

Finite State Asynchronous Resource Renegotiation Scheme for MPEG Traffic: R++

Sahinoglu, Z.; Porikli, F.; Vetro, A.

TR2003-56 July 2003

Abstract

This paper presents a finite state resource management scheme, so called R++, for dynamically allocating bandwidth for variable bit-rate(VBR)traffic in a network or component of the network that supports resource renegotiations (e.g. ATM,RSVP etc.). The introduced scheme does not assume any a-priori knowledge of traffic, and it uses multiple bandwidth decision units in a hierarchy to eliminate large fluctuations in allocated bandwidth. The performance is evaluated on different MPEG-1 coded traces. Simulation experiments show that the new approach achieves better link utilization and lower 0.99-quantile queue sizes after less number of renegotiations than other methods in the literature.

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Publication History:

1. First printing, TR-2003-56, July 2003



Finite State Asynchronous Resource Renegotiation Scheme for MPEG Traffic: R++

Zafer Sahinoglu, F. Porikli, Anthony Vetro
Mitsubishi Electric Research Labs, Murray Hill, NJ
zafer@merl.com, avetro@merl.com

Abstract—This paper presents a finite state resource management scheme, so called R++, for dynamically allocating bandwidth for variable bit-rate (VBR) traffic in a network or component of the network that supports resource renegotiations (e.g. ATM, RSVP etc.). The introduced scheme does not assume any a-priori knowledge of traffic, and it uses multiple bandwidth decision units in a hierarchy to eliminate large fluctuations in allocated bandwidth. The performance is evaluated on different MPEG-1 coded traces. Simulation experiments show that the new approach achieves better link utilization and lower 0.99-quantile queue sizes after less number of renegotiations than other methods in the literature.
Index terms: Renegotiation, VBR, QoS, bandwidth.

I. INTRODUCTION

Quality of Service (QoS) provisioning for VBR traffic is achievable with stringent packet/cell loss rate and delay constraints, but at the expense of low network utilization. This paper presents an online renegotiation-based dynamic bandwidth allocation scheme to increase the network utilization, by achieving minimum number of bandwidth reallocations under given buffering, under-utilization and renegotiation cost constraints.

Assume a network that consists of a pair of source and destination nodes and RSVP routers lined between them as in Fig.1. RSVP routers support receiver driven resource reservation by conveying a reservation request, which originates at the destination, to the source. As long as the end-to-end application QoS (e.g., delay, bit rate etc.) is provisioned once a reservation is made, any two consecutive RSVP routers can renegotiate their service rates with each other. In the example that Fig.1 shows, the two routers try to provision 100ms end-to-end delay constraint for the application. This can be achieved by adjusting service rates and accordingly queue occupancies. Such a resource negotiation paradigm helps to achieve both a more efficient buffer usage at each router and high link utilization. We, in this article, present a resource renegotiation scheme that supports this paradigm.

There are various bandwidth predictors and renegotiation methods available in the literature. However, either most them are designed for off-line systems, or they have high complexity and computational overload that are not proper for on-line

QoS provisioning. Off-line systems can determine the exact bandwidth characteristics of a stream a-priori. However, on-line algorithms are needed in many real time applications.

The main contribution of the paper is, under more realistic link utilization, buffering and bandwidth-renegotiation-signaling cost assumptions, a finite state bandwidth decision mechanism that controls bandwidth reallocations in step-wise increments/decrements based on monitoring of cost metrics. Cost metrics are monitored to decide when to reallocate resources. The multiple-state decision mechanism (with three predictors) is a solution to prevent bandwidth level changes from high jumps, and also to lower the impact of prediction errors on performance. Exploiting prediction results of multiple predictors increases the efficacy of bandwidth allocation (or network resources in general). Therefore, we use a wavelet filter based bandwidth predictor. Energies in wavelet filter sub-bands are used to compute how much to readjust resources. The rest of the paper is organized as follows: Section II summarizes other approaches in the literature. Section III presents the renegotiation control unit (with underlying realistic cost functions and interrupts), which decides when to reallocate bandwidth. Section IV gives a wavelet based traffic envelope detection approach. Section V explains the introduced finite state bandwidth reallocation mechanism (FSBRM) in R++ structure,

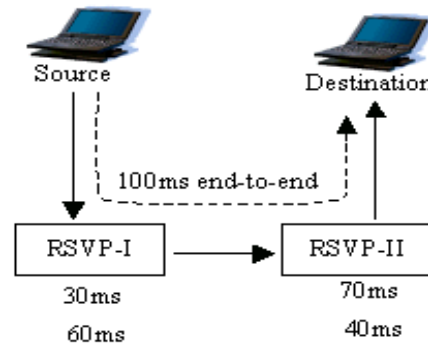


Figure 1 Illustration of a network in which two routers between the source and destination nodes negotiate their resources to provision 100ms end-to-end delay constraint to the application.

which decides how much bandwidth to reallocate at any renegotiation instant. Performance comparison of R++ with other online approaches is given in Section VI. The notations used are listed in Table I.

TABLE I
NOTATIONS

Notation	Definition
$X(n)$	# of bits received(counted) in time slot n
$q(n)$	Queue size in time slot n (bits)
$s(n)$	Under-utilization in time slot n (bits)
E_i	Signal energy in frequency sub-band i
$a(n)$	Bandwidth allocated during time slot n
$\bar{\rho}$	Average utilization
$\rho(n)$	Instantaneous utilization in time slot n
T_{\max}	Maximum renegotiation-cost (bits)
T_{\min}	Minimum renegotiation-cost (bits)
X_{dc}	Bandwidth for starvation prevention
α	Buffering cost coefficient
β	Under-utilization cost coefficient
r	Feasible renegotiation interval
$u(n)$	Under-utilization cost at the end of time slot n
$b(n)$	Buffering cost at the end of time slot n
CFI	Current Frame Index
LRI	Frame index at last renegotiation
IRI	Inter renegotiation interval
B	0.99 quantile queue size

II. ONLINE METHODS

Dynamic bandwidth allocation methods are split into two groups as synchronous and asynchronous. In synchronous methods, resources are modified periodically, at fixed time intervals, unlike the asynchronous in which the allocated resources to the traffic are updated on a need basis. Some of the well-known online methods in the literature are summarized below.

Synchronous approaches:

i) *Periodic Renegotiations* [1] Computes the average arrival rate within a given time interval periodically, and allocate ratio of that as the new bandwidth for the next time slot.

Asynchronous approaches:

i) *Renegotiated Constant Bit Rate (RCBR)* [2] Assumes constant cost per renegotiation and per allocated bandwidth. The method is based on an *AR* (l) bandwidth estimator and buffer thresholds. Three

parameters have to be tuned: a high and low buffer thresholds (B_h, B_l as introduced in the original text) and a time constant T .

ii) *Renegotiated Dynamic Bandwidth Allocation RDBA* [3] Takes queue size, underutilization and renegotiation costs into consideration. Time-variant renegotiation cost is first time defined in this work, but some of the assumptions related to renegotiation cost are not realistic.

iii) *Normalized Least Mean Square (NLMS)* [4] Bases the dynamic bandwidth allocation on the prediction of the next GOP rate, using a normalized LMS algorithm.

III. COST FUNCTIONS&INTERRUPTS

We consider three cost functions, and three interrupts generated due to temporal changes in defined cost metrics. These interrupts are used to switch from one decision state to another in the FSBRM as explained in the next section. Cost functions are *buffering* cost, *underutilization* and *renegotiation* costs. They are computed as follows:

$$b(n) = \alpha \max(0, q(n-1) + X(n) - a(n)) \quad (1)$$

$$u(n) = \beta \cdot s(n) \quad (2)$$

where $s(n) = \begin{cases} \min(0, s(n-1) + X(n) - a(n)), & q(n) = 0 \\ 0, & q(n) > 0 \end{cases}$

$$T(n) = \begin{cases} T_{\max}, & CFI - LRI < r \\ T_{\min}, & CFI - LRI > r \end{cases} \quad (3)$$

We assume that the completion of signaling for resource renegotiation between the two network components (e.g., routers) takes r time slots. Therefore, starting a new renegotiation just after the previous one (unless it is longer than r since the last renegotiation) is not feasible

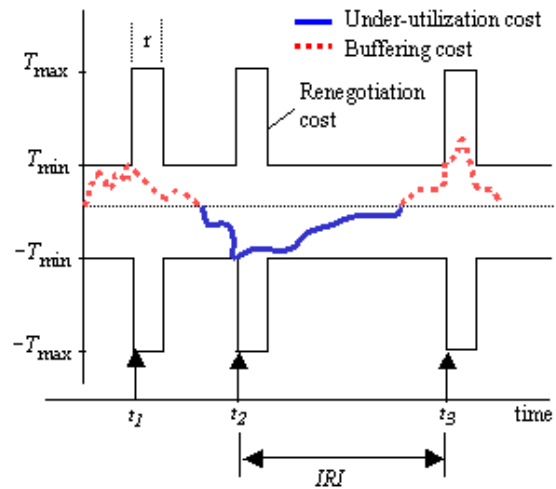


Figure 2 Illustration of the renegotiation instants according to R++ with corresponding cost curves vs. time. Interrupts t_1 and t_3 are due to buffering, and t_2 due to underutilization.

and the cost of renegotiation at that instant (T_{\max}) is very high. We also define a lower threshold for the renegotiation cost, because even though a renegotiation might be feasible, it would still induce a signaling load on the network. Whenever the under-utilization or buffering cost crosses the renegotiation cost boundary, an interrupt is created (Fig.2). Another interrupt is generated whenever the instantaneous utilization $\rho(n)$ is very low to maintain high utilization between the network and the receiver. In order to process this interrupt, the renegotiation cost must be at its lower limit. In the next two sections, explained is how much bandwidth to reallocate at each renegotiation instant.

IV. TRAFFIC ENVELOPE DETECTOR

We decompose time series traffic data, each element of which consists of bit arrival rate information, into different frequency bands. This method separates low and high frequency components in the arrival process. The energy distribution in each sub-band frequency informs us of the contribution of each sub-band energy component to the main traffic volume. This information is used as feedback for the prediction of traffic trend (e.g., scene changes). Assume a vector $\underline{X}_k = [X(n-M+1) X(n-M+2) \dots X(n)]$ at any time instant n , where k is the time resolution and M an integer. Any two consecutive bit arrival rate information can be identified by their sum and difference. The subject of interest is the dynamic behavior of the traffic that manifests itself through the differences among neighboring samples. The difference operator reveals sharp changes in the arrival rate. Arrival rate vector \underline{X}_{k+1} consisting of M consecutive time slots when represented at time resolution $k+1$ is

$$\begin{aligned} \underline{X}_{k+1} = & 1/2[X(n-M+1)+X(n-M+2) \\ & X(n-M+3)+X(n-M+4) \dots X(n-1)+X(n)] \end{aligned} \quad (4)$$

The difference of the arrivals between two consecutive time slots is denoted by vector \underline{Y}_{k+1} such that

$$\begin{aligned} \underline{Y}_{k+1} = & 1/2[X(n-M+1)-X(n-M+2) \\ & X(n-M+3)-X(n-M+4) \dots X(n-1)-X(n)] \end{aligned} \quad (5)$$

When generalized

$$\underline{X}_{k+1}(i) = 0.5(\underline{X}_k(2i-1) + \underline{X}_k(2i)) \quad (6)$$

$$\underline{Y}_{k+1}(i) = 0.5(\underline{X}_k(2i-1) - \underline{X}_k(2i)) \quad (7)$$

Note that (6) and (7) imply the scaling and wavelet transform coefficients of the Haar wavelet at resolution k with only difference being the value of the constant multiplier. The scaling and wavelet coefficient vectors of the Haar wavelet are $\phi = [1/\sqrt{2} \ 1/\sqrt{2}]$ and $\varphi = [1/\sqrt{2} \ -1/\sqrt{2}]$ respectively.

It is true that for $\forall i, j$, traffic bit arrival rate process is positive, that is $\underline{X}_i(j) \geq 0$. Wavelet domain modeling

of positive processes requires the constraint that a positive output is ensured. To guarantee the constraint that the process is positive, the sufficient and necessary condition is $|\underline{Y}_i(j)| \leq \underline{X}_i(j)$. The Haar wavelet provides this constraint, but not higher order wavelets such as 4 or more tap Daubechies. Also LMS and RSL based predictors may return negative values, and therefore violate positive output constraint.

Having R as the $M \times M$ wavelet transform matrix composed of parameters of vectors ϕ and φ , and \underline{X} as the vector data with length M , the wavelet transform operation can be expressed as $\underline{W} = \underline{X} \cdot R$ where \underline{W} is the wavelet transform vector with size M . For example, in a three level dyadic tree, there are three high frequency sub-bands and one low frequency band. After taking the Haar wavelet transform of the traffic data, the signal energy in each high frequency sub-band can be computed as in (8)

$$E_j = \sum_{n=2^{j-1}+1}^{2^j} |\underline{w}(n)|^2 \quad (8)$$

where j is the index of high frequency sub-bands such that $0 < j < 4$ and $j = \log_2 \text{size}(X) - j + 1$. The energy in the

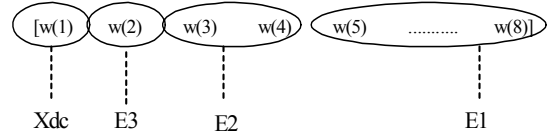


Figure 3 Wavelet transformed data vector \underline{W} and indexing of sub-band energies for $M=8$.

highest frequency band is represented by E_1 as illustrated in Fig.3.

It is easily proven that $X_{dc} > \text{mean}(\underline{X})$. We define $X_{dc} + \sqrt{\max(E_i)}$ as high and $X_{dc} + \sqrt{\min(E_i)}$ as low envelope levels respectively. These two constitute base bandwidth levels for any bandwidth increase and decrease. According to the type of the most recent interrupt, first, envelope levels are compensated, and second, the new bandwidth decision is returned. The following section explains how these compensation terms are obtained and final bandwidth decisions are made.

V. BANDWIDTH MANAGEMENT

In the FSBRM, there are three decision states and each decision state has a different generosity level in bandwidth allocation. State-II, that is S-II, allocates more bandwidth than State-I and III respectively. Each state computes the required bandwidth as follows:

$$S-II : X_{dc} + \sqrt{\max(E_i) + b(n)}/(\alpha \cdot IRI) \quad (9)$$

$$S-I : X_{dc} + \sqrt{\min(E_i)} \quad (10)$$

$$S-III : X_{dc} + \sqrt{\min(E_i) - u(n)}/(\beta \cdot IRI) \quad (11)$$

S-II would be only active provided that the renegotiation was due to buffering and that the previous bandwidth allocation was performed in either S-I or S-II itself. Therefore, the queue size $(b(n)/\alpha)$ that triggers the renegotiation request IRI time unit after the last renegotiation instant is added to the new bandwidth as a compensation term in S-II. On the other hand, if the renegotiation is due to underutilization and the previous allocation was granted in S-I or S-III, the new bandwidth is to be decreased by the under-utilized capacity per IRI time interval. Depending on the type of the interrupt causing a bandwidth renegotiation, state-to-state transitions occur and the bandwidth decision in the proper state is granted as shown in Fig.4. For example, assume that the last renegotiation amount was decided in S-I (Fig.5), and the new interrupt is generated due to buffer occupation. The state flows from S-I to S-II and new bandwidth is determined in state S-II, which would allocate more bandwidth than S-I and S-III. There is no direct transition from S-II to S-

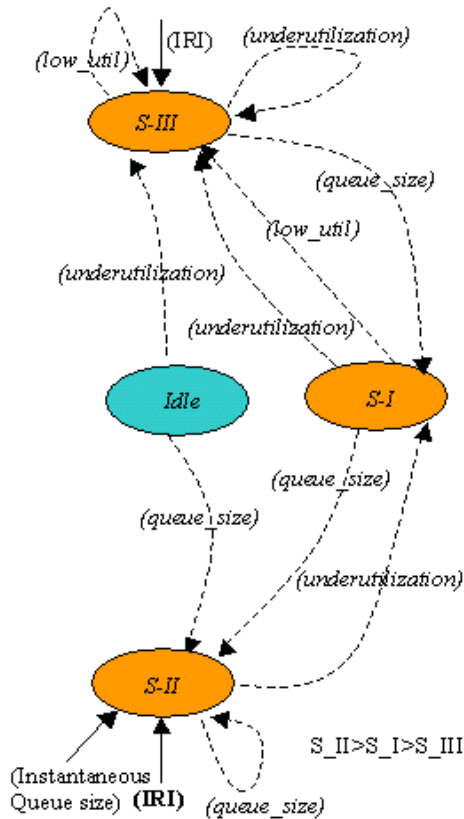


Figure 4 Finite-state bandwidth reallocation mechanism with three decision states.

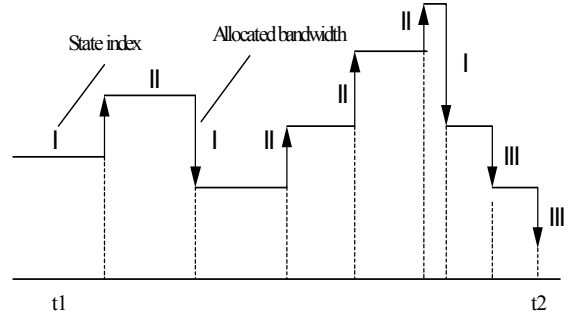


Figure 5 Illustration of the order of state-to-state transitions for each bandwidth decision.

III. This transition can only occur in two steps through S-I. The underlying reason is to prevent high fluctuations in the allocated bandwidth and to provide finer bandwidth control. In practice, it is harder to give additional bandwidth to a stream, because the network has to release bandwidth from other streams to meet that demand. This multi-state decision structure is a solution to prevent bandwidth level changes from high jumps, and also to lower the impact of prediction errors on performance.

VI. RESULTS and DISCUSSION

For performance testing, we use MPEG-1 frame size traces, which can be obtained from [5]. The frame sizes are in bits. We look into the following performance metrics for the comparison of each online method summarized in Section-II: *average utilization*, *number of renegotiations*, and *0.99-quantile-queue size*. Because of space limitations we present results for only two traces (The *StarWars* trace because of its high peak to mean arrival rate ratio, and the *Soccer* trace because of its high burstiness, that is high standard deviation of the arrival rates). The statistical properties of the traces are given in Table II.

The parameters for each method are selected considering the recommendations in related references. The same parameter values are used throughout the testing of each trace.

TABLE II
STATISTICS OF STARWARS AND SOCCER TRACES
(Note: STD-standard deviation, PMR-Peak-to-mean-ratio)

Trace	Peak(Kbpf)	Mean(Kbpf)	STD	PMR
StarWars	138	10.5	14.2	13.14
Soccer	190	25.1	21.2	7.6

Number of renegotiations can be set to a required value in synchronous methods. Therefore, to compare [1] with R++, the period of renegotiations are chosen so as to get the same number of renegotiations as R++ generates. We run R++ also in GOP prediction base mode to be able to compare it to NLMS. In the GOP mode, buffering and underutilization cost coefficients α and β respectively are set to higher values.

There are four parameters to adjust in R++: T_{\max} , T_{\min} , α , and β . Higher α results in smaller queue sizes and higher β in higher utilization. If both are increased simultaneously, the number of renegotiations multiplies.

TABLE III
PERFORMANCE COMPARISON OF R++ WITH DYNAMIC BANDWIDTH ALLOCATION METHODS

	Method	N	B (Kb)	$\bar{\rho}$	Parameters
Trace: Star Wars (GOP: 12f)	R++	882	237	91	$T_{\min}=150Kb$ $\alpha=1.6, \beta=1.7$ $T_{\max}=1.1Mb$
	[1]	882	577	90	1 renegot/45f
	RCBR	1291	243	88	Bh=200Kb, Bl=10Kb, $\Delta=150Kbps, T=5f$
	RDBA	1129	230	72	$T_{\max}=1Mb$ $\alpha=1.6, \beta=1.7$ $T_{dec}=25Kb$
	NLMS	780	410	78	$\mu=0.1$
	[1]	474	620	82	1 renegot/85f
	R++(GOP)	474	251	82	$\alpha=16, \beta=17$
Trace: Soccer (GOP: 12f)	R++	1040	299	93	$T_{\min}=150Kb$ $\alpha=1.6, \beta=1.7$ $T_{\max}=1Mb$
	RCBR	1321	324	93	Bh=200Kb, Bl=10Kb, $\Delta=150Kbps, T=5f$
	RDBA	1314	390	83	$T_{\max}=1Mb$ $\alpha=1.6, \beta=1.7$ $T_{dec}=25Kb$
	NLMS	660	2500	92	$\mu=0.1$
	[1]	1040	1570	93	1 renegot/38f

It is presented in Table III that, for the *Star-Wars* trace, R++ can achieve the same queue size performance as RCBR, but 3% better utilization and 24% less renegotiations. It also outperforms RDBA, NLMS, and the synchronous method in [1]. We should note the relative difference on behalf of R++ that NLMS is GOP based and it does 780 reallocations for 3333 GOPs, and R++ 882 for 40000 frames. When R++ is run in GOP prediction mode, it also achieves better N, B and $\bar{\rho}$ than NLMS. Also, for a highly variable bit rate traffic (the *Soccer*), R++ adapts better than the compared methods, and attains the same utilization level and smaller 0.99 quantile queue size after less number of renegotiations.

CONCLUSION and FUTURE WORK

It is shown that the performance of R++ is better than other renegotiation based methods given in the literature. However, we haven't studied and do not

know the impact of renegotiation delay on the efficiency of R++ and the other compared methods yet. This would be a subject to future work. The presented work is concerned with resource management for a single stream. We also would like to enhance R++ for scenarios that multiple streams with different cost constraints compete for the same resources. Moreover, future work will include analysis of the characteristics of the bandwidth increment and decrement processes of available dynamic methods, and heavy-tailedness of queue sizes resulting from each renegotiation scheme. We will also investigate how the latency in completion of resource renegotiations would degrade the performance of the introduced scheme.

REFERENCES

- [1] E. Casilari, F. Sandoval, "Bandwidth Renegotiation Scheme for VBR Video Services," IEE Electronics Letters, v:35, n:18, pp.1509-1510, September 1999.
- [2] M. Grossglauser, S. Keshav, D. N. C. Tse, "RCBR: A Simple and Efficient Service for Multiple Time Scale Traffic," IEEE Trans. on Networking, v:5, n:6, pp. 741-755, December 1997.
- [3] F. Porikli, Z. Sahinoglu, "Dynamic Bandwidth Allocation with Optimal Number of Renegotiations in ATM Networks," in proc. of ICCCN'01, pp.290-295, Scottsdale, AR, October 2001.
- [4] A. M. Adas, "Using Adaptive Linear Prediction to Support Real-time VBR Video Under RCBR Network Service Model," IEEE Trans. on Networking, v:6, n:5, pp.635-644, October 1998.
- [5] <http://nero.informatik.uni-uerzburg.de/MPEG>