

Model-free Adaptive Hysteresis for Dynamic Bandwidth Reservation

Nail Akar

Electrical and Electronics Engineering Department
Bilkent University, Ankara, Turkey
Email: akar@ee.bilkent.edu.tr

Abstract—Dynamic bandwidth reservation refers to the process of dynamically updating the bandwidth allocation to a connection between two network end points on the basis of actual aggregate traffic demand of the connection. We assume a scenario in which bandwidth updates for the connection should not be performed too frequently and the frequency of updates are thus limited to a so-called desired update rate. We propose an asynchronous model-free adaptive hysteresis algorithm for dynamic bandwidth reservations with such update frequency constraints. We validate the effectiveness of the proposed approach by comparing its bandwidth efficiency with that of a synchronous model-based dynamic bandwidth reservation mechanism from the existing literature.

I. INTRODUCTION

In this paper, we consider a generic connection carrying aggregate traffic between two network end points with a certain bandwidth reserved for this connection. In this setting, Dynamic Bandwidth Reservation (DBR) refers to the process of dynamically changing the connection's bandwidth reservation on the basis of the instantaneous aggregate traffic demand of the connection. We assume that the route of the connection is fixed and the connection's bandwidth reservation can dynamically be resized on-line using signaling mechanisms without a need for tearing it down and re-establishing it. In this paper, we use "connection" as an umbrella term that encompasses a variety of connection-oriented networking technologies, for example an Asynchronous Transfer Mode (ATM) VP [1], a Label Switched Path (LSP) in a Multi-Protocol Label Switching (MPLS) network [2], or a single aggregate RSVP (Resource ReserVation Protocol) reservation aggregating individual end-to-end RSVP reservations [3]. Dynamic bandwidth reservation also arises in a wide variety of network resource allocation problems such as dynamic time division duplexing, reconfigurable SONET/WDM networks, passive optical networks, and fixed broadband wireless access systems like Wimax.

The proposed framework is more general and does not assume a traffic model to be available. However, for the sake of being able to compare our results with those in the existing literature, we focus our attention in this paper to a single-class traffic scenario in which individual calls arrive at the connection-oriented network according to a non-homogeneous Poisson process with rate $\lambda(t)$ and call holding times are exponentially distributed with mean $1/\mu$. We set the maximum arrival rate $\lambda_m = \max_t \lambda(t)$ over the time interval

of interest. These individual calls are then aggregated into a connection in the core network whose bandwidth needs to be dynamically adjusted on the basis of instantaneous aggregate traffic demand. One possibility is that the bandwidth of the connection may be adjusted every time a new call arrives or an existing call terminates for best use of bandwidth resources. This approach, however, defeats one of the key benefits of aggregation, i.e., reduction of message processing cost in the aggregation region [3]. Using ATM terminology, this approach will be referred to as the SVC (Switched Virtual Circuit) approach; see for example [1]. Another approach to engineer the connection is through reserving bandwidth for the largest traffic demand over a long time window (e.g., 24-hour period). This approach would not suffer from signaling and message processing requirements since each reservation update would take place only once in a very long time window. However, the downside of this approach is that the capacity may be vastly underutilized when the actual traffic demand is significantly lower than the allocated bandwidth. We call this approach the PVP (Permanent Virtual Path) approach.

In our model, each individual call requires one unit of bandwidth. We also impose a maximum bandwidth reservation denoted by C_m . We suggest to set C_m to the bandwidth required for the connection to achieve a desired call blocking probability P_b in the worst case scenario, i.e., $\lambda(t) = \lambda_m$. The parameter C_m can be derived using the Erlang's B formula which gives the blocking probability $P(\rho_m, C_m)$ in terms of the maximum traffic intensity $\rho_m = \lambda_m/\mu$ and C_m [4]:

$$P_b = P(\rho_m, C_m) = \frac{\rho_m^{C_m}/C_m!}{\sum_{k=0}^{C_m} \rho_m^k/k!}. \quad (1)$$

We introduce a desired update rate parameter β to address the tradeoff between bandwidth efficiency and message processing costs. Our goal in dynamic bandwidth reservation is then to select the update decision epochs to dynamically vary the allocated bandwidth $R(t)$ at time t for the connection as a function of the number of ongoing calls in the system denoted by $N(t)$ so as to minimize the average bandwidth use over time subject to the bandwidth constraint

$$N(t) \leq R(t) \leq C_m \quad (2)$$

while keeping the long-term frequency of bandwidth updates less than β .

There are two different approaches to dynamic bandwidth reservation for connection-oriented networks under update frequency constraints, namely the synchronous and asynchronous approaches. In the synchronous approach, the bandwidth reservation for a connection is adjusted at regularly spaced time epochs with a frequency dictated by the signaling constraints. The ref. [5] proposes that at a decision epoch, a new capacity is reserved for the aggregate depending on the current system occupancy so that the expected time average of the blocking probability in the forthcoming interval will be less than a predefined limit. A numerical algorithm is proposed by [6] for the efficient numerical calculation of such time-dependent blocking probabilities. A practical example for synchronous bandwidth adjustment for MPLS LSPs but for more general data networks is the auto-bandwidth allocator by [7]. In the autobandwidth allocator proposal, there are two types of intervals, a Y-type interval (default 24 hours) and an X-type interval (default 5 minutes). The average bandwidth requirement is observed for each X-type interval within a Y-type interval and the highest of these X-type averages is allocated for the aggregate for the next Y-type interval. The ref. [8] proposes a scheme for inter-domain resource management by estimating the inter-domain traffic using a Kalman filter and then forecasting the capacity requirement at a future instant by the use of transient probabilities of the system states. An ARIMA-based traffic model in conjunction with a traffic forecasting and synchronous bandwidth provisioning scheme is proposed in [9].

Restricting the bandwidth update decisions to regularly spaced time epochs as in the synchronous approach might result in inefficiencies in bandwidth utilization. In the asynchronous DBR approach, such bandwidth adjustments take place asynchronously and corresponding decision instants depend on the current system state. An early work on this approach is by [10] which proposes the increase of bandwidth by a constant predetermined step each time the current bandwidth cannot accommodate a new call and the bandwidth is decreased by the same constant step when the bandwidth requirement drops back to the original value. Two drawbacks of this proposal are the potential oscillations around a threshold which might substantially increase the signaling load and the wastage of bandwidth as the number of active calls grows due to the use of the constant step. In [11], a model-based bandwidth allocation policy is proposed that eliminates the above problems by applying adaptive thresholds and hysteresis. In [12], simple operational rules are derived to determine the amount of bandwidth resources to different connections while balancing between bandwidth waste and connection processing overhead. A heuristic is proposed for a similar problem for a channel sharing application by [13] which however falls short of ensuring a desired update rate. The ref. [14] proposes a scheme for MPLS networks that uses continuous-time Markov decision processes. The proposed scheme decides on when an LSP should be created and how often it should be re-dimensioned while taking into consideration the trade-off between utilization of network resources

and signaling/processing load incurred on the network.

In this paper, we propose a model-free adaptive algorithm for dynamic bandwidth reservations. Our proposed approach is asynchronous since potential bandwidth update epochs coincide with a new call arrival or call departure. Moreover, the proposed system does not assume a traffic model to be available and therefore it is applicable to a wide range of scenarios with unpredictable traffic patterns. The approach uses hysteresis to control the number of updates but the hysteresis operation regime and the band of the hysteresis vary adaptively over time based on system state and the occupancy of a leaky bucket that we incorporate for the aim of update frequency control.

Section 2 discusses the synchronous approach for DBR. In Section 3, we present the model-free adaptive hysteresis-based DBR algorithm that we propose in this paper. Numerical examples are provided in Section 4. We conclude in the final section.

II. SYNCHRONOUS DYNAMIC BANDWIDTH RESERVATION

In synchronous DBR, the system is sampled at regularly spaced epochs $kT, k = 0, 1, 2, \dots$, where T is the update period. The minimum bandwidth reservation $R(kT) = R_k$ that ensures a desired blocking probability P_b throughout the time interval $[kT, (k+1)T)$ is then chosen on the basis of $N_k = N(kT)$ and $\rho_k = \frac{\lambda_k}{\mu}$ where λ_k is an estimate of the call arrival rate in the interval $[kT, (k+1)T)$. For calculating R_k we need to study the following problem. Given the number of calls N_k in progress at time 0 with a bandwidth allocation of R , the task is to calculate the probability of finding the system in state R at time t denoted by $P_{R|N_k}(t)$. The average blocking probability in an interval of length T is then given by

$$P(\rho_k, R, N_k, T) = 1/T \int_0^T P_{R|N_k}(t) dt \quad (3)$$

which approaches to $P(\rho_k, R)$ as $T \rightarrow \infty$. The bandwidth allocation R_k is then chosen as the minimum R for which $P(\rho_k, R, N_k, T)$ stays below the desired blocking probability P_b . In cases when R_k is larger than the physical limit C_m , R_k is set to C_m . The time-dependent probabilities in (3) can be calculated using numerical transient solutions of continuous-time Markov chains as in [15]. The overall procedure can be completed offline leading to a lookup table relating R_k to N_k . This synchronous DBR approach is model-based; we need to know the arrival and holding time distributions to obtain R_k given N_k which leads to a number of drawbacks:

- (i) When the traffic is non-stationary the traffic model parameters need to be estimated on-line and large lookup tables are necessary to address non-stationary behavior adding substantially to the implementation complexity of the approach.
- (ii) Solving large systems might be tedious.
- (iii) Restricting the decision epochs to regularly spaced time epochs as opposed to the asynchronous approach may not lead to the most effective strategy.

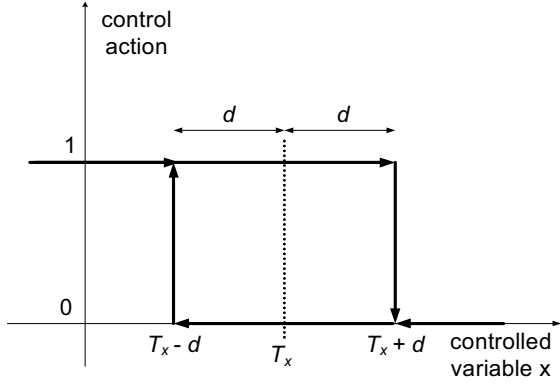


Fig. 1. A binary control system using static hysteresis.

Since this proposed algorithm is based on the transient solution of continuous-time Markov chains, we refer to this model-based algorithm by TADIR (TrAnsient solution-based Dynamic bandwidth Reservation).

III. ASYNCHRONOUS DYNAMIC BANDWIDTH RESERVATIONS WITH ADAPTIVE HYSTERESIS

Conventional static hysteresis-based control systems possess two actions, say 0 and 1, a controlled variable x , a threshold parameter T_x on the controlled variable, and a hysteresis band (or simply hysteresis) parameter d . The actions 0 and 1 help the controlled variable x to move lower and higher, respectively. If the value of x drops below the lower threshold $T_x - d$ then the action is 1; if this value is larger than the upper threshold $T_x + d$ then the action is 0; otherwise if the value is within the hysteresis band $(T_x - d, T_x + d)$ then the previous action does not change; see Fig. 1 depicting the hysteresis relationship between the controlled variable x and the control action. It is clear that the hysteresis control mechanism keeps the controlled variable close to the threshold value x whereas by suitably choosing the hysteresis band parameter d , the frequency of action changes can be controlled.

For the DBR problem of interest, we propose an adaptive hysteresis whose threshold and hysteresis parameters are made to vary appropriately in time. For this purpose, we first introduce a leaky bucket of size B_m that is drained at a rate of β unless the bucket is empty. This bucket is incremented by one credit every time a bandwidth update occurs. Let $B(t)$ denote the bucket occupancy at time t . Obviously, the bucket occupancy $B(t)$ staying around the bucket size B_m is indicative of too many recent bandwidth updates that would jeopardize the bandwidth update frequency constraint. In this case, the hysteresis band needs to be widest possible, i.e., $d(t) = C_m$, so that new bandwidth updates would not happen. On the other hand when $B(t) = 0$, the hysteresis band needs to be narrowest possible, i.e., $d(t) = 0$, since otherwise bandwidth update credits would be wasted. We therefore allow the hysteresis band $d(t)$ to be proportional with $B(t)$. In

particular, we propose to use linear control

$$d(t) = \frac{C_m}{B_m} B(t), \quad (4)$$

which clearly meets the requirements at the boundaries $B(t) = 0$ and $B(t) = B_m$. We leave other types of control (e.g., nonlinear) for future research. We'll now describe how the hysteresis threshold varies in time and how the bandwidth reservation updates are to be made. For this purpose, let t_i denote the i th bandwidth update epoch and let the bucket occupancy be $B(t_i^+)$ and the hysteresis parameter be $d(t_i^+)$ at time t_i^+ . Here, t_i^+ denotes the epoch just after the decisions are made, e.g., bandwidth update decisions, for the event occurring at t_i . We also assume that the current allocation be just changed at time t_i to $R(t_i^+)$. Let $N(t)$ denote the number of calls in progress in the system. We define two hysteresis thresholds, namely the lower threshold $N(t_i^+) - d(t)$ and the upper threshold $N(t_i^+) + d(t)$. Note that these two thresholds depend on the number of ongoing calls at the instant of the latest update. In time, $N(t)$ will vary randomly, the bucket occupancy will drop linearly, and the hysteresis band $(N(t_i^+) - d(t), N(t_i^+) + d(t))$ will shrink due to the drainage of the leaky bucket. Following t_i , a new arrival at time t will be admitted in the connection if $N(t) < C_m$ but otherwise would be dropped. In this case, $N(t^+)$ will be set to $N(t) + 1$. If the current reservation cannot accommodate the new call, i.e., $R(t) < N(t^+)$, then a bandwidth update needs to take place. On the other hand, when an existing call departs, we write $N(t^+) = N(t) - 1$. We now define an event as the union of an arrival or a departure. After an event takes place at time t , we need to decide on making a bandwidth update if one of two conditions below are met:

- (i) $N(t^+) > R(t)$ (5)
- (ii) $N(t^+) \notin (N(t_i^+) - d(t), N(t_i^+) + d(t))$ (6)

Note that when the second condition is met, the system occupancy does not lie in the hysteresis band making it possible for us to make a bandwidth reservation update. Upon an update decision, say at time t_{i+1} , the new bandwidth reservation and the new bucket values are expressed as:

$$R(t_{i+1}^+) = \min(C, N(t_{i+1}^+) + \lceil d(t_{i+1}) \rceil), \quad (7)$$

$$B(t_{i+1}^+) = \min(B_m, B(t_{i+1}) + 1), \quad (8)$$

$$d(t_{i+1}^+) = \frac{C_m}{B_m} b(t_{i+1}^+), \quad (9)$$

where $\lceil x \rceil$ gives the smallest integer $\geq x$. Note that at the bandwidth update instant t_{i+1} we rewrite the lower and upper thresholds of the hysteresis as $N(t_i^+) - d(t_{i+1}^+)$ and $N(t_i^+) + d(t_{i+1}^+)$, respectively, and the hysteresis band immediately starts to reduce in size in time. This procedure is repeated afterwards.

In order to describe how the proposed algorithm works, we construct an example system that starts at $t = 0$ and for which $C_m = B_m = 10$, $N(0^+) = 5$, $R(0^+) = 6$, $b(0^+) = 2$ and $\beta = 1/4$ updates/min. We assume at $t = 0^+$,

a bandwidth update has just occurred. Note that with this choice of β we have 15 update opportunities per hour. Instead of a teletraffic model, we introduce arrivals and departures at pre-specified instances for this system. The evolution of $N(t)$, $R(t)$, and the lower and upper hysteresis thresholds are illustrated in Fig. 2. Let us first focus our attention to the update epochs. At $t = 3$ and $t = 7$, we have departures from the system and condition (ii) in (6) is satisfied or in other words $N(t^+)$, $t = 3, 7$, lies outside the hysteresis band. Therefore, these two time instances are used for bandwidth updates as described in (7). At the time epochs $t = 14$ and $t = 15$, we have arrivals and we have corresponding bandwidth updates since the conditions (ii) and (i) in (6) and (5) are met for the first and second of these time epochs, respectively. We have two more updates at the time epochs $t = 16.5$ and $t = 19.5$ stemming from condition (ii). We study the same

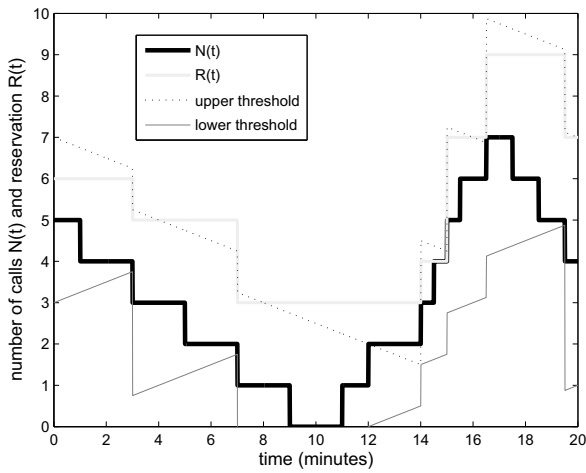


Fig. 2. The evolution of number of ongoing calls $N(t)$ and the reservation $R(t)$ as a function of t for a sample scenario for which $C_m = B_m = 10$, $N(0) = 5$, $R(0) = 6$, $b(0) = 2$ and $\beta = 1/4$ updates/min.

scenario but with the desired update rate increased to $\beta = 4$ updates/min in Fig. 3. It is clear that when the desired update rate increases, the proposed policy approaches to the SVC policy in which $R(t)$ tracks $N(t)$ since practically there is no signaling constraint. To see this, the desired update rate is large relative to the arrival rate and therefore the width of the hysteresis band drops to zero almost immediately and hence the occurrence of an arrival or departure triggers a bandwidth update. However, since the band is of zero width in all these events, $R(t)$ tracks $N(t)$ for this example as described in (7). These two examples are not meant to quantify the effectiveness of the approach but rather to help the reader in visualizing the basic features of the proposed algorithm. Throughout this paper, this proposed model-free method will be referred to as the ADAHIR (ADaptive Hysteresis-based asynchronous dynamic bandwidth Reservation) algorithm.

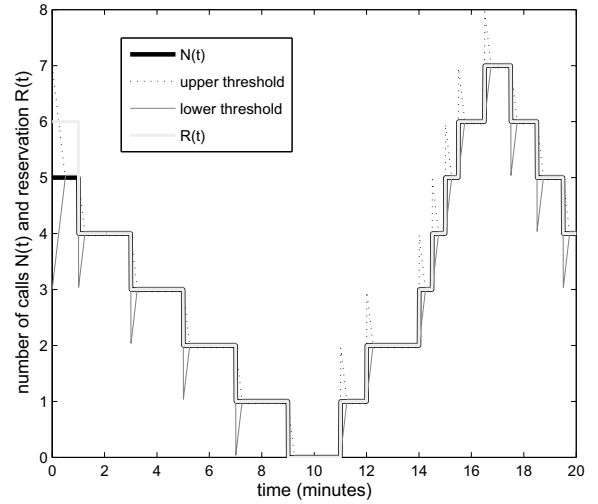


Fig. 3. The same example as in Fig. 2 with the desired update rate set to $\beta = 4$ updates/min.

IV. NUMERICAL EXPERIMENTATION

We compare the TADIR and ADAHIR algorithms through simulations for a time-homogenous system, i.e., $\lambda(t) = \lambda \forall t$, with $B_m = C_m = 16$, $1/\mu = 180$ sec., and $P_b = 0.01$. A single simulation is run for 2×10^7 overall calls and 5 such independent simulations are run to obtain the performance measures of interest, such as average reserved bandwidth, blocking probability, etc. We assume a fixed arrival rate $\lambda(t) = \lambda = 0.0493055$ calls/s. by means of recursively employing the Erlang B formula. We plot the average bandwidth reserved by these two algorithms as well as the average bandwidth used by the PVP and SVC methods as a function of the desired update rate β in Fig. 4. For the unit for β we use updates/hr. since this would be easier to visualize whereas for all other time units we prefer to use the conventional seconds. Note that in the PVP approach the reserved bandwidth always equals $C_m = 16$. For the SVC case, the average reserved bandwidth equals $\lambda(1 - P_b)/\mu$ since each accepted call will occupy one unit of bandwidth for $1/\mu$ sec. and the average reserved bandwidth equals 8.785 for the current example. We observe that the ADAHIR approach outperforms the TADIR approach for all values of the desired update rate β by taking advantage of asynchronous updates. While doing so, we note that ADAHIR is model-free and does not assume a traffic model to be available only except the value of C_m that ensures a blocking probability P_b . Moreover, as expected, for low values of β , ADAHIR approaches to the PVP policy whereas for very large β it approaches to the SVC policy. In the TADIR case, even for very large values of β , the bandwidth use would be slightly larger than that of the SVC policy. This is because $R_k \rightarrow \min(N_k + 1, C_m)$ as $T \rightarrow 0$ in (3) and the gap between N_k and R_k explains the gap between the SVC and TADIR curves for large β in Fig. 4. For ADAHIR, we

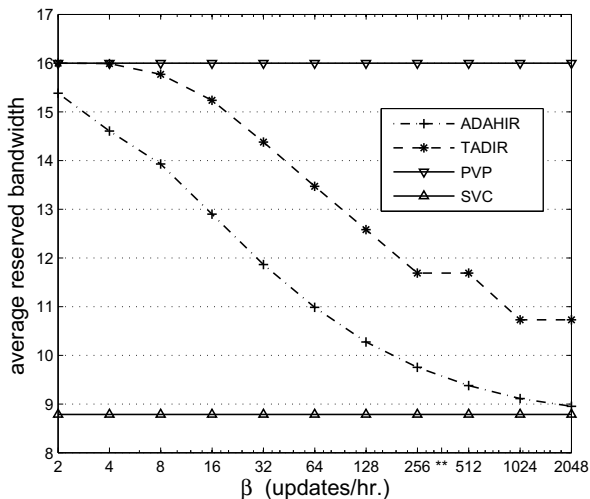


Fig. 4. Average bandwidth reserved by the ADAHIR and TADIR algorithms as a function of β ; average bandwidth consumed by PVP and SVC approaches are included for reference.

obtained blocking probabilities between 0.00996 and 0.0100 and grade of service requirement is met for all values of β . On the other hand, we obtained blocking probabilities between 0.00998 and 0.01004 for TADIR.

Let β_a denote the actual frequency of bandwidth updates in updates/hr. We would like to know whether β and β_a are close. We are also interested in quantifying the gain attainable in using ADAHIR with respect to the PVP approach. For this purpose, let G denote the ratio of the absolute difference between C_m and the average reserved bandwidth using ADAHIR to C_m . Our results on β_a and G are tabulated in Table I. For low values of β , i.e., $\beta \leq 64$, we observe that the desired update rate is attained in the simulations. We also note that the frequency of events (arrival and departure) in an hour is denoted by $\beta_{SVC} = 2\lambda * 3600$ which equals 355.0 for this example. Since bandwidth updates can only occur at event epochs, β_a should be less than β_{SVC} irrespective of β . On the other hand, when $\beta = 355.0$ (this point is marked with a ** on Fig. 4), the ADAHIR approach only uses a portion of the available credits, i.e., $\beta_a = 252.7$. In this case, the reserved bandwidth turns out to be 9.566 compared with the SVC bandwidth 8.785. The gap between the two, i.e., %8 normalized difference, is the price we pay for the traffic uncertainty when using ADAHIR. It is also possible to see using Table I that more than %25 gains are achievable even for the case of an order of magnitude reduction in signaling load.

For the same example, we fix $\beta = 60$ updates/hr. and vary the bucket size B_m which was fixed to 10 previously. The results are given in Table II. We observe that the average reserved bandwidth decreases with increasing B_m since when B_m is small, update credits can be wasted. On the other hand, when B_m is large, too many back-to-back bandwidth updates

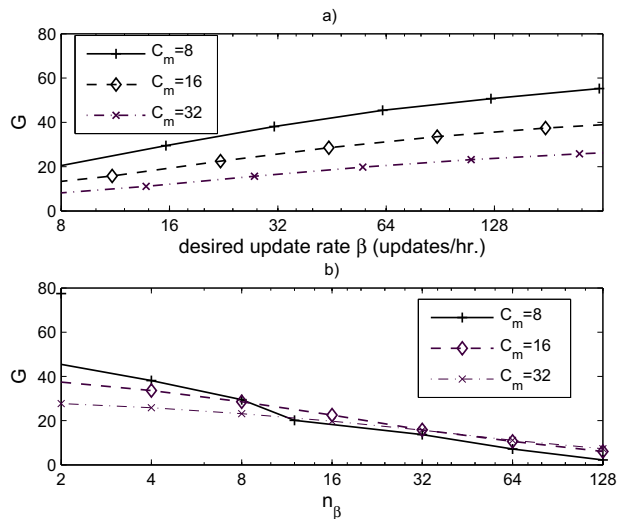


Fig. 5. The gain G in using ADAHIR wrt the PVP approach as a function of a) β b) n_β , for three different values of C_m

would take place. In this paper, we fix B_m to 16 as a balanced choice and the results appear to be fairly insensitive to the particular choice of B_m in this regime.

In the next example, we study the effect of C_m on the performance of ADAHIR. We use the same value for μ but study three choices for the call arrival rate λ corresponding to three choices for the maximum capacity $C_m = 8, 16, 32$. The gain G in using ADAHIR with respect to the PVP approach is plotted with respect to the desired update rate β in Fig. 5a. It is clear that the gains are more significant for systems with low capacity, e.g., $C_m = 8$ for the same desired update rate. We then introduce a ratio n_β defined as $n_\beta = \beta_{SVC}/\beta$. Note that n_β characterizes the reduction in signaling load with respect to the SVC approach. In Fig. 5b, we plot the gain G this time as a function of n_β . For low values of n_β , the gain is again more significant with decreasing C_m . However, this gain (although being relatively small) is more significant for larger values of C_m when n_β is relatively large.

V. CONCLUSION

In this paper, we propose an adaptive hysteresis-based algorithm for dynamic bandwidth reservations. We assume that the frequency of bandwidth updates should be less than a desired update rate which makes the algorithm suitable for a system sensitive to signaling load. The proposed algorithm is model-free and can operate in an environment with unpredictable and non-stationary traffic patterns. We tested the algorithm for a system with Poisson arrivals and exponential call holding times against a model-based synchronous bandwidth management algorithm that relies on the transient solution of continuous-time Markov chains. In the examples we study, we show that the proposed method outperforms the synchronous approach for all values of the desired update rate. Due to its simplicity, this proposed algorithm is a viable alternative to

β (updates/hr.)	4	32	64	128	256	355.0	512	1024	2048
β_a (updates/hr.)	4.0	32.0	63.9	124.4	211.7	252.7	290.5	332.6	347.4
G	8.72	25.85	31.35	35.79	39.04	40.21	41.39	43.05	44.04

TABLE II
THE AVERAGE RESERVED BANDWIDTH AS A FUNCTION OF B_m .

B_m	1	2	4	8	16	32	64	128	256
avg. bandwidth	12.410	12.271	11.414	11.107	11.061	11.085	11.095	11.098	11.096

existing dynamic bandwidth reservation approaches, e.g., auto-bandwidth allocator, for next generation networks. Future work will consist of tuning and testing the proposed algorithm in a non-stationary environment.

ACKNOWLEDGMENT

This work is supported in part by The Science and Research Council of Turkey (Tübitak) under project No. EEEAG-106E046.

REFERENCES

- [1] M. de Prycker, *Asynchronous Transfer Mode, Solution for Broadband ISDN*, 3rd ed. Prentice Hall, 1995.
- [2] B. Davie and Y. Rekhter, *MPLS: Technology and Applications*. Morgan Kaufmann Publishers, 2000.
- [3] F. Baker, C. Iturralde, F. L. Faucheur, and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations," RFC 3175, Sept. 2001.
- [4] L. Kleinrock, *Queueing Systems, Vol. 1, Theory*. John Wiley, New York, 1975.
- [5] U. Mocci, P. Pannunzi, and C. Scoglio, "Adaptive capacity management of virtual path network," in *IEEE Globecom*, London, 1996, pp. 750–754.
- [6] J. T. Virtamo and S. Aalto, "Calculation of time-dependent blocking probabilities," *Proceedings of the ITC Sponsored St. Petersburg Regional International Teletraffic Seminar: Teletraffic Theory as a Base for QoS: Monitoring, Evaluation, Decisions*, pp. 365–375, June 1-7 1998.
- [7] Cisco, "Cisco MPLS autobandwidth allocator for MPLS traffic engineering: A unique new feature of Cisco IOS software," 2001.
- [8] T. Anjali, C. Scoglio, and G. Uhl, "A new scheme for traffic estimation and resource allocation for bandwidth brokers," *Comput. Networks*, vol. 41, no. 6, pp. 761–777, 2003.
- [9] B. Krithikaivasan, K. Deka, and D. Medhi, "Adaptive bandwidth provisioning envelope based on discrete temporal network measurements," in *IEEE INFOCOM*, Hong Kong, 2004, pp. 1786–1796.
- [10] S. Ohta and K. Sato, "Dynamic bandwidth control of the virtual path in an asynchronous transfer mode network," *IEEE Trans. Commun.*, vol. 40, no. 7, pp. 1239–1247, 1992.
- [11] A. Orda, G. Pacifici, and D. E. Pendarakis, "An adaptive virtual path allocation policy for broadband networks," in *INFOCOM (1)*, 1996, pp. 329–336.
- [12] H. Levy, T. Mendelson, and G. Goren, "Dynamic allocation of resources to virtual path agents," *IEEE/ACM Trans. Networking*, vol. 12, no. 4, pp. 746–758, 2004.
- [13] N. Argiriou and L. Georgiadis, "Channel sharing by rate-adaptive streaming applications," *Perform. Eval.*, vol. 55, no. 3-4, pp. 211–229, 2004.
- [14] T. Anjali, C. Scoglio, J. C. de Oliveira, I. F. Akyildiz, and G. Uhl, "Optimal policy for label switched path setup in MPLS networks," *Comput. Networks*, vol. 39, no. 2, pp. 165–183, 2002.
- [15] A. Reibman and K. S. Trivedi, "Numerical transient analysis of Markov models," *Comput. Oper. Res.*, vol. 15, no. 1, pp. 19–36, 1988.