respect to previous work [2]. Fig. 1 shows the circuit of a four stage travelling wave amplifier which is used as a benchmark. This circuit is driven by two different input signals represented by the I_L and I_H independent current sources, both of amplitude 1mA and working at frequencies f_L = 100MHz and f_H = 101MHz, respectively. A square frequency schema of order five has been used for the HBM [1, 2].

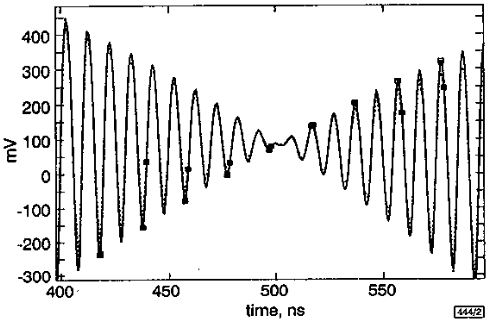

Fig. 2 *Portion of vout voltage-time waveforms at output of travelling waveform amplifier with I_H = 1.0mA and I_L = 1.0mA*

■ DFT
□ APFT
○ MTFT

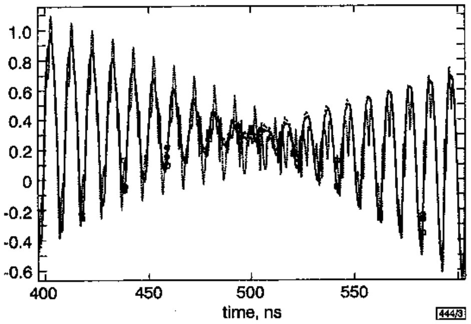

Fig. 3 *Portion of v_{out} voltage-time waveforms at output of travelling waveform amplifier with I_H = 1.8mA and I_L = 1.8mA*

■ DFT
□ APFT
○ MTFT

In the next simulation, the amplitudes of both the I_L and I_H current sources were increased to 1.8mA since in this case the amplifier no longer operates in the 'linear region' and the magnitudes of the high frequency harmonics are augmented. Note that, if the amplitudes of the current sources are further increased, the APFT fails to converge. The spectrum truncation due to the unavoidable finite-order schema of the HBM (H = 61) generates aliasing that must not be amplified. Fig. 2 shows a portion of the steady state waveform obtained in the first case with both APFT and MTFT and Fig. 3 shows the results in the second case. We note that the waveforms are in good agreement in the first case when there is little aliasing, but are very different in the latter case. Figs. 2 and 3 also show the results (DFT) obtained with the conventional HBM for a fundamental frequency $f_H - f_L$ = 1MHz and 256 harmonics. We can consider these results as exact; their good agreement with the results obtained for the MTFT is shown.

Conclusions: A simple approach that extends the HBM to simulate circuits driven by multitone signals has been presented. The proposed method always generates orthogonal conversion matrices as conventional DFTs and has the same numerical properties and stability. Simulation results confirmed the improved accuracy of the proposed technique with respect to previous methods presented in literature.

© IEE 1999 *26 July 1999*
Electronics Letters Online No: 19991241
DOI: 10.1049/el:19991241

A. Brambilla (*Dipartimento di Elettronica e Informazione, Politecnico di Milano, p.za Leonardo da Vinci, 32,I-20133 Milan, Italy*)

References

1 KUNDERT, K., SORKIN, G.B., and SANGIOVANNI-VINCENTELLI, A.: 'Applying harmonic balance to almost-periodic circuits', *IEEE Trans. Microw. Theory Tech.*, 1988, **36**, (2), pp. 366–377

2 CHUA, L.O., and USHIDA, A.: 'Algorithms for computing almost periodic steady-state response of nonlinear systems to multiple input frequencies', *IEEE Trans. Circuits Syst.*, 1981, **CAS-28**, pp. 953–971

3 DREIFUSS, J., MADJAR, A., and BAR-LEV, A.: 'An improved version of the almost periodic Fourier transform algorithm with applications in the large-signal domain', *IEEE Trans. Microw. Theory Tech.*, 1991, **39**, (3), pp. 571–575

4 LIU, Q.H., and NGUYEN, N.: 'An accurate algorithm for nonuniform fast Fourier transform (NUFFTs)', *IEEE Microw. Guid. Wave Lett.*, 1998, **8**, (1), pp. 18–20

Negotiated disk admission control in video streaming

W. Lee and J. Srivastava

A novel soft-quality of service framework for continuous media servers, which provides a dynamic and adaptive disk admission control and scheduling algorithm, is presented. Using this framework, the number of simultaneously running clients for video servers could be increased, and better resource utilisation under heavy communication traffic requirements could be ensured.

Introduction: To ensure continuous and stringent real-time constraints of video delivery in continuous media (CM) servers, several factors such as disk bandwidth, buffer capacity, network bandwidth etc. should be considered carefully and should be handled efficiently. A video stream being viewed requires timely delivery of data, but is able to tolerate some loss of the data for small amounts of time. Thus, an acceptable method of degrading service quality is simply to reduce the requested resources of CM applications. The idea of achieving higher utilisation by introducing soft-quality of service (QoS) is not novel in the networking field [1, 2], but the problem here is dual (i.e. disk admission control and I/O bandwidth management in CM server systems) to that of network call admission control and dynamic bandwidth management. According to the evaluation of human perception on video and audio [4], up to 23% of aggregate video loss and 21% of aggregate audio loss are tolerable. Henceforth, using this fact, we could restrict and steal some resources of running streams, and give them to the other new input requests, based on the logged information and the status of remaining resources given by our admission control algorithm, without causing any perceptible changes for users.

In this Letter, we present an adaptive disk admission control strategy for providing fair scheduling and better performance in video streaming. This new stream scheduler provides a proficient admission control functionality which can optimise disk utilisation with a good response ratio for the requests of clients and in particular under heavy loaded traffic environments. The presented scheduler still presents similar levels of stream quality (how well the requested rate is attained in CM servers) compared to the basic greedy admission control algorithm.

Negotiated disk admission control: Most of the existing admission control approaches are purely greedy (basic) in strategy in the sense that a new application (video stream) can be accepted only if the server could give the client all the requested resources [3]. These approaches are too conservative and admit too few streams, thereby under-utilising the server resources. The key advantage of the basic greedy strategy is that it is simple. However, owing to the other shortcomings of the basic greedy admission control algorithms, we should be able to think of another strategy (RAC: reserve-based admission control algorithm) to allow more streams to run concurrently in continuous media servers by degrading the requested quality of newly arriving streams and by adapting the returned resources to these streams. We can say that our approach is adaptive and negotiated in that it is capable of re-adjusting resources according to the amount of remaining resources. The key idea here is to assign a portion of the resources as the reserves, and when the applications start to dip into the reserves, another strategy is invoked. Given the average seek time t_s, cycle length T_{svc}, disk block size b, rotation time t_r, and disk transfer rate R, the consumption rate of each client request (q_i) is a variable in typical I/O bandwidth constraints. We initialise the total available rate (T_{avail}) with T_{svc} (= T_{total}), and set $T_{reserve}$ (the threshold value for criteria to check *congested_bit*). Initially the *congested_bit* is 0 (not congested). The resource (here I/O bandwidth) is congested if the resource usage (T_{used}) is more than ($T_{svc} - T_{reserve}$). We can set the $T_{reserve}$ amount of resources to admit more requests at a reduced quality. Here, $T_{usage} + T_{avail} = T_{svc}$; i.e. the *congested_bit* is set (= 1) only when T_{avail} becomes smaller than $T_{reserve}$; otherwise, the basic I/O bandwidth constraint is just applied (i.e. the new request is admitted without being degraded). While the *congested_bit* is set, we restrict the degree of resource assignment to the new request because there are insufficient resources remaining. On adding the request q_i, if the resource remains uncongested, we then admit it with no degradation: $T_{alloc} = q_i$. We let the current usage be T_{used}; then, the addition of the application q_i will lead to an increase in usage to ($T_{new} = T_{used} + q_i$). If ($T_{new} \geq T_{svc} - T_{reserve}$), then the addition of q_i will make the resource congested. When the resource becomes congested, we have to dip into the resources: $T_{free\_res} = \max(0, T_{svc} - T_{reserve} - T_{used})$. This is the unused resource outside the reserves. We let T_{avail} be the amount of resources in the reserves: $T_{free\_res} = \min(T_{reserve}, (T_{svc} - T_{used}))$. The new application is allocated, and then $T_{alloc} = T_{free\_res} + \min((q_i - T_{free\_res}), \Psi(T_{free\_res}))$. That is, we allocate the resources that are outside the reserves and, in addition, we give a maximum $\Psi(T_{free\_res})$ of the resources in the reserves. $\Psi(.)$ is a function of T_{free_res} and returns an appropriate amount of resources according to T_{free_res}. If $\Psi(T_{free\_res})$ is too small (in this case, we must not support the new request because the display quality may be too poor), we simply reject the request. If the running stream (s_k) is over, then the resource v_i is reclaimed. We return the resource occupied by the stream and adjust (if required) the resource allocation of streams not being fully serviced. Here the unused resource is allocated to applications such that the least serviced streams could obtain the returned resource first.

We have implemented two heuristic-based methods for negotiation (reclamation) and tested their performance.

(i) *Dynamic reserve adaptation* (*DRA*): DRA is invoked only if applications exist that need resources and do not have them. On the departure of streams, DRA returns the assigned resources to the leaving streams (q_i) back to the available resource pool and recalculates the new T_{avail} using the total available resources (i.e. old $T_{avail} + q_i$). Among the requests that are not fully serviced yet, we first select a request (s_{min}) from the queue (DS), with smallest *done_ratio* (ratio of serviced resource to requested resource for a stream), and assign the proper resource to the request. The maximum $\Phi(T_{free\_res})$-rule applies here. We assign $T_{alloc} = \min(\Phi(T_{free\_res}), remaining\_rate_{min})$ to the selected request (s_{min}). The loop continues until there are no more available resources to assign or T_{alloc} is too small to assign.

(ii) *Reclamation within returned reserve* (*RWR*): The RWR method is similar to the DRA method, but the difference is that it redistributes only the resources returned from the leaving streams (v_i). In DRA, we used T_{avail} (instead of v_i) for reclamation. The T_{alloc} per stream is calculated according to the ratio of *remaining_ratio* to *sum_remaining_ratio*. The *remaining_ratio* is the ratio of the resource yet serviced to the requested ratio for a stream, and *sum_remaining_ratio* is the total sum of the *remaining_ratio*s of the degraded streams. $T_{alloc} = T_{avail} * (remaining\_ratio_k/sum\_remaining\_ratio)$ The key idea here is not to touch the reserves, but to utilise only the returned resources due to the departure of streams.

When the resources in the reserves are not used for a long time, some are reclaimed. For every k period, if there is no request to the reserve, then we release some amount of the reserve to be reclaimed. This is a kind of reserve adaptation method.

Experimental results: We performed extensive simulation to validate our admission control and scheduling algorithm on a Sun Ultra Sparcstation with a Seagate Barracuda 4 GB disk (ST34371N). We measured the following metrics: (i) admission ratio and accumulated number of admitted streams over time, (ii) total system utilisation, and (iii) total quality.

The major advantage of the RAC is that it guarantees a greater number of streams will be admitted and can run simultaneously with a tolerable degradation of quality. The admission ratio is used to evaluate this property, i.e. the ratio of the number of admitted streams to the total number of requested streams. We observe that the RAC (both RWR and DRA) strategy noticeably increases the admission ratio compared to the greedy method (as shown in Fig. 1*a*).

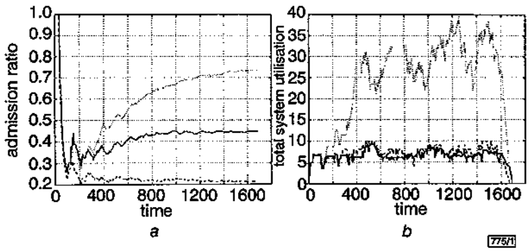

Fig. 1 *Admission ratio and total system utilisation against time*

a Admission ratio
b Total system utilisation
·········· $T_{reserve}$ = 0.3 (RWR mode)
——— $T_{reserve}$ = 0.3 (DRA mode)
– – – – $T_{reserve}$ = 0.0 (greedy method)

We next evaluated the effect of the RAC algorithm with respect to total system utilisation and display quality. As shown in Fig. 1*b*, it is observed that the total system utilisation achieved by RWR is much higher during most of the period (i.e. the RWR algorithm utilises the system resources more efficiently than the greedy algorithm). Nevertheless, the total quality achieved by the RWR algorithm is almost the same as that achieved by the greedy algorithm. In contrast, the performance of the DRA algorithm in total quality is low most of the time. This arises because the DRA algorithm holds off the reserves for future streams, which arrive soon before some of the current running streams depart. However, under medium traffic, the performance of DRA improves.

A summary of the performance evaluation results is shown in Table 1.

Table 1: Summary of performance results: RAC (RWR and DRA) against greedy method

	RAC		Greedy
	RWR	DRA	
Number of running streams	300–400%	180–200%	100%
Admission ratio (max: 1.0)	0.3–0.75	0.3–0.45	0.2–0.3
System utilisation	250–380%	95–100%	100%
Video quality (max: 100%)	85–100%	60–100%	100%

Conclusions: We have presented an integrated adaptive admission control and scheduling algorithm for CM server systems. We have also presented results of a simulation evaluation of our algorithm and a greedy algorithm with respect to several metrics designed to measure the admission ratio, total quality, and system utilisation. It was observed that under heavy traffic, the proposed algorithm achieves much better performance than the greedy algorithm. Using this scheme, we could expect that more streams could be running with an acceptable range of data quality in a given system resource.

Acknowledgment: This work is supported by Army Research Lab (ARL) number DA/DAKF11-98-9-0359 to the University of Minnesota.

9 August 1999
Electronics Letters Online No: 19991258
DOI: 10.1049/el:19991258

W. Lee and J. Srivastava (*Department of Computer Science and Engineering, University of Minnesota at Minneapolis, MN 55455, USA*)
E-mail: WJLEE@CS.UMN.EDU

References

1 CHANG, J.T.C.-S.: 'Effective bandwidth in high-speed networks', *IEEE J. Sel. Areas Commun.*, 1995, **13**, (6), pp. 1091–1100

2 GROSSGLAUSER, D.M.: 'Measurement-based call admission control: Algorithms and analysis'. IEEE INFCOM'97, April 1997,

3 RAGHAVAN, S.K.T.S.V.: 'Networked multimedia systems: concepts, architecture, and design' (Prentice Hall, 1998)

4 WIJESEKERA, D., and SRIVASTAVA, J.: 'Experimental evaluation of loss perception in continuous media', to be published in *Multimedia Syst. J.*, 1999

Noise robust alternating fixed-point algorithm for stereophonic acoustic echo cancellation

N.T. Forsyth, J.A. Chambers and P.A. Naylor

A novel algorithm specifically for use in stereophonic acoustic echo cancellation (SAEC) environments is introduced. It is based on an alternating fixed-point (FP) structure. Analysis provides bounds to ensure that the algorithm has the form of a contraction mapping (CM). Simulation results show improved performance over algorithms with similar computational complexity in the presence of noise.

Introduction: In communications using hands-free telephony and teleconferencing systems, there is often a need for some kind of echo cancellation. An acoustic echo canceller is designed to adaptively model the acoustic path through which speech travels. To give more spatial realism, two channels can be used, but new difficulties are experienced in identifying multiple echo paths uniquely [1]. Consider Fig. 1; the transmission room on the right has two microphones to pick up the source signal in stereo via the two acoustic paths with impulse responses $g_1(n)$ and $g_2(n)$. These signals then pass to loudspeakers in the receiving room on the left and are coupled to one of the microphones via paths $h_1(n)$ and $h_2(n)$; the signal $y(n)$ is then produced. Stereophonic acoustic echo cancellation (SAEC) is then undertaken on this signal $y(n)$, otherwise the same signal would be passed to the transmission room as an echo. The echo canceller derives an estimate using two finite impulse response filters $\hat{h}_1(n)$ and $\hat{h}_2(n)$ to model the two echo paths in the receiving room. The loudspeaker signals $x_1(n)$ and $x_2(n)$ then form the inputs to these estimated paths, the outputs of which are summed to produce $\hat{y}(n)$. This is then subtracted from the echo signal, $y(n)$, giving the error signal. Assuming the error is driven to zero, we have the relationship:

$$\left(h_1(n) - \hat{h}_1(n)\right) * x_1(n) + \left(h_2(n) - \hat{h}_2(n)\right) * x_2(n) = 0 \tag{1}$$

where * denotes discrete convolution. It can be seen that unless $x_1(n)$ and $x_2(n)$ are linearly independent, the desired result $h_1(n) - \hat{h}_1(n) = h_2(n) - \hat{h}_2(n) = 0$, $\forall n$, cannot be guaranteed. Clearly this is not the case as $x_1(n)$ and $x_2(n)$ come from a common source and are inherently highly correlated. This coupling between the solutions leads to non-uniqueness in the solutions for both echo paths and something must be done to remedy this. The extended least mean square algorithm [1] takes into account the cross-correlation between the inputs signals, but its performance is limited. Other generalisations of single channel algorithms, such as normalised-LMS, (NLMS) and recursive least squares (RLS), to the two channel case either perform poorly or have an impractically high computational complexity due to the lengths of impulse responses involved, $O(1000)$, in room environments. A multichannel affine projection algorithm was proposed in [2], but it suffers from considerable sensitivity to noise. Therefore a robust algorithm for enhanced SAEC performance is needed.

Alternating fixed-point normalised least mean square two-channel algorithm (AFP-NLMS2): To avoid local minima associated with non-uniqueness, a new search-space is created by freezing one channel of the adaptive echo canceller and updating the other for a number of iterations. The proposed new algorithm is based on this concept, but at each sample we first use the ε-NLMS2 algorithm as a main loop to update both channels and then fix one channel and update the other in a fixed-point (FP) manner. The ε-NLMS2 algorithm is a two-channel NLMS algorithm with an adaptation gain $\mu = \mu'/(\varepsilon + \|\underline{x}\|_2^2)$ where $\underline{x} = [\underline{x}_1^T, \underline{x}_2^T]^T$, $[.]^T$ is a vector transpose and $\|.\|_2$ is an l_2 norm. After the main loop at the next time sample, the FP procedure alternates to the other channel. To try to make the solution unique, orthogonal projections in the form of Gram-Schmidt procedures are used within the fixed-point iterations (FPIs) to remove information that is deemed to be detrimental to the identification of a distinct echo path for that channel. The first FPI maps one channel onto a subspace that is orthogonal to the other channel at that time, denoted $\mathbf{P}^{\perp}_{x2(n)} x_1(n)$. This can be viewed as a cross-channel projection as it removes information in the data for one channel that is correlated to that in the other channel. The second FPI uses the previous channel input and removes information that is correlated with the present channel input, $\mathbf{P}^{\perp}_{x1(n)} x_1(n-1)$. The third is similar, but looks forward two iterations, $\mathbf{P}^{\perp}_{x1(n),x1(n\text{-}1)} x_1(n-2)$. These second and third FPIs are cross-through-channel projections as they remove data that are correlated to the future data in that channel. The number of FPIs can be increased further, although complexity increases with performance. The algorithm is formalised below (subscripts denote main loop iteration number, channel number; c denotes channel 1 or 2; main index denotes FPI number) with μ normalised as above.

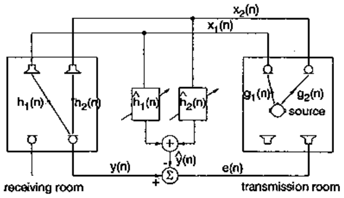

Fig. 1 *Schematic diagram of stereophonic acoustic echo cancellation*

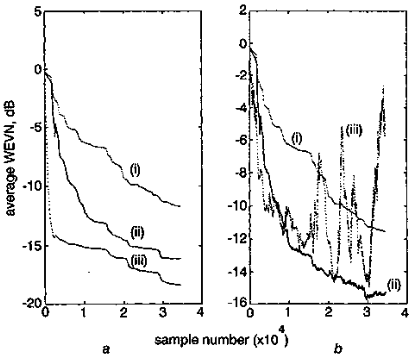

Fig 2 *Comparison of weight error vector norms of ε-NLMS2, AFP-NLMS2 and MCAPA algorithms*

a no noise; *b* SNR = 30dB
(i) ε-NMMS2, (ii) AFP-NLMS2 ($N = 3$), (iii) MCAPA ($N = 3$)

Main loop iterations – n:

$$\begin{aligned}\underline{\hat{h}}_{(n+1,1)} &= \underline{\hat{h}}_{(n,1)} + \mu \underline{x}_{(n,1)} e_{(n)} \\ \underline{\hat{h}}_{(n+1,2)} &= \underline{\hat{h}}_{(n,2)} + \mu \underline{x}_{(n,2)} e_{(n)}\end{aligned} \tag{2}$$

where $e_{(n)} = d_{(n)} - \underline{x}_{(n,1)}{}^T \, \underline{\hat{h}}_{(n,1)} - \underline{x}_{(n,2)}{}^T \, \underline{\hat{h}}_{(n,2)}$

Fixed-point iterations - k: If mod(n) = 0, $c = 1$, $\bar{c} = 2$, else $c = 2$, $\bar{c} = 1$

Initialisations: $\underline{\hat{h}}_{(c)}[1] = \underline{\hat{h}}_{(n+1,c)}$, $e[1] = e_{(n)}$

(i) First update $\underline{\hat{h}}_{(c)}[2] = \underline{\hat{h}}_{(c)}[1] + \mu e[1] \mathbf{P}^{\perp}_{\underline{x}\,(n,\bar{c})} \underline{x}_{(n,c)}$

(ii) Then calculate the new error $e[2] = d_{(n-1)} - \underline{x}^T{}_{(n-1,c)} \underline{\hat{h}}_{(c)}[2] - \underline{x}^T{}_{(n-1,\bar{c})} \underline{\hat{h}}_{(n+1,\bar{c})}$
with update $\underline{\hat{h}}_{(c)}[3] = \underline{\hat{h}}_{(c)}[2] + \mu e[2] \mathbf{P}^{\perp}_{\underline{x}\,(n-1,\bar{c}),\underline{x}(n,c)} \underline{x}_{(n-1,c)}$

(iii) The new error $e[3] = d_{(n-2)} - \underline{x}^T{}_{(\cdot-2,c)} \underline{\hat{h}}_{(c)}[3] - \underline{x}^T{}_{(n-2,\bar{c})} \underline{\hat{h}}_{(n+1,\bar{c})}$
then update $\underline{\hat{h}}_{(c)}[4] = \underline{\hat{h}}_{(c)}[3] + \mu e[3] \mathbf{P}^{\perp}_{\underline{x}\,(n-2,\bar{c}),\underline{x}(n-1,c),\underline{x}(n,c)} \underline{x}_{(n-2,c)}$