Minimum Rate Guarantee without Per-Flow Information

Yuseok Kim, W.K. Tsai, Mahadevan Iyer, and Jordi Ros
Department of Electrical and Computer Engineering
University of California, Irvine, CA 92697
Email: {ykim, miyer, jros, wtsai}@ece.uci.edu

ABSTRACT

This paper introduces a scalable maxmin flow control protocol which guarantees the minimum rate for each connection-oriented flow without requiring per-flow information. The protocol is called MR-ASAP (Minimum Rate guaranteeing Adaptive Source-link Accounting Protocol). MR-ASAP is an extension of ASAP [Tsa98a, Kim99], the first exact maxmin flow control protocol for best-effort connection-oriented traffic in integrated service networks, without requiring per-flow accounting at the intermediate network node. In the classical maxmin computation, only the maximum rate constraints are considered; in this paper the minimum rate requirements are treated similarly as the maximum rate constraints. Existing protocols that achieve exact maxmin optimality with minimum rate guarantee require per-flow information and complex computation such as sorting of the minimum rates at the switch. By generalizing the concept of constraint, the complex sorting and per-flow accounting required in the existing protocols are avoided. Simulation demonstrates fast convergence to optimality. The convergence of MR-ASAP is also proved analytically.

Key words: Generalized maxmin, flow control, and minimum rate guarantee.

1. INTRODUCTION

Rate-based flow control for best-effort connection-oriented traffic in the integrated service networks – such as ATM networks – has dual goals at the steady state: maximizing bandwidth utilization and achieving fairness among all the virtual connections (VC's). This leads to the concept of maxmin fairness originally proposed by Hayden and Jaffe [Hay81, Jaf81]. Maxmin optimality has been adopted as a network bandwidth sharing criterion for best-effort traffic by the ATM forum in the TM 4.0 specification [ATM96]. An exact maxmin protocol is defined to be the one that eventually assigns the maxmin rate to all the VC's. So far, quite a few exact maxmin protocols have been proposed [Aru96, Cha95, Hou98, Jai96, Kal95, Kal97, Kim99, Tsa98]. However, in order to achieve maxmin optimality, most exact maxmin protocols have employed per-VC accounting at the switch. To the best of our knowledge, the protocol ASAP presented in [Tsa98a, Kim99] is the only exact maxmin protocol that does not require per-VC accounting at the switch. Since ASAP does not use per-VC accounting, the protocol is scalable.

Submitted to ICNP'99.

Even though maxmin protocols are originally targeted for the best-effort traffic, it is good to have a certain degree of guarantee, especially, in multimedia communication, which necessitates minimum rate guarantee (MRG) by the protocol. Among those exact maxmin protocols, two protocols provide MRG [Hou98, Kal97]. The Generalized Max-Min (GMM) protocol by Hou et al. [Hou98] is an extension of the maxmin protocol proposed by Charny et al. [Cha95] by incorporating MRG. In order to guarantee the minimum rate for an individual VC, Hou's protocol not only employs per-VC accounting, but also involves complex computation and sorting among VC's in terms of their minimum rates, which is obviously not desirable. It has been shown in [Tsa99] that the complex computation and sorting required by their protocol is not mandatory. Another maxmin protocol by Kalampoukas [Kal97] also requires per-VC accounting.

In this paper, the original maxmin optimality is generalized by incorporating a minimum rate requirement and a maximum rate limitation for each connection. The concept of "constraint VC" is extended to include not only those VC's which cannot *increase* their rates above their peak cell rates (PCR's) but also those VC's which cannot *decrease* their rates below their minimum cell rates (MCR's). This generalization leads to simple modification of existing protocols to compute maxmin rates when the sources have non-zero MCR requirements.

This paper introduces a maxmin protocol that guarantees the MCR for each VC and yet does not require per-VC accounting. This protocol is called MR-ASAP (Minimum Rate guaranteeing Adaptive Source-link Accounting Protocol). MR-ASAP is an extension of the protocol ASAP presented in [Tsa98, Kim99]; as a derivative of ASAP, MR-ASAP is the first and only exact maxmin protocol that guarantees MCR requirements without employing per-VC accounting. Simulation demonstrates that MR-ASAP converges fast to maxmin optimality under various network disturbances such as dynamic changes in the available bandwidth and dynamic arrival and departure of VC's. The convergence of MR-ASAP is also proved analytically.

In this paper, much of terminology is borrowed from the ATM community. However, the protocol MR-ASAP can be implemented in any connection-oriented network that needs minimum rate guarantee in a scalable manner. This paper is organized as follows: Section 2 introduces the concept of generalized maxmin optimality. Section 3 describes in detail the protocol MR-ASAP with some examples. Section 4 shows some simulation results. Section 5 describes an abbreviated convergence proof of the MR-ASAP protocol. The paper is concluded in Section 6.

2. GENERALIZATION OF MAXMIN OPTIMALITY

There exist two cases where a source can be constrained on the entire VC: the source could be limited by the maximum rate (PCR) it can transmit or associated with the minimum cell rate (MCR) requirement imposed by the network-user contract. For the case of non-zero MCR, the ATM forum has adopted a few options in the TM 4.0 specification [ATM96]. Among those options, the most difficult one to handle is the option of Max(Maxmin rate, MCR): the source must be allocated the MCR if its maxmin rate is smaller than its MCR, otherwise, the source must be allocated the maxmin rate.

The key idea to solve this problem is to recognize that the MCR constraint is a constraint by itself. In the original maxmin formulation, a constrained VC at a link is one that is unable to *increase* its rate to reach the advertised rate at the link. A straightforward generalization is to define a constrained VC at a link as one that is unable to either *increase* or *decrease* its rate to reach the advertised rate. This simple observation, trivial it may appear to be, is a key to design distributed maxmin protocols without per-VC computation. There appear to be two maxmin protocols from the literature that handle the MCR constraint [Hou98, Kal97]. The GMM protocol presented in [Hou98] requires per-VC-sorting, which is quite undesirable. We shall adopt the definition of generalized maxmin optimality by Hou et al. For each VC i, let PCR_i and MCR_i denote the PCR and MCR, respectively.

Assumption 1. The sum of the MCR's of all the VC's crossing each link is less than the capacity of the link.

Let A_i denote the rate of VC i. Let C_j denote the ABR (available bit rate) link capacity at link *j*. Let V_j denote the set of VC's crossing link *j*. Let F_j denote the total input rate from all the ABR VC's crossing link *j*, thus $F_j = \sum_{i \in V_j} A_i$.

Definition 1. A vector of source rates $\{A_i\}$ is said to be *feasible* if $F_j \le C_j$ for all link *j*, and for each VC i, $MCR_i \le A_i \le PCR_i$.

Definition 2. A vector of source rates $\{A_i\}$ is said to be *Generalized Maxmin (GMM) optimal* if it is feasible and for each VC *i*, A_i cannot be increased while maintaining feasibility without decreasing A_k for some VC *k* for which $A_k \le A_i$.

Let Vm_j denote the set of VC's with the condition that $A_i = MCR_i$ at link j.

Definition 3. The quantity R_j defined below is called the advertised rate for link j:

$$R_j \equiv \begin{cases} \max\{A_i \mid i \in V_j,\ A_i > MCR_i\} & \text{if } Vm_j \ne V_j,\ F_j = C_j; \\ \infty & \text{if } F_j < C_j; \\ 0 & \text{if } Vm_j = V_j,\ F_j = C_j. \end{cases}$$

Definition 4. Link j is said to be a bottleneck link for VC i if $F_j = C_j$, $A_i > MCR_i$, and for every VC k crossing link j with $A_k > MCR_k$, and $A_i \geq A_k$; furthermore, VC i is said to be **BC** (*bottleneck link constrained*) at link j.

Definition 5. VC i is said to be **MC** (*MCR constrained*) if $A_i = MCR_i$ and $A_i > \min\{R_j | j \in P_i\}$. Similarly, VC i is said to be **PC** (*PCR constrained*) if $A_i = PCR_i$ and $A_i < \min\{R_j | j \in P_i\}$.

Definition 6. VC i at link j is said to be a *constrained* VC at link j if

(1) VC i is BC at link $k \neq j$; or

(2) VC i is MC; or

(3) VC i is PC.

Therefore, if a VC is constrained at a link, it is either *lower-bound constrained* (MC) or *upper-bound constrained* (either PC or BC at a different link). VC i at link j is said to be an *unconstrained* VC at link j if it is not constrained at the same link. This definition arises from the view that VC i is unconstrained *from link j point of view* because VC i is not constrained by some other means to set its rate lower or higher than the advertised rate.

The main difference between the above bottleneck definitions and that of [Hou98] is that we impose more technical conditions such as the advertised rate conditions to ensure that the upper bound and lower bound constraints are treated separately and cleanly. This turns out to be the key to design an optimality condition which requires no sorting in the corresponding protocol.

In the network maxmin literature, the advertised rate is commonly defined to be the maximum VC rate at a link. But this turns out to be highly inadequate. The complications come from two sources: the minimum rate constraint and *pseudo-saturation*, which is defined below.

Definition 7. A link j is said to be *saturated* if the total flow is equal to its capacity: $F_j = C_j$.

Definition 8. A link j is said to be *pseudo-saturated* if every VC at the link is either BC somewhere else or PC at the link and its capacity is *not* fully utilized: $F_j < C_j$.

The case of pseudo-saturated link creates many technical difficulties in the maxmin problem. In fact, most maxmin protocols were not designed with pseudo-saturation in mind and can be shown to be nonconvergent, see [Tsa99]. With the minimum rate constraint, there comes another complication: VC's might be constrained at a minimum rate higher than other VC's whose rates are not at their respective minim.

In the definition of advertised rate, the first case is the ordinary case where link j has at least one unconstrained VC. The second case is one in which link j is pseudo-saturated. The third case is a pathological case where all VC's are at their respective minimum rates and the link is saturated. In this case, the entire available bandwidth is used by all the VC's that are not allowed to drop their rates. Notice that we do not define R_j for the case where $Vm_j = V_j$ and $F_j > C_j$. This is an infeasible (hence uninteresting) case where no feasible solution exists.

Intuitively, the advertised rate at a link is the largest rate of the VC's with rates higher than their respective minimum rates at the link so that the link is saturated. If saturation is impossible, the advertised rate is defined to be at least as large as the largest peak rate constraint in the network, i.e., ∞.

Definition 9. The ratio C_j / N_j is called the *fair share* at link j; it is the fair share for all the VCs crossing link j while assuming none of them is constrained somewhere else.

Definition 10. The ratio $(C_j - Fc_j - Fm_j)/(N_j - Nc_j - Nm_j)$ is called the *remaining fair share* at link j, where Fc_j is the sum of rates for upper-bound constrained VCs (PC or BC at another link) at link j, Fm_j is the sum of rates for lower-bound constrained (MC) VCs at link j, Nc_j is the number of upper-bound constrained VCs at link j and Nm_j is the number of MC VCs at link j, if $N_j - Nc_j - Nm_j > 0$.

Intuitively the remaining fair share at a link is the fair share after taken the bandwidth of its constrained VC's.

Proposition 1 (Bottleneck Optimality Condition). A feasible source rate vector $\{A_i\}$ is GMM optimal iff for every VC, it is either BC at some link, or is MC, or PC.

Proof. (Only-If part): Suppose that the feasible source rate vector $\{A_i\}$ is optimal. To arrive at a contradiction, assume that there exists a VC i which is not BC at every link crossed by the VC, and is not MC, and is not PC. For every link j crossed by VC i, there are two cases to consider: (1) Link j is saturated and (2) Link j is not saturated. First, consider Case (1). Since VC i is not BC, there must exist a VC k such that $A_k > A_i$ and $A_k > MCR_k$. Thus we can increase A_i without violating the capacity constraint by dropping A_k. Now, consider Case (2). Since VC i is not PC, we can increase A_i without dropping any other VC's rate to reach the capacity. Now, we can increase A_i at every link by a non-zero amount without increasing the rate of A_k for any $A_k \le A_i$ while maintaining feasibility. This contradicts the optimality of the feasible source rate vector $\{A_i\}$.

(If part): Conversely, suppose that the feasible source rate vector $\{A_i\}$ satisfies the Bottleneck Optimality Condition. Now, for every VC i, there are three cases to consider: (1) it is PC, or (2) it is MC, or (3) it is BC at some link. Case (1): Since VC i is PC, its rate cannot be increased without violating its own peak constraint. Case (2): There exists a link j such that $R_j = \min\{R_k | k \in P_i\}$. Now there are two subcases to consider: Case (2a): Link j is saturated and Case (2b): Link j is pseudo-saturated. Consider Case (2a): If we increase the rate A_i, we must drop some other rate in order to satisfy the capacity constraint. We cannot drop any VC whose rate is at its minimum. Thus the only rate we can drop must be from any

VC k with the constraint: $A_k > MCR_k$. But by the definition of MC, we have $A_i > \max\{A_k \mid A_k > MCR_k\}$. Thus if we increase A_i, we must decrease some A_k with the condition that $A_i > A_k$. Consider Case (2b). By the definition of the advertised rate and MC, we must have, $MCR_i > \max\{PCR_k \mid k \in V_j\}$, which is impossible. Case (3) VC i is BC at a link j. Thus, in order to increase A_i without violating the capacity constraint, one must decrease a rate A_k such that $A_k \le A_i$ and $A_i > MCR_k$ at link j. ■

3. THE PROTOCOL MR-ASAP

3.1 Resource management (RM) cell

In MR-ASAP, the source end-system (SES) transmits a special cell called a resource management (RM) cell towards the destination end-system (DES) periodically (interval-basis) and/or every time after transmitting a certain number of data cells (counter-basis). This RM cell is called a forward RM (FRM) cell. Upon receiving an FRM cell, the DES returns another RM cell called a backward RM (BRM) cell towards the SES. While traveling along the forward and backward paths of a VC, RM cells deliver information from the SES to the links (switches) and vice versa.

The switch calculates the advertised rate *R* for each outgoing link, that is the maximum rate at which the switch allows the SES of any VC crossing the link to transmit cells providing that the VC is not MCR-constrained. If a VC is MCR-constrained (MC), the VC is allowed to transmit cells at its MCR. While traversing the backward path of a VC, a BRM cell picks up the minimum *R* and delivers it to the SES. Upon receiving a BRM cell, the SES compares the minimum *R* and its *MCR*. If the minimum *R* is larger than or equal to its *MCR*, the SES is allowed to transmit cells at that minimum *R*. Otherwise, the SES transmits cells at its *MCR*. The maximum rate at which the SES is allowed to transmit cells is called an allowed cell rate (ACR). Table 1 describes the structure of an RM cell. Note that the structure of the RM cell used in MR-ASAP is different from the one defined in [ATM96].

Table 1. RM cell structure

Field	Meaning
RM.ACR	Value of ACR of the SES
RM.Old_ACR	Previous value of ACR of the SES
RM.BID	ID of the bottleneck link in the VC's path
RM.Old_BID	Previous value of BID
RM.ER	(In FRM cell) Peak cell rate of the SES (In BRM cell) Advertised rate of the bottleneck link
RM.Hop_Count	Hop distance from the SES (used to identify each link in the VC's path)

3.2 Original BID (bottleneck ID) scheme

The core of the protocol MR-ASAP is the bottleneck ID (BID) scheme which provides two significant advantages: (1) it provides more stable convergence to the maxmin optimality [Tsa98b, Kim99] and (2) it eliminates the need for per-VC accounting in implementing maxmin protocols. In this subsection, the original BID scheme used in ASAP [Kim99] is briefly described.

The *Hop_Count* field in the RM cell is used as an ID of each link in the path of the VC. First, the SES initializes *RM.Hop_Count* to 0. Upon receiving an FRM cell, each switch increases *RM.Hop_Count* by one, and right before transmitting a BRM cell to the upstream switch, each switch decreases *RM.Hop_Count* by one. Therefore, the value of *RM.Hop_Count* effectively identifies each link in the VC's path. Note that a link may have different ID's relative to different VC's crossing the link.

When a BRM cell arrives at the switch, the explicit rate (ER) field in the BRM cell is compared against the advertised rate *R* at the switch, and updated as follows:

if RM.ER ≥ R, then RM.ER ← R and RM.BID ← RM.Hop_Count.

Therefore, when a BRM cell arrives at the SES, the *RM.ER* field will contain the smallest AR from all the links in its backward path and the *RM.BID* field indicates the ID of the link with the smallest AR. Note that if more than one link has the same smallest value of *R*, *RM.BID* carries the ID of the link closest to the SES.

Upon receiving a BRM cell, the SES updates its *ACR* according to *Max (MCR, Min (RM.ER, PCR))*. The value of *RM.BID* is then passed to the switches in the VC's path via the next FRM cell sent by the source.

Upon receiving an FRM cell, the switch determines the constraint status for each VC i as follows:

if RM.BID = RM.Hop_Count, then VC i *is considered unconstrained,*
otherwise, VC i *is considered constrained.*

In most rate-based maxmin protocols, the switch calculates the maxmin rates based upon the number of constrained VC's (*Nc*) and the total input flow from the constrained VC's (*Fc*) at the switch. In those protocols, the constraint status of a VC is determined according to the comparison between the value of *R* at the switch and the value of *ACR* in the FRM cell. This scheme is called a Rate Comparison (RC) scheme. Based upon the new constraint status, the switch updates its *R*. Therefore, in an RC scheme, it is possible that even though a VC is transmitting cells at a constant rate with a specific constraint status, the switch may change the VC's constraint status, which could in turn cause a rate change and possible oscillations.

In order to reduce inconsistency and oscillation problems, the RC scheme artificially enlarges the *R* updating interval (RUI) longer than the maximum feedback latency of VC's crossing the switch. The idea behind the longer RUI is to make sure that the sources of all the VC's have changed their rates according to the new *R*. However, feedback latency is very difficult to estimate accurately and thus determining the optimal RUI is very difficult. If the RUI is too long, the flow control becomes too slow to respond to the congestion and if the RUI is too short, oscillations could result.

In the BID scheme, the switch determines the constraint status of a VC based on the *BID* value in the FRM cell, not based on the comparison between the value of *R* at the switch and the value of *ACR* in the FRM cell. Thus, the scheme effectively reduces inconsistency and potential oscillations encountered in the RC scheme. A formal definition of inconsistency and examples comparing the effects of both RC scheme and BID scheme are presented in [Kim99].

One of the most distinguished features of the BID scheme is that the scheme is not required to estimate the feedback latency since the scheme allows the switch to update its *R* only when the switch receives the confirmation (in the form of *BID*) from any source crossing the switch. Thus the BID scheme guarantees updating *R* with an RUI exactly equal to the round-trip delay.

The second advantage of the BID scheme is that it eliminates the need for per-VC accounting. The reason that in most rate-based maxmin protocols, the switch has to store per-VC information (e.g., constraint status of each VC) is to estimate the values of *Nc* and *Fc*. However, in the BID scheme, the switch only remembers the aggregate information such as *Nc* and *Fc*, and the SES keeps *BID*, *Old_BID*, *ACR*, *Old_ACR*. The FRM cells carry a pair of *BID* and *Old_BID* and another pair of *ACR* and *Old_ACR*, based upon which the switch can correctly update *Nc* and *Fc*. An obvious objection to this scheme is that problems could result if the FRM cells got lost. Most of the time in the FRM cell, the values of *BID* and *Old_BID*, and the values of *ACR* and *Old_ACR* are identical. In these cases, loss of RM cells does not incur any harm. If the FRM cell gets lost when it carries a pair of different *BID* and *Old_BID* and/or a pair of different *ACR* and *Old_ACR*, some errors are introduced at the downstream switches. However, this problem can be easily handled by several different schemes such as "Fast Track" RM cell scheme proposed in [Tsa98]. In "Fast Track" RM cell scheme, RM cells are given higher priority at switches than data cells, which effectively prevents loss of RM cell due to congestion. The loss of RM cell caused by other reasons such as errors in the physical layer should be taken care by the error control schemes employed in that specific layer.

3.3 Source end-system (SES) protocol

The SES keeps the following variables: *ACR*, *Old_ACR*, *BID*, and *Old_BID*. The behavior of the SES is as follows:

Initialization:

BID ← 0
Old_BID ← 0
ACR ← ICR
Old_ACR ← 0

If it's time to send an FRM cell:

Create an FRM cell

// Initialize an FRM cell
RM.Hop_Count ← 0
RM.BID ← BID
RM.Old_BID ← Old_BID
RM.ACR ← ACR
RM.Old_ACR ← Old_ACR
RM.ER ← PCR

// Update SES variables
Old_BID ← RM.BID
Old_ACR ← RM.ACR

Send an FRM cell

When a BRM cell arrives at the SES:

BID ← RM.BID
if (RM.ER < MCR) BID ← –3
ACR ← Max (MCR, Min (RM.ER, PCR))

If a new VC is created at the SES:

Create an FRM cell

// Initialize an FRM cell
RM.Hop_Count ← 0; RM.BID ← –2
RM.ACR ← ACR; RM.ER ← PCR

Send an FRM cell

If the VC is terminated:

Create an FRM cell

// Initialize an FRM cell
RM.BID ← –1
RM.Old_BID ← Old_BID
RM.Old_ACR ← Old_ACR

Send an FRM cell

When a VC is created, the VC starts transmitting cells at its initial cell rate *ICR*, which is assumed to be smaller than any advertised rate at the switch crossed by the VC. *BID* value of –2 indicates that this VC is a new VC and *BID* of –1 means that this VC is being terminated. In MR-ASAP, the original BID scheme is slightly modified such that upon receiving a BRM cell, if the *RM.ER* value from the BRM cell is less than *MCR*, *BID* is set to –3 meaning that this VC is MCR-constrained.

3.4 Destination end-system (DES) protocol

Upon receiving an FRM cell, the DES returns a BRM cell after initializing *RM.BID* to 0 and copying the values of *RM.ER* and *RM.Hop_Count* from the FRM cell into the corresponding fields in the BRM cell.

3.5 Switch protocol

The switch protocol operates at each port to an outgoing link of a switch. The variables kept at the switch are listed in Table 2.

Table 2. Variables at the switch (MR-ASAP)

Variable	Meaning
C	Link capacity for available bit rate (ABR) traffic
Fc	Total input rate from constrained VC's
Fm	Total input rate from MCR-constrained VC's
N	Number of VC's

Nc	Number of constrained VC's
Nm	Number of MCR-constrained VC's
R	Advertised rate

Note that *C* is periodically updated at the switch as follows:

C ← *Total link capacity – Total constant or variable bit rate input flows*.

Therefore, when an FRM cell arrives, the switch updates *R* based on the most up-to-date value of *C*. Initially, *Fc*, *Fm*, *Nc,* and *Nm* are all set to 0.

Upon receiving an FRM cell, the switch does the following:

```
RM.Hop_Count ← RM.Hop_Count + 1
if (RM.BID = –2)                        // this VC is new
        N ← N + 1; Nc ← Nc + 1
        Fc ← Fc + RM.ACR
        R ← C / N                       // fair share (Step A)
else
        Update_N_Fc_Fm_Nc_Nm()
        Update_R()
```

Upon receiving a BRM cell, the switch does the following:

```
if (RM.ER ≥ R) RM.ER ← R; RM.BID ← RM.Hop_Count
RM.Hop_Count ← RM.Hop_Count – 1
```

When a new VC joins at a switch, the protocol resets *R* to the fair share (Step A), which turns out to be simple and yet effectively gets rid of potential overloading at the link due to the new VC. If the FRM cell is received from any existing VC, the variables *N*, *Fc*, *Fm*, *Nc,* and *Nm* are updated according to the change in the VC's state and *R* is in turn updated. Figure 1 depicts the possible states of any existing VC and possible transitions among those states. The state "Constrained" refers to both states "bottleneck constrained" and "PCR-constrained".

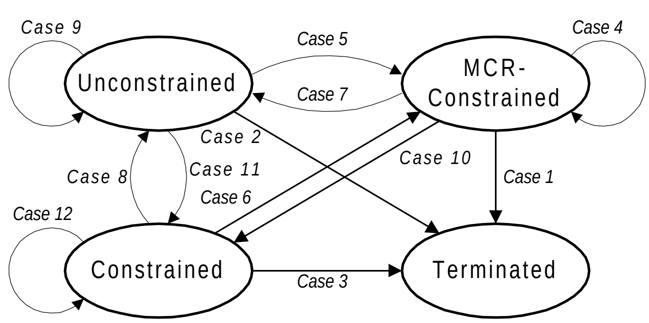


Figure 1. VC state changes

The function *Update_N_Fc_Fm_Nc_Nm()* updates *N*, *Fc*, *Fm*, *Nc,* and *Nm* as follows:

```
if (RM.BID = –1)  // VC is terminated
        if (RM.Old_BID = –3)                          // (Case 1)
                N ← N – 1; Nm ← Nm – 1
```

```
                Fm ← Fm – RM.Old_ACR
        else if (RM.Old_BID = RM.Hop_Count)            // (Case 2)
                N ← N – 1
        else                                            // (Case 3)
                N ← N – 1; Nc ← Nc – 1
                Fc ← Fc – RM.Old_ACR
else if (RM.BID = –3)  // VC becomes MCR-constrained
        if (RM.Old_BID = –3)                            // (Case 4)
                Fm ← Fm + (RM.ACR – RM.Old_ACR)
        else if (RM.Old_BID = RM.Hop_Count)            // (Case 5)
                Nm ← Nm + 1; Fm ← Fm + RM.ACR
        else                                            // (Case 6)
                Nm ← Nm + 1; Fm ← Fm + RM.ACR
                Nc ← Nc – 1; Fc ← Fc – RM.Old_ACR
else if (RM.BID = RM.Hop_Count)  // VC becomes unconstrained
        if (RM.Old_BID = –3)                            // (Case 7)
                Nm ← Nm – 1; Fm ← Fm – RM.Old_ACR
        else if (RM.Old_BID ≠ RM.Hop_Count)            // (Case 8)
                Nc ← Nc – 1; Fc ← Fc – RM.Old_ACR
        else                                            // (Case 9)
                no updates
else  // VC becomes constrained
        if (RM.Old_BID = –3)                            // (Case 10)
                Nm ← Nm – 1; Fm ← Fm – RM.Old_ACR
                Nc ← Nc + 1; Fc ← Fc + RM.ACR
        else if (RM.Old_BID = RM.Hop_Count)            // (Case 11)
                Nc ← Nc + 1; Fc ← Fc + RM.ACR
        else                                            // (Case 12)
                Fc ← Fc + (RM.ACR – RM.Old_ACR)
```

R is then updated in the function *Update_R()* as follows:

```
if (N – Nc – Nm ≠ 0)    // there exist unconstrained VC's
        R ← (C – Fc – Fm) / (N – Nc – Nm)                          // (Step B)
else                    // there exist NO unconstrained VC's
        if (Nm ≠ 0)     // there exist BOTH constrained VC's and MCR-constrained VC's
                if  (Fm + Fc > C)
                        R ← Min (Fc / Nc, C / N, (C – Fm) / (N – Nm))   // (Step C)
                else if  (Fm + Fc < C)
                        R ← Fm / Nm                                 // (Step D)
        else            // there exist ONLY constrained VC's
                if (Fc > C) R ← C / N                               // (Step E)
```

When an FRM cell arrives from any existing (not new) VC, if the number of unconstrained VC's ($Nu = N - Nc - Nm$) is non-zero, *R* is updated by dividing the remaining bandwidth ($= C - Fc - Fm$) by *Nu* (Step B). If *Nu* is zero, there are two cases to consider: (a) $Nm \neq 0$, and (b) $Nm = 0$. In case (a), there exist both constrained and MCR-constrained VC's, while, in case (b), there exist only constrained VC's. It is easy to prove that it is not possible to have only MCR-constrained VC's by Assumption 1.

In the former case (a), there are three sub-cases to consider: (a-1) $Fm + Fc > C$, (a-2) $Fm + Fc < C$, and (a-3) $Fm + Fc = C$. The first sub-case (a-1) implies that some of the constrained VC's are transmitting at the rates higher than their corresponding GMM rates. In this case, by setting R to the minimum among Fc / Nc (i.e., average flow from constrained VC's), C / N, and $(C - Fm) / (N - Nm)$ at Step C, it is guaranteed that after one round trip delay, at lease one constrained VC becomes unconstrained and reduces its rate eventually until R reaches its GMM rate. Note that at Step C, Nc and $(N - Nm)$ are non-zero.

The second sub-case (a-2) implies that some of the MCR-constrained VC's are transmitting at the rates lower than their corresponding GMM optimal rates. In this case, by setting R to Fm / Nm (i.e., average input flow from MCR-constrained VC's) at Step D, it is guaranteed that after one round trip delay, at lease one MCR-constrained VC becomes unconstrained and can increase its rate eventually until R reaches its GMM rate. In the third sub-case (a-3), R is not updated.

When there exist only constrained VC's as in case (b), R is not updated except for the case when C drops significantly such that C becomes smaller then Fc. In that case, R is simply reset to the fair share (Step E), which again turns out to be the most reasonable choice. Consider a switch where $N = Nc$ and now C drops smaller than Fc. Based upon the new C, the switch is assumed to be a new bottleneck switch. However, if R does not change, then BRM cells fail to inform the sources of the change in the bottleneck switch. By resetting R to the fair share, the ID of the new bottleneck switch can be forwarded to the sources of all the VC's. An example illustrating this situation is found in [Tsa99].

MR-ASAP explicitly handles various disturbances in the network such as changes in the available bandwidth for ABR traffic, VC creation and VC termination (tear-down). Note that the switch's processing time incurred upon receiving an RM cell is constant (independent of the number of VC's). It should be also noted that since the switch does not keep any per-VC information, MR-ASAP is completely scalable.

Finally, a code segment for *re-marking* [Cha99] is included as an option in the switch protocol:

```
// if the VC is constrained but the ACR is larger than new R
if (RM.BID ≠ RM.Hop_Count) and (RM.BID ≠ -1) and (RM.BID ≠ -3) and (RM.ACR ≥ R)
        R_Temp ← (C – Fc – Fm + RM.ACR) / (N – Nc – Nm + 1)
        if ACR ≥ R_Temp, then RM.BID = RM.Hop_Count; RM.ER = R_Temp
```

This optional code segment is executed as the last step upon receiving an FRM cell. The code segment improves the performance and is useful for the convergence proof. The advertised rate R that is computed before the re-marking step is called the first-round rate, and the second computation of the advertised rate *R_Temp* is called the re-marking rate.

3.6 Handling non-persistent sources

Due to the nature of ABR traffic, the behavior of the source is often non-persistent and bursty. The MR-ASAP can be modified to handle the non-persistent source behavior as follows: At the SES, the actual data rate is measured and stored into an additional variable called *Actual_Rate*. When the SES sends out an FRM cell, if *Actual_Rate* is less than *ACR*, in other words, if the SES does not fully utilize its *ACR*, the SES copies *Actual_Rate* instead of *ACR* into *RM.ACR*, and copies 0 instead of *BID* into *RM.BID*. Other parts of the protocol remain unchanged. By copying 0 into *RM.BID*, the SES declares to the switches on its path that the SES cannot fully utilize the rates advertised by those switches. It is assumed that *Actual_Rate* is no smaller than the *MCR* at the SES.

Upon receiving the FRM cell from the SES, each switch reallocates the unused portion (= *ACR* – *Actual_Rate*) of the VC's *ACR* to other VC's. Note that changes in *BID* and/or *ACR* of the VC will cause the switch variables to change, and in turn *R* to be updated. It should be noted that the variables *ACR* and *BID* at the SES keep being updated according to the values of *ER* and *BID* fields from the BRM cell.

Later, if *Actual_Rate* becomes greater than or equal to the *ACR*, in other words, if the SES has enough cells to fill up its *ACR*, the SES copies *ACR* into *RM.ACR* and copies *BID* into *RM.BID* next time when the SES sends out an FRM cell. Thus, upon receiving the FRM cell from this VC, each switch again recalculates its advertised rate using the new values from the FRM cell. This scheme does not introduce any additional overhead except a simple routine at the SES, which measures the actual transmission rate and does simple arithmetic operation.

3.7 Examples

Let's consider a simple network configuration as depicted in Figure 2. Table 2 specifies the values of PCR and MCR for each VC used in Example 1. Initially, Switch 1 sets *R* to the fair share (= 37.5). After one round trip delay, since the PCR of VC 1 is less than *R*, VC 1 is regarded constrained at Switch 1. Thus both VC's 3 and 4 become MCR-constrained since their MCR's are greater than *R*. Therefore, at Switch 1, $Nc = 1$, $Fc = 15$, $Nm = 2$, $Fm = 110$. *R* is in turn calculated as (150 – 15 – 110) / (4 – 1 – 2) = 25. Then, the VC 2 will transmit its data at *R* and remain as an unconstrained VC. The GMM ACR value for each VC is also shown in Table 3.

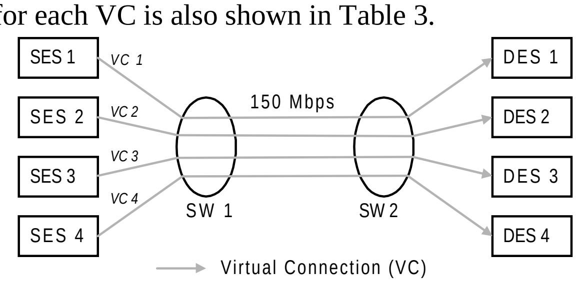


Figure 2. Sample configuration

Table 3. PCR, MCR, and GMM ACR of each VC (Example 1)

VC #	VC 1	VC 2	VC 3	VC 4
PCR	15	150	150	150
MCR	0	0	50	60
ACR	15	25	50	60

Table 4. PCR, MCR, and GMM ACR of each VC (Example 2)

VC #	VC 1	VC 2	VC 3	VC 4
PCR	15	25	150	150
MCR	0	0	50	60
ACR	15	25	50	60

Table 5. PCR, MCR, and GMM ACR of each VC (Example 3)

VC #	VC 1	VC 2	VC 3	VC 4
PCR	15	25	150	150
MCR	0	0	30	70
ACR	15	25	40	70

In the next example (Example 2) shown in Table 4, the PCR for VC 1 becomes 25. Switch 1 again initializes R to the fair share (= 37.5). After one round trip delay, since the PCR's of both VC's 1 and 2 are less than R, both VC's are regarded as constrained at Switch 1. Thus both VC's 3 and 4 become MCR-constrained since their MCR's are greater than R. Therefore, at Switch 1, $Nc = 2$, $Fc = 40$, $Nm = 2$, $Fm = 110$. Now, since $N - Nc - Nm = 0$ and $Nm \neq 0$ and $Fc + Fm = C$, R remains unchanged from 37.5.

In Example 3 as shown in Table 5, the MCR's for VC 3 and VC 4 are changed to 30 and 70, respectively. After one round trip delay from initialization of R, both VC's 1 and 2 are regarded constrained at Switch 1 since their PCR's are less than R. While, VC 4 becomes MCR-constrained. Thus, at Switch 1, $Nc = 2$, $Fc = 40$, $Nm = 1$, $Fm = 70$. Now, $R = (150 - 40 - 70) / (4 - 2 - 1) = 40$. At this new R, VC 3 will transmit data and will remain unconstrained.

4. Simulation Results

In this section, we perform simulations to demonstrate the protocol MR-ASAP converges fast to the GMM optimality after the network is disturbed: dynamic changes in the available bandwidth for ABR traffic, and/or dynamic arrival/departure of VC's. The protocol is simulated using NIST's ATM Network Simulator [Gol95]. Figure 3 shows a metropolitan area network (MAN) configuration used in the simulation, which is simple and yet effectively shows the generalized maxmin (GMM) fairness. The distances of Link 1 and Link 2 are 20 km each and the distances of all other links are 0.2 km each. The actual capacities of Link 1 and Link 2 are 155 Mbps.

First, we simulate the MR-ASAP under dynamic changes in the available bandwidth for ABR traffic. The simulation ran for 20 msec, which is long enough to show the convergence in this MAN configuration. In order to simulate dynamic changes in available bandwidth for ABR traffic, a VBR VC is added at Link 2, which transmit cells at 55 Mbps for the intervals of 5 to 10 msec and 15 to 20 msec. Due to the VBR traffic, the available bandwidth for ABR traffic effectively varies between 155 Mbps and 100 mbps.

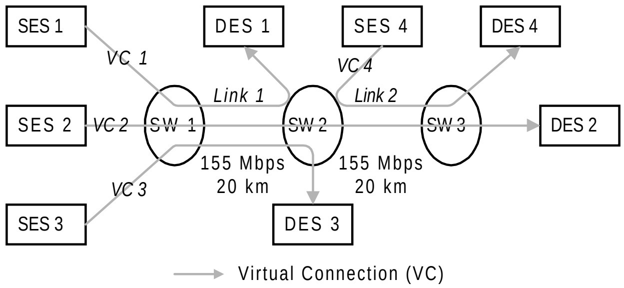


Figure 3. Simulation configuration

Table 6. VC parameters

VC #	VC 1	VC 2	VC 3	VC 4
PCR	30	155	155	155
MCR	0.155	0.155	70	80

Table 7. GMM optimal rate (ACR) for each VC

VBR VC	VC 1	VC 2	VC 3	VC 4
Idle	30	55	70	100
Active	30	20	105	80

Table 6 lists the PCR constrain and MCR requirement of each VC. The SES transmits an RM cell every 100 μsec and the time interval used to measure the input flow from the VBR traffic is 500 μsec.

The GMM optimal rates for each VC are shown in Table 7. Figure 4 shows the changes in ACR values under background VBR traffic changes. Upon changes in the background traffic, each VC converges to its GMM rate within 1.7 msec. However, since it takes 0.5 msec for the switch to detect the background traffic change, the actual convergence time is less than 1.2 msec. Note that the little oscillation occurred after 16 msec for both VC 2 and VC 3 is caused by the VBR input traffic measurement error.

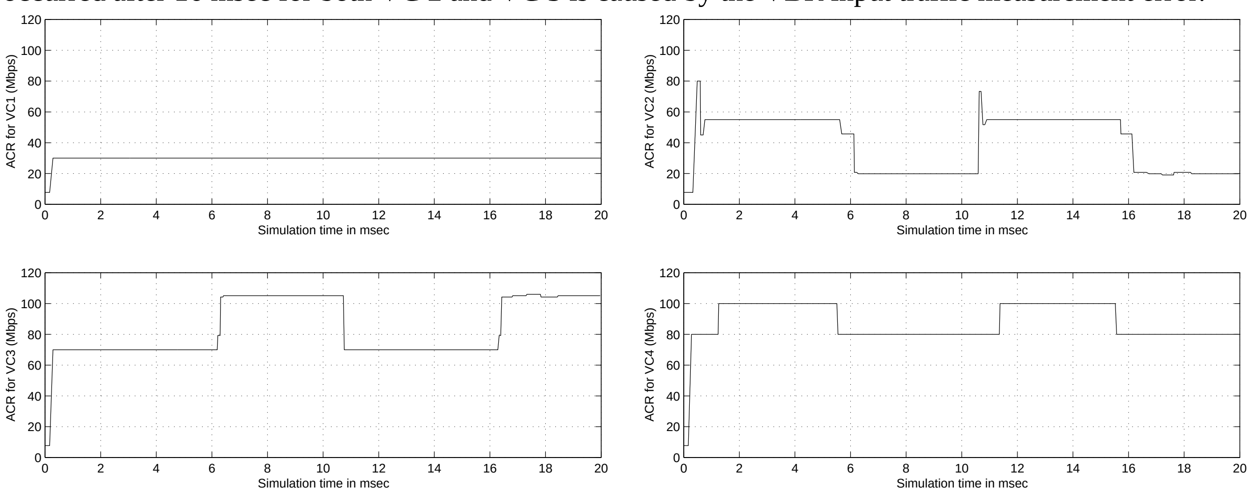
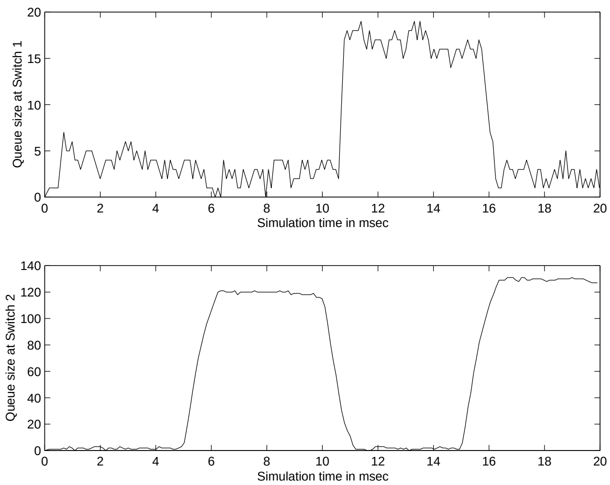


Figure 4. Changes in ACR for each VC under dynamic background traffic

Figure 5 shows the stable queue sizes at Switch 1 and Switch 2, and Figure 6 demonstrates excellent link utilization at both switches.



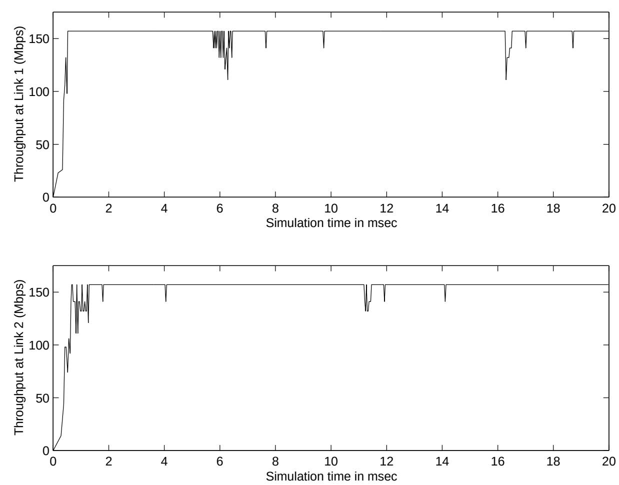


Figure 5. Queue sizes at Switch 1 and Switch 2

Figure 6. Throughputs at Link 1 and Link 2

Second, MR-ASAP is simulated under dynamic arrival and departure of VC's. In this simulation, each VC transmits cell according to the schedule as shown in Table 8. Table 9 shows the GMM optimal rate for each VC for each interval. Figure 7 shows the changes in ACR values. Note that every VC converges to its GMM optimal rate upon arrival and departure of other VC within less than 1.5 msec. Again, by subtracting the VBR measurement interval 0.5 msec, the protocol converges within 1 msec.

Table 8. Arrival time and departure time of each VC (msec)

	VC 1	VC 2	VC 3	VC 4
Starts at	10	5	0	0
Stops at	15	20	30	30

Table 9. GMM optimal rate (ACR) for each VC

Interval	VC 1	VC 2	VC 3	VC 4
0 ~ 5 msec	x	x	155	155
5 ~ 10 msec	x	75	80	80
10 ~ 15 msec	30	55	70	100
15 ~20 msec	x	75	80	80
20 msec ~	x	x	155	155

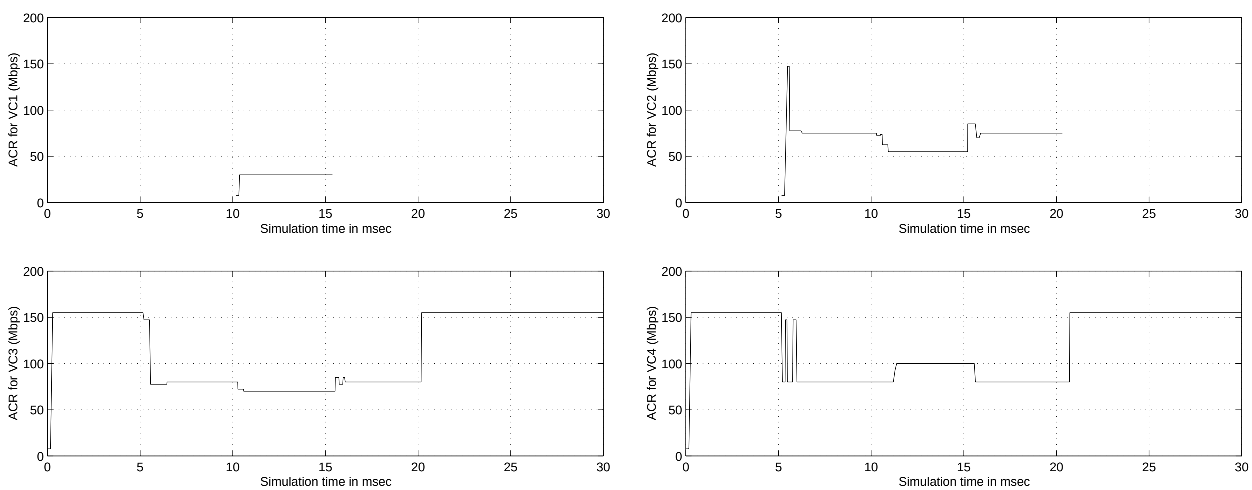


Figure 7. Changes in ACR for each VC under dynamic VC arrival and departure

5. CONVERGENCE PROOF OF MR-ASAP

The protocol MR-ASAP can be shown to converge to the GMM optimal rate assignment. However, due to space limitation, it is impossible to deal with all the possible boundary conditions in this paper. GMM optimality is really the original maxmin optimality with additional boundary conditions. The boundary conditions do not add any major content to the proof, while they make the proof unnecessarily complex. The boundary conditions are useful basically for checking to see if the converged rate vector is truly GMM optimal. Thus we shall in this section prove only the convergence of ASAP. With the proofs provided here one can construct a proof for MR-ASAP.

With the proofs presented here, ASAP becomes the third exact maxmin protocol that has a global convergence proof; the first protocol with a proof is that of Charny et al. [Cha95] and its extension by Hou [Hou97], the second is the CPG protocol [Tsa99]. The proof for ASAP is much more difficult than the first two as ASAP does not require per-flow accounting while the other protocols use per-flow information, which greatly simplifies their proofs. In order to prove the convergence, we have to make some synchronization assumptions. These assumptions do not mean that the protocol must be adapted to fit these synchronization assumptions; rather, the proof with the synchronization assumption gives us an excellent justification that the truly asynchronous version will converge. We shall call an advertised rate computation as a link update.

In the proofs that follow, the notation of [Tsa99] is used. The proof will use the concept of CPG (Constraint Precedence Graph) which was introduced in [Tsa99]. If one does not have access to the reference [Tsa99], one can interpret a level l AR/link as the l-th AR/link which converges to its maxmin value in the idealized parallel version of the maxmin protocol. The proof is self-contained except this notion of CPG level is needed. Let L be the number of levels in the CPG which can be interpreted to be the number of distinct rates in the maxmin assignment, under a simplistic assumption. Due to space limitation we shall not elaborate any more on the concept of CPG.

Assumption 2 (Synchronization Assumption). We assume that minimization of ER is done in the forward path and both advertised rate update and re-marking are done in the backward path.

Definition. The first round-trip update after t at link k is defined to be the first possible computation of the advertised rate at link k, after the event that every VC crossing link k has at least one BRM cell arriving after t, and this BRM cell has caused a new BRM cell to return to link k.

Definition. Let t_k^0 be the time at which the first link update is computed at link k after the disturbance. Let $t^0 = \max\{t_k^0 : k \in L\}$; thus, t^0 is the time at which the first link updates have been computed at all links in the network. Furthermore, let t_k^1 be the time at which the first round-trip update after t^0 is computed at link k. Let $t^1 = \max\{t_k^1 : k \in L\}$; thus, t^1 is the time at which the first update after t^0 is computed at all links in the network. Furthermore, let $\hat{t}_k^l$ denote the first round-trip update time for link k after t^{l-1}. Let $\hat{t}^l = \max\{\hat{t}_k^l : k \in L\}$. Define t_k^l denote the first round-trip update time for link k after $\hat{t}^{l-1}$and $t^l = \max\{t_k^l : k \in L\}$. Intuitively, t^l is the second round-trip update after t^{l-1}.

Definition. For every link k, let $C_k^1 = C_k$ and $N_k^1 = N_k$. For every link k, and every level l, let $C_k^l = C_k^{l-1} - Fc_k^l$ and $N_k^l = N_k^{l-1} - Nc_k^l$. For every link k, for every level l, the rate $R_k^l = C_k^l / N_k^l$ is called the remaining fair share at level l. Note that R_k^1 is commonly known as the fair share at link k. For any link k, the average constrained flow at level l is defined to be Fc_k / Nc_k where (Fc_k, Nc_k) corresponds to the constrained flows which are at least level l or higher.

Lemma 1. For all link k, if all the denominators in the following expressions are positive,
(a) $C_k/N_k = Fc_k/Nc_k$ iff $(C_k - Fc_k)/(N_k - Nc_k) = C_k/N_k$;$C_k/N_k > Fc_k/Nc_k$ (b) $C_k/N_k < Fc_k/Nc_k$ iff $(C_k - Fc_k)/(N_k - Nc_k) > C_k/N_k$;t_k^1(c) $C_k/N_k > Fc_k/Nc_k$ iff $(C_k - Fc_k)/(N_k - Nc_k) < C_k/N_k$.

Proof. The proof is trivial and is omitted.

Lemma 2.

(a) For all links j, $R_j(t) \geq R_j^1$, for all $t \geq t^0$.

(b) For all k at the CPG level 1 and for all $t \geq t^1$, $R_k(t) = R_k^1$.

Proof. Part (a): The proof is trivial since by time t^0, every link has performed at least one advertised rate update and the ASAP update equation forces the advertised rate to be no smaller than the fair share.

Part (b): Consider first the update at time t_k^1 for link k. There are two sub-cases to consider.

Case (b1): $Nc_k(t_k^1) = 0$. Then $R_k(t_k^1) = R_k^1 = C_k/N_k$.

Case (b2): $Nc_k(t_k^1) \neq 0$. By part (a), after all the link advertised rates have been relayed to all the sources, all VC's must have a flow no smaller than C_k/N_k, which is the smallest of the fair shares among all the links, thus the average constrained flows must be at least as large as C_k/N_k, i.e. for all $t \geq t_k^1$ $Fc_k(t)/Nc_k(t) \geq C_k/N_k$. By Lemma 1 part (a) and (b), and the ASAP update equation, $R_k(t_k^1) = R_k^1$. Now, at every subsequent update at link k, the same argument continues to apply. ■

Suppose that at time t there is an update at link j, then let $R_j(t)$ denote the first-round advertised rate and let $R_j(t^+)$ denote the re-marking advertised rate. A departing BRM cell after the update and re-marking have been completed at a link will be called a DBRM cell.

Lemma 3. Under synchronization assumption 1, suppose at time $t \geq t_j^l$ link j receives a BRM cell from VC i and after the first-round update,

$$Fc_j(t)/Nc_j(t) \leq C_j^{l+1} / N_j^{l+1}, \tag{1}$$

for some level l. Then after re-marking, one of the following is true:

(a) VC i is marked as unconstrained in the BRM cell and it remains unconstrained in the DBRM cell;

(b) VC i is marked as unconstrained in the BRM cell and it is marked as constrained in the DBRM cell and

$$[Fc_j(t)+A_i]/[Nc_j(t)+1] \le C_j^{l+1} / N_j^{l+1};$$

(c) VC i is marked as constrained in the BRM cell and it remains constrained in the DBRM cell;

(d) VC i is marked as constrained in the BRM cell and it is marked as unconstrained in the DBRM cell and

$$[Fc_j(t)-A_i]/[Nc_j(t)-1] \le C_j^{l+1} / N_j^{l+1}.$$

Proof. From Lemma 1,

$$R_j(t) \ge C_j^{l+1} / N_j^{l+1}. \tag{2}$$

There is nothing to prove in Case (a) and (c).

Case (b): After the first-round update, VC i must be marked as constrained and it remains constrained after the re-marking. Thus $A_i < R_j(t)$ and $A_i < R_j(t^+)$. In the re-marking process, the average constrained flow is updated to $[Fc_j(t)+A_i]/[Nc_j(t)+1]$. Now there are two sub-cases:

(b1) $[Fc_j(t)+A_i]/[Nc_j(t)+1] \le C_j^{l+1} / N_j^{l+1}$, which is what we need to prove.

(b2) $[Fc_j(t)+A_i]/[Nc_j(t)+1] > C_j^{l+1} / N_j^{l+1}$. In this case, by Lemma 1, $R_j(t^+) < C_j^{l+1} / N_j^{l+1}$. The fact that VC i remains constrained after re-marking implies $A_i < C_j^{l+1} / N_j^{l+1}$ which in conjunction with (1) implies

$$[Fc_j(t)+A_i]/[Nc_j(t)+1] < C_j^{l+1} / N_j^{l+1},$$

contradicting the assumption of (b2). Thus case (b2) is impossible to occur.

Case (d): There will be two possibilities here:

Case (d1) After the first round, VC i is marked as unconstrained and remains unconstrained after the re-marking: $A_i \ge R_j(t)$ and $A_i \ge R_j(t^+)$. Now by (2), $A_i \ge C_j^{l+1} / N_j^{l+1}$. Now this fact together with (1) implies that $[Fc_j(t)-A_i]/[Nc_j(t)-1] < C_j^{l+1} / N_j^{l+1}$.

Case (d2) After the first-round update, VC i is marked as constrained and is re-marked as unconstrained: In this case, $R_j(t^+) = R_j(t)$ and it is impossible for VC i to be re-marked as unconstrained. Therefore this case is impossible to occur. ■

Lemma 3 implies that if condition (1) is satisfied, then after one RTT (round trip time), when the feedback from the source returns, condition (1) remains true, everything else being the same. Lemma 4 (for which the proof is skipped as it is trivial) shows that if condition (1) is not satisfied, the feedback from the source will drive the average constrained flow toward satisfying condition (1).

Lemma 4. Suppose at time $t \geq t_j^l$ link j receives a BRM cell from VC i and after the first-round update,

$$Fc_j(t)/Nc_j(t) > C_j^{l+1}/N_j^{l+1}, \quad (1a)$$

for some level l. Then if $A_i > C_j^{l+1}/N_j^{l+1}$, VC i will be marked as unconstrained in the DBRM cell.

Lemma 4 implies the following: If the average constrained flow is greater than the remaining fair share, the average constrained flow in the future (after one RTT) will be reduced.

Assumption 3 (Second Synchronization Assumption). We assume that each VC has only one RM cell circulating in the path at any moment in time. The following two lemmas requires this assumption.

Lemma 5. For all level $l \geq 2$ and all links j, $R_j(t) \geq R_j^l$, for all $t \geq \hat{t}^l$.

Lemma 6. For all level $l \geq 2$ and all links k at CPG level l and for all $t \geq t^l$, $R_k(t) = R_k^l$.

The proof for Lemma 6 is very similar to that of Lemma 2 Part (b) and is very straightforward to derive. We shall skip the proof for Lemma 6.

Proof of Lemma 5. Consider an arbitrary link j and let $x_j(t) = Fc_j(t)/Nc_j(t) - C_j^{l+1}/N_j^{l+1}$ denote the deviation from the remaining fair share C_j^{l+1}/N_j^{l+1} at level l. For simplicity, let us drop the subscript j. Without loss of generality, we shall adopt a discrete-time notation to describe the dynamic of the average constrained flow:

$$\{x\!: x \leq 0\}, \quad (3)$$

where d is the RTT, $x(t+d)$ denotes the average constrained flow at time $t+d$ (which is one RTT into the future from time t), $x(t+d-1)$ denotes the average constrained flow at time $t+d-1$, which is interpreted to be the time instant right before the average constrained flow changes to the value $x(t+d)$, and $\alpha(t)$ is a scalar time-varying quantity to be specified below. By Lemma 3 and Lemma 4, we have:

$$\alpha(t)x(t) = \begin{cases} < 0, & \text{if } x(t) > 0; \\ \leq 0, & \text{if } x(t) \leq 0. \end{cases} \quad (4)$$

From equation (4), it is clear that the dynamic system described by (3) is stable in the following sense: After an initial transient, the system will enter a steady-state region described by $\{x\!: x \leq 0\}$, or the region where the average constrained flow is no larger than the remaining fair share. To see this, one can solve for the trajectory of $x(t)$ by direct substitution:

$$x(t+d+m) = x(t+d-1) + \sum_{i=0}^{m} \alpha(t+i)x(t+i) .$$

Now by the second synchronization assumption and equation (4), we must conclude that after one RTT $x(t)$ will enter the desired steady-state region $\{x\!: x \leq 0\}$. ■

With these Lemmas, a formal theorem can be stated as follows:

Theorem 1. The protocol ASAP (MR-ASAP) converges to the maxmin (or GMM) optimal rate assignment within 2RTTL, where RTT is the maximum round trip delay in the network.

6. CONCLUSIONS

This paper presents the first non-per-VC accounting (scalable) exact maxmin flow control protocol that guarantees the minimum rate requirements. This protocol is readily applicable to ATM, Frame Relay and other circuit-switched networks. For IP networks, adaptation is needed as paths can change and the sources do not have to go through call set-up and tear-down as in a circuit-switched network. However, as long as the speed of path change is a lot slower than the speed of convergence of the protocol, MR-ASAP can be adapted to work in the IP environment. Work is in progress in this direction.

The maxmin fairness doctrine is most suitable for best-effort traffic but may not be suitable for truly QoS-guaranteed traffic where users negotiate a contract containing specific rate requirements and other traffic parameters. But for the traffic that does not require hard QoS satisfaction except for the minimum rate, MR-ASAP is very applicable. For example, MR-ASAP can potentially be adapted to the differentiated service network where the QoS sessions are required to inform the network of their worst-case requirements (in terms of minimum rates and more).

REFERENCES

[Aru96] A. Arulambalam, X. Chen, and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM Networks," IEEE Communications Magazine, November 1996, pp.92-100.

[ATM96] The ATM Forum. 'The ATM Forum Traffic Management Specification,' v.4.0, 1996.

[Cha95] A. Charny, D. Clark, and R. Jain, "Congestion Control with Explicit Rate Indication," Proc. ICC'95, June 1995, pp.1954-1963.

[Gol95] N. Golmie, A. Koenig, and D. Su, 'The NIST ATM Network Simulator Operation and Programming,' NISTIR 5703, Version 1.0, August 1995.

[Hay81] H. Hayden, "Voice Flow Control in Integrated Packet Networks," Report LIDS-TH-1152, Cambridge, MA: MIT Laboratory for Information and Decision Systems, 1981.

[Hou98] Y. Hou, H. Tzeng, and S. Panwar, "A Generalized Max-Min Rate Allocation Policy and Its Distributed Implementation Using the ABR Flow Control Mechanism," Proc. IEEE Infocom '98, San Fransisco, April 1998.

[Jaf81] J. Jaffe, "Bottleneck Flow Control," IEEE Trans. on Communications, July 1981, pp.954-961.

[Jai96] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan, "ERICA Switch Algorithm: A Complete Description," ATM Forum cont. 96-1172, August 1996.

[Kal95] L. Kalampoukas, A. Varma, and K.K. Ramakrishnan, "An Efficient Rate Allocation Algorithm for ATM Networks Providing Maxmin Fairness," Proc. 6th IFIP Int'l. Conf. High Performance Networking, September 1995, pp.143-154.

[Kal97] L. Kalampoukas, 'Congestion Management in High Speed Networks,' Ph.D Dissertation, University of California at Santa Cruz, August 1997.

[Kim99] Y. Kim, 'A Non-Per-VC Accounting Max-Min Protocol for ABR Flow Control (ASAP) and its Multiple-Time-Scale Extension,' Ph.D Dissertation, University of California, Irvine, January 1999.

[Tsa98a] W. Tsai, Y. Kim, and L. Hu, "ASAP: A Non-Per-VC Accounting Max-Min Protocol for ABR Flow Control with Optimal Convergence Speed," Proc. IEEE SICON '98, Singapore, July 1998, pp.139-153. http://www.eng.uci.edu/~netrol.

[Tsa98b] W. Tsai, Y. Kim, and C. Toh, "A Stability and Sensitivity Theory for Rate-Based Max-Min Flow Control for ABR Service," Proc. IEEE SICON '98, Singapore, July 1998, pp.139-153. http://www.eng.uci.edu/~netrol.

[Tsa99] W. Tsai and Y. Kim, "Re-Examining Max-Min Protocols: A Fundamental Study on Convergence, Complexity, Variations, and Performance," to appear in Proc. IEEE Infocom '99, New York, March 1999. http://www.eng.uci.edu/~netrol.