDWIDTH BASED ER MARKING ALGORITHM

We now present the proposed new algorithm T-VERMA, aiming at achieving the objectives outlined in the Introduction. T-VERMA consists of two major operations: *rate allocation* and *virtual bandwidth estimation*.

3.1 Rate Allocation

This operation is executed each time the link processes a backward RM cell. Here, it first computes the local allocation for the connection using formula (8). It then compares the local allocation with the value in the ER field of the RM cell and overwrites the ER field with the local allocation if the former is larger than the latter. This operation is summarized in table 3 for equal share criterion and table 4 for proportional share criterion.

Table 3. The Rate Allocation for the case of Equal Share Criterion.

```
If (A BRM Cell Departs) {
     R ← BRM_MCR + (VB * Adjust(ABR_Q) - SUM) / N;
     If (R < BRM_ER) BRM_ER = R;
}
```

Table 4. Rate Allocation - Proportional Share Criterion.

```
If (A BRM Cell Departs) {
     R ← VB * Adjust(ABR_Q) * BRM_MCR / SUM;
     If (R < BRM_ER) BRM_ER = R;
}
```

In the above algorithm, we use the function `Adjust()` to adjust the virtual bandwidth, a technique used by a number of ER marking algorithms (see [12] and [16]). The value of `Adjust()` depends on the current queue length. The longer the queue, the smaller the value of `Adjust()`. The idea

here is to reserve some bandwidth to drain the queue when its length exceeds a certain threshold [1]. A simple example is given as follows:

$$Adjust(x) = \begin{cases} c - \dfrac{c-1}{T_0} x, & 0 \le x \le T_0, \\ 1, & T_0 < x \le T_1, \\ 1 - \dfrac{1-d}{T_2 - T_1}(x - T_1), & T_1 < x \le T_2, \\ d, & x > T_2, \end{cases} \tag{9}$$

where $c > 1 > d > 0$ and T_0, T_1, and T_2 are queue thresholds. The interval $[0,T_0]$ defines the underload region, $[T_0,T_1]$ is the steady-state region, $[T_1,T_2]$ is the overload region, and $[T_2,\infty]$ is the heavy-load region. More details on this topic and some excellent examples of this type of queue control functions can be found in [28].

3.2 Virtual Bandwidth Estimation

In this operation, T-VERMA estimates the virtual bandwidth of the link. Here, the virtual bandwidth is progressively updated. Each time, it is unchanged, increased, or decreased from its present value, depending on the link congestion state, the available bandwidth, and the total ABR traffic volume. In steady-state, the estimated virtual bandwidth is expected to converge to the true virtual bandwidth of the link after some iterations.

T-VERMA periodically monitors the available bandwidth for ABR connections and the total ABR traffic load. Let $\{t_0, t_1, t_2, \cdots, t_k, \cdots\}$ be the sequence of time epochs at which the measurements are made. The time interval $(t_k, t_{k+1}]$ is called the k^{th} *measurement interval.*

3.2.1 Basic Updating Rules

First, for the link concerned, we define the following variables:

$ABR\_Capacity(t)$: the measured available capacity for ABR connections;

$ABR\_Load(t)$: the measured total ABR traffic load at time t;

$ABR\_U(t)$: the ABR utilization at time t and is equal to $ABR\_Load(t)$ / $ABR\_Capacity(t)$;

$ABR\_Q(t)$: the length of the ABR queue at time t;

$\tilde{C}(t)$: the true virtual bandwidth of the link, calculated in theory based on values of $ABR\_Capacity(t)$ of all links; and

$VB(t)$: the estimated virtual bandwidth at time t.

The rules for updating the estimated virtual bandwidth $VB(t)$ are as follows. If $ABR\_U(t) = 1$, $VB(t)$ is unchanged. If $ABR\_U(t) > 1$, $VB(t)$ is decreased to avoid sustained congestion. If $ABR\_U(t) < 1$, $VB(t)$ is increased to avoid under-utilization.

3.2.2 The Updating Function

The magnitude of the change (decrease or increase) of $VB(t)$ depends on the values of $ABR\_U(t)$ and $ABR\_Q(t)$. Suppose that an update is to be carried out at time t. Then, we update the estimated virtual bandwidth using the following formula:

$$VB(t^+) = f(ABR\_Q(t), ABR\_U(t)) \cdot VB(t), \tag{10}$$

where $VB(t^+)$ is the new virtual bandwidth and $f(x, y)$ is a non-negative function, called the *virtual bandwidth updating function*. The form of the function $f(x, y)$ is not strictly specified. However, it should be non-increasing in x and in y; below 1.0 as either x, or y, or both approach infinity; and above 1.0 as either x, or y, or both approach zero. As an example, we present the following virtual bandwidth updating function:

$$f(x, y) = \begin{cases} \max(1, g(x)), & 0 \le x < \infty;\ 0 \le y < U_0, \\ g(x), & 0 \le x < \infty;\ U_0 \le y \le U_1, \\ \min(1, g(x)), & 0 \le x < \infty;\ U_1 < y < \infty, \end{cases} \tag{11}$$

where $g(x) = 1 - \alpha + \alpha \cdot q(x)$, α is a smoothing factor, and $q(x)$ is a standard queue control function. Typical values of α should be around 0.1. An example of $q(x)$ is the piece-wise hyperbolic function given in [16] as:

$$q(x) = \begin{cases} \dfrac{b \cdot Q_0}{(b-1) \cdot x + Q_0}, & 0 \le x \le Q_0, \\ 1.0, & Q_0 < x \le Q_1, \\ \dfrac{a \cdot (Q_2 - Q_1)}{(1-a) \cdot x + a \cdot Q_2 - Q_1}, & Q_1 < x \le Q_2, \\ a, & x > Q_2, \end{cases} \tag{12}$$

where a (<1.0) and b (>1.0) are respectively the minimum and maximum values of $q(x)$. The parameters U_0 and U_1 specify a targeted utilization band and the parameters Q_0, Q_1, and Q_2 specify a number of congestion zones.

3.2.3 RM Cell Driven vs. Traffic Driven

We now discuss the issue of *when* to update $VB(t)$. First, we point out that the timing of updating control variables is crucial to an ER marking algorithm. Most existing ER marking algorithms update at least some of their control variables at the time of processing RM cells and hence are said to be of *RM cell driven*. For these algorithms, updating control variables requires some information carried in RM cells and therefore *has to* be done when processing RM cells. The major disadvantage of being RM cell driven is that it can cause the control variables and the link congestion state out of synchronization, especially for progressive ER marking algorithms.

Table 5. Virtual Bandwidth Estimation.

```
If (End of Measurement Interval) {
    Update ABR_Capacity according to traffic measurements;
    Update ABR_Load according to traffic measurements;
    ABR_U ← ABR_Load / ABR_Capacity;
```

```
        VB ← VB * f(ABR_Q, ABR_U);
        If (VB < ABR_Capacity) VB ← ABR_Capacity;
    }
```

Here we adopt a different approach: to have the update of control variables driven by traffic. This approach ensures that the values of control variables can more accurately represent the current congestion-state of the link. A necessary condition for being traffic-driven is that the update does not need information carried by RM cells. From the previous subsection, we can see that updating the estimated virtual bandwidth $VB(t)$ requires only $ABR\_U(t)$ and $ABR\_Q(t)$, and can be done without any information from RM cells. Indeed, *this is the most important advantage of combining virtual bandwidth with progressive marking*! We choose the updating time to be at the end of each measurement interval $(t_k, t_{k+1}]$. The value of $ABR\_Q(t)$ is always accurate. To summarize, we present the pseudo code for the second (and last) part of the virtual bandwidth algorithm (T-VERMA) in Table 5.

4. SIMULATION RESULTS

We will now show simulation results for three algorithms: ERICA with Weighted Fairness (ERICA-WF) [27], EDERA [12], and our new algorithm T-VERMA. We have focused on three scenarios. In the first, we test the responsiveness of the three algorithms when there is a sudden increase in bandwidth and a sudden decrease in bandwidth. We also test in this scenario whether or not these algorithms converge to the max-min fair allocation after each change. In the second scenario, we test how the algorithms perform when some of the sources are non-persistent, meaning that they may transmit at a rate below their allowed cell rates and sometimes even below their MCRs. In Scenario 3, we compare the performance of the three algorithms in the presence of VBR sources.

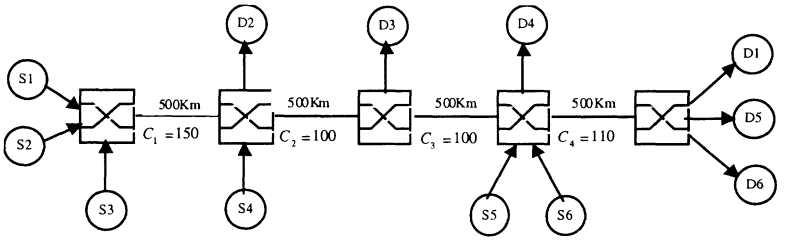

Figure 2. Network Configuration for Experiments.

All experiments are designed based on the network configuration shown in *Figure 2*. Link capacity is in Mbps. The distance between a switch and a source or a destination is 10 kilometers.

Due to space limitation, only brief simulation results (ACR plots) of scenario 1 are shown here (see *Figure 3*). It is observed that both EDERA and T-VERMA response to the sudden significant change in bandwidth very well, with EDERA being slightly better. ERICA-WF reacts well (so do the

other two) to the sudden decrease in bandwidth but its response to the sudden increase in bandwidth is relatively slow.

Simulation results for scenario two indicates that in T-VERMA, non-persistent sources are treated fairly and are allocated a rate proportional to their respective MCR at the bottleneck link. However, it is observed that both EDERA and ERICA-WF seem to penalize non-persistent sources by allocating them a rate that is smaller, in proportion, than the rates received by their persistent peers at a bottleneck link.

In scenario 3, it is observed that all three algorithms respond to traffic disruptions caused by the VBR source very well!

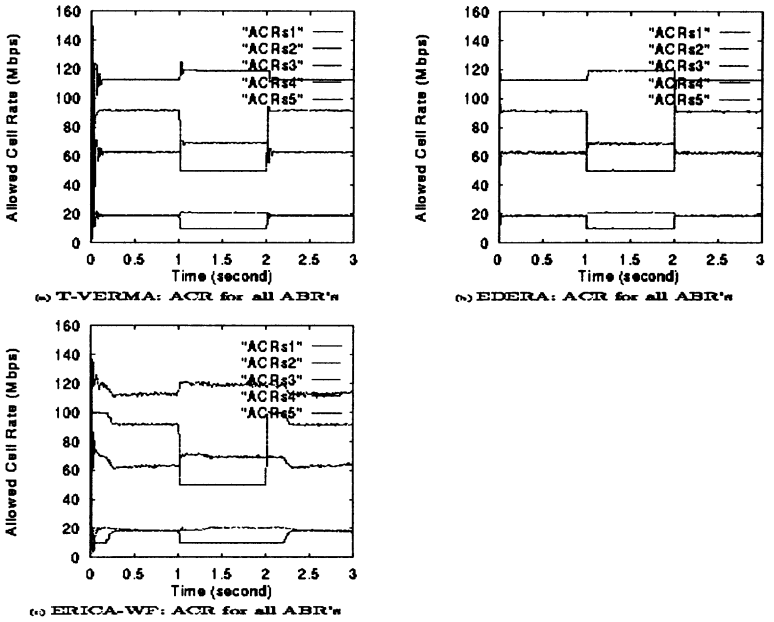

Figure 3. Comparison of T-VERMA, EDERA and ERICA-WF under scenario 1.

5. CONCLUSIONS

In this paper, we introduced the concept of virtual bandwidth and showed how it can be used to do global calculation of the max-min fair allocation for ABR connections in an ATM network. Based on the concept of virtual bandwidth, we developed a simple ER marking algorithm, called T-VERMA, that has O(1) storage complexity and O(1) computational complexity.

We have conducted extensive simulation studies on T-VERMA and compared it with two fairly recent algorithms, namely EDERA [12] and ERICA-WF [27]. From these studies, we can conclude that the responsiveness of T-VERMA is comparable to that of the more sophisticated

EDERA and ERICA-WF. Furthermore, T-VERMA is capable of treating all ABR connections, including those that are non-persistent, in a fair fashion and can converge to max-min fair allocation in steady state. Finally, T-VERMA has shown remarkable robustness in presence of VBR connections.

References

[1] Arulambalam, A., X. Chen, and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM Networks," IEEE Communication Magazine, vol. 34, 92-100, 1996.

[2] Arulambalam, A., X. Chen, and N. Ansari, "An Intelligent Explicit Rate Control Algorithm for ABR Service in ATM Networks," ICC'97, 200-204, 1997.

[3] G. Bianchi, L. Fratta, and L. Musumeci, "Congestion Control Algorithms for the ABR Service in ATM Networks," IEEE GLOBECOM, 1996.

[4] F. Bonomi and K.W. Fendick, "The Rate-Based flow Control Framework for the Available Bit Rate ATM Service," IEEE Network Magazine, March/April 1995.

[5] Charny, A., D. Clark, and R. Jain, "Congestion Control with Explicit Rate Indication," Proc. ICC'95, June 1995.

[6] Chiussi, F., A. Arulambalam, Y. Xia, and X. Chen, "Explicit Rate ABR Schemes Using Traffic Load as Congestion Indicator," IEEE GLOBECOM'97, 76-84, 1997.

[7] Chiussi, F., T. Wang, "An ABR Rate-Based Congestion Control Algorithm for ATM Switches with Per-VC Queueing," IEEE GLOBECOM'97, 2108-2117, 1997.

[8] Chiussi, F., Y. Xia, and V. Kumar, "Dynamic Max Rate Control Algorithm for Available Bit Rate Service in ATM networks", IEEE GLOBECOM'96, 2108-2117, 1996.

[9] F.M. Chiussi, Y. Xia, and V.P. Kumar, "Virtual Queueing Techniques for ABR Service: Improving ABR/VBR Interaction," IEEE INFOCOM'97, 1997.

[10] S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal, and B. Vandalore, "On Determining the Fair Bandwidth Share for ATM Connections in ATM Networks," Proceedings of the IEEE International Conference on Communications (ICC) 1998, June 1998.

[11] N. Ghani and J.W. Mark, "Dynamic Rate-Based Control Algorithm for ABR Service in ATM Networks", IEEE GLOBECOM'96, Vol. 2, Nov. 1996, London, UK, pp.1074-1079.

[12] N. Ghani and J.W. Mark, "An Enhanced Distributed Explicit Rate Allocation for ABR services", 15th International Teletraffic Congress (ITC-15), Washington D.C., June 1997.

[13] Jain, R., S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan, "ERICA Switch Algorithm: A Complete Description," ATM Forum 96-1172, August, 1996.

[14] R. Jain, S. Kalyanaraman, and R. Viswanathan, "The OSU Scheme for Congestion Avoidance in ATM Networks: Lessons Learnt and Extensions," Performance Evaluation (North-Holland), Special Issue on

Traffic Control in ATM Networks, Vol. 31, No.1-2, November 1997, pp. 67-88.
[15] L. Kalampoukas, A. Varma, K.K. Ramakrishnan, An Efficient Rate Allocation Algorithm for ATM Networks Providing Max-Min Fairness," High Performance Networking VI. IFIP sixth Int. Conf. on High Performance Networking, 1995.
[16] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, and B. Vandalore, "The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks," Submitted to IEEE/ACM Trans. on Networking, Nov. 1997, http://www.cis.ohiostate.edu/~jain/papers/erica.htm.
[17] S. Muddu, F.M. Chiussi, C. Tryfonas, and V.P. Kumar, "Max-Min Rate Control Algorithm for Available Bit Rate Service in ATM Networks", Proc. ICC'96, June 1996.
[18] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, "Rate-Based Congestion Control for ATM Networks," Computer Comm. Review, ACM SIGCOMM, 1995.
[19] Roberts, L., "Enhanced PRCA (Proportional Rate-Control Algorithm)," ATM Forum 94-0735R1, 1994.
[20] Shirish S. Sathaye, "ATM Forum Traffic Management Specification 4.0," ATM Forum af-tm-0056.000, April 1996.
[21] K.-Y. Siu and H.-Y. Tzeng, "Adaptive Proportional Rate Control (APRC) with Intelligent Congestion Indication," ATM Forum 94-0888, September 1994.
[22] K.-Y. Siu and H.-Y. Tzeng, "Limits of Performance in Rate-Based Control Schemes," ATM Forum Contribution 94-1077, November 1994.
[23] Siu, K. and T. Tzeng, "Intelligent Congestion Control for ABR Service in ATM Networks," Computer Communication Review, 24(5), 81-106, October 1995.
[24] Wei K. Tsai, Yuseok Kim and Lee Hu, "ASAP: A Non-per-VC Accounting Max-Min Protocol for ABR Flow Control with Optimal Convergence Speed," IEEE SICON'98, Singapore, June 30 - July 3, 1998.
[25] Tsang, D., W. Wong, S. M. Jiang and E. Liu, "A fast Switch Algorithm for ABR Traffic to Achieve Max-Min Fairness," IEEE 1996 International Zurich Seminar on Digital Communications, February 19-23, 1996.
[26] Tsang, D. and W. Wong, "A New Rate-Based Switch Algorithm for ABR Traffic to Achieve Max-Min Fairness with Analytical Approximation and Delay Adjustment, INFOCOM'96, 1174-1181, 1996.
[27] B. Vandalore, S. Fahmy, R. Jain, R. Goyal, and M. Goyal, "A Definition of General Weighted Fairness and its Support in Explicit Rate Switch Algorithms," submitted to ICNP'98, May 1998, http://www.cis.ohiostate.edu/~jain/papers/icnp98_bv.htm.
[28] B. Vandalore, R. Jain, R. Goyal, and S. Fahmy, "Design and analysis of queue control functions for explicit rate switch," Submitted to the IC3N'98, May 1998. Also available at http://www.cis.ohio-state.edu/~jain/papers.html.