

A Delay Pricing Scheme for Real-Time Delivery in Deadline-Based Networks

Yanni Ellen Liu and Xiao Huan Liu

Department of Computer Science, University of Manitoba,
Winnipeg, MB R3T 2N2, Canada

Abstract. We introduce a novel delay pricing and charging scheme in deadline-based networks to support real-time data delivery. We study the delay performance observed by individual users when different levels of deadline urgency are specified and take a market-based approach to determining a delay price. Our pricing and charging scheme can be used to prevent greedy users from gaining an advantage by specifying arbitrarily urgent deadlines, and can also aid in network load control when equipped with user budget constraints.

1 Introduction

Current and future computer networks are expected to accommodate an increasing number of real-time applications. These applications may require timely delivery of real-time data. Example real-time data include stock quote updates, bids in an online auction, state update messages in distributed multi-player interactive games, audio and video data in video conferences, and voice data in IP telephony. To ensure timely delivery of real-time data, quality of service (QoS) support at the transport network is required.

Deadline-based network resource management [12, 6] is a framework that has been developed to support real-time data delivery. In this framework, the notion of application data unit (ADU) is used. An ADU may correspond to a file or a frame in audio or video transport. Each real-time ADU is associated with a delivery deadline, which is provided by the sending application. It represents the time at which the ADU should be delivered at the receiver. The ADU deadlines are mapped to packet deadlines at the network layer, which are carried by packets and used by routers for channel scheduling. Deadline-based channel scheduling algorithms are employed inside networks; packets with more urgent deadlines are transmitted first. It has been shown that deadline-based scheduling achieves superior performance to FCFS (First-Come First-Served) with respect to the percentage of ADUs that are delivered on time [7, 6].

In deadline-based scheduling, the delay performance experienced by real-time packets is largely affected by the deadline information that they carry, which depends on the ADU deadlines provided by sending applications. If one is free to specify the ADU deadline, a sender may try to gain an advantage by using arbitrarily tight deadlines. This raises the issue of fairness as seen by network

users. In this paper, we first study the impact of deadline urgency on the delay performance experienced by real-time data. Our experiments show that by specifying more urgent deadlines, a user can receive better service in terms of the end-to-end response time. Therefore without a control mechanism in place, a greedy user may obtain good service quality by specifying very tight deadlines. Besides deadline urgency, the delay performance in deadline-based networks is also affected by the load conditions along the ADU path. When the load is light, the delay performance is good. When the load is heavy, congestion may occur; queues at bottleneck links may grow significantly, the delay performance deteriorates.

To prevent greedy users from specifying arbitrarily urgent deadlines, and to control the level of load in order to maintain good delay performance, we develop a novel delay pricing and charging scheme that takes into account both deadline urgency and network load conditions. At each network channel, a *delay price* is periodically computed based on the traffic deadline urgency and the traffic load so that (i) the higher the level of deadline urgency, the higher the price, and (ii) the heavier the network load, the higher the price. Each passing-by packet is charged based on the delay it experiences at this channel and the current channel price: the lower the delay it experiences, the higher the charge; the higher the current channel price, the higher the charge. This charge is carried by the packet and is accumulated along the entire packet path. Depends on the size of network maximum transfer unit (MTU), an ADU may be fragmented into multiple packets for transmission. If an ADU is delivered to the receiver on-time, the ADU is charged based on the packet charges of all its packets. In determining the channel price, a market-based approach from the field of microeconomics is taken. At each channel, the demand is derived from the deadline information carried by real-time packets, and the supply reflects the amount of time that is needed to service these packets. Such a delay pricing scheme encourages users to submit deadline requirements that best match their needs and capacity. Given limited user budget for network transmissions, such a delay pricing and charging scheme may aid in the process of load control so that performance degradation due to congestion can be alleviated. We present our pricing and charging scheme, and evaluate its performance by simulation.

This paper is organized as follows. In Section 2, the delay performance in deadline-based networks when there are no pricing and charging schemes in place is studied. The delay pricing and charging scheme that we developed is presented in Section 3. Simulation results on its performance are reported in Section 4. In Section 5, we review the related literature. Finally, in Section 6, we conclude our work and suggest some topics for future research.

2 Deadline-Based Data Delivery

In this section, we study the impact of ADU deadlines on the delay performance experienced. In previous studies on deadline-based networks, only the aggregated performance of all traffic that is transmitted over the network was studied. In

this work, we study the performance observed by individual ADUs and individual users. The different delay performance obtained will incur different cost to senders after we introduce our pricing and charging scheme.

We first describe our performance model. At a sender, each generated ADU is characterized by: size, source and destination addresses, deadline, and arrival time. For simplicity, only real-time ADUs are considered. The support to best-effort traffic will be discussed in Section 3.3. Segmentation of an ADU into packets is performed at the sender before the packets are admitted to the network. The maximum packet size at the network layer is 1500 bytes. Packets are routed through the network until they reach their destination node. They are then delivered to the receiver where packet re-assembly is performed. We assume that fixed shortest-path routing is used and there are no transmission errors. For simplicity, the processing times at the sender and the receiver are not included in our model, and each packet carries the deadline of the ADU to which it belongs.

The deadline-based channel scheduling algorithm implemented is the T/H- $p(m)$ algorithm [6]. T stands for the time left (or packet deadline - current time) and H is the number of remaining hops to destination. The value T/H is calculated when a packet arrives at a router, it can be viewed as the urgency of a packet; specifically, a packet with a smaller T/H means that it is more urgent. At each scheduler, there are m queues, namely Q1, Q2, ..., Q m for real-time traffic and one queue for best-effort traffic. The T/H values of packets in Q1 are the smallest among all queues, followed by Q2 which has the next smallest, then Q3, Q4, until Q m . The fraction of real-time packets that are sent to Q i is $1/m$, $i = 1..m$. Head-of-the-line priority is used to serve packets in these queues, including the best-effort queue, which has lower priority than Q m . Let T_f denote the sum of the packet transmission time and the propagation delay on the current channel. If a real-time packet is already late ($T < T_f$) upon arrival at a router, the packet is downgraded to best-effort. In our experiments, m of 4 is used.

For a real-time ADU, the delivery deadline is modeled as follows. Let x be the end-to-end latency when there is no queuing and no segmentation. Also let x_p be the end-to-end propagation delay, y the size of the ADU, and c_j the capacity of the j -th channel along the path based on shortest-path routing. Then x can be estimated by $x = x_p + \sum_j y/c_j$. The allowable delay is assumed to be proportional to x . Hence, the delivery deadline for the ADU is given by $d = arrival\ time + kx$, where k is referred to as a *deadline parameter* ($k > 1$). In general, a smaller k means that the ADU has a more urgent deadline.

A 13-node network model is used in our simulation. Its topology is depicted in Figure 1. The capacity of each channel is assumed to be 155 Mbit/sec. The value shown on each link is the distance in miles. This is used to determine the propagation delay. For each arriving ADU, the source and destination nodes are selected at random. The ADU interarrival time is assumed to be exponentially distributed, and the average ADU arrival rate is λ (in number of ADUs per second). The size of each ADU is assumed to belong to one of two ranges: [500, 1500], and [1500, 500000], in bytes. The first range reflects the sizes of small

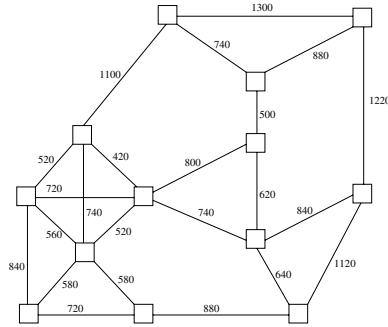


Fig. 1. Network model

ADUs, *i.e.*, one packet per ADU. The proportion of small ADUs is kept at 25%. ADU size is assumed to be uniformly distributed within each of these two ranges. At each scheduler, we assume that a finite buffer is in place. Modern routers usually have large buffer sizes that can accommodate between 100 and 200 ms’ worth of data with respect to link capacity [2]. We will consider buffer sizes within this range. For simplicity, we assume that packets dropped due to buffer overflow are not re-transmitted. The performance measure of interest is the end-to-end response time.

Two experiments were carried out. The first experiment is to compare the performance of two benchmark ADUs that have the same size, arrival time, sender and receiver, but have different deadlines. The following model parameters are used for background traffic in this experiment. We choose 1200 ADUs/sec as the ADU arrival rate λ . This corresponds to 90% utilization on the bottleneck. At this level, the delay difference between two ADUs with different deadline urgency is larger, thus is more obvious, than at a lower load level. The deadline parameter k was assumed to be $1 + \tilde{e}$ where \tilde{e} is exponentially distributed with mean 0.4. With this model, a variety of deadline urgency can be represented. For each outgoing channel, the total buffer size was assumed to be 3.1MByte. This corresponds to 160 ms’ worth of data with respect to the link capacity. The sizes of the two benchmark ADUs are 5000 bytes. The end-to-end latency x in this case is 33 ms. Let end-to-end deadline denote the time period between when an ADU is submitted by a sending application for transmission and the ADU deadline. The end-to-end deadlines for the two ADUs are chosen to be 100 and 500 ms respectively. Thus one deadline is more urgent than the other. The response times obtained by these two ADUs are shown in Table 1. Ignore the

Table 1. Two ADUs with different deadlines

ADU	End-to-end deadline (ms)	Response time (ms)	ADU charge (cu)
1	100	39.57	1.50
2	500	120.84	0.14

Table 2. A flow when with different levels of deadline urgency

Deadline parameter	Mean response time (ms)	Mean ADU charge (cu)
$k = 1.4$	80.3	353.52
$k = 1.5$	86.4	306.30

last column for now, it can be observed that the ADU with the more urgent deadline has lower response time.

In the second experiment, we compare the performance of a benchmark flow (flow 0) when its level of deadline urgency is varied. In our experiments, a flow denotes a sequence of ADUs that are sent between a given sender and a given receiver. All background traffic is the same as in the first experiment, except that the value of \bar{e} is 0.5 for the deadline parameter k . The deadline parameter for Flow 0's ADUs is varied from 1.4 to 1.5, thus we can compare the delay performance when flow 0's deadline is more urgent than the average deadline of background traffic with the delay performance when flow 0's deadline has the same urgency as the average deadline of background traffic. Flow 0's mean response time performance is shown in Table 2. It can be observed that when with identical background traffic, flow 0 is able to achieve lower average response time when its ADU deadlines are more urgent. We conclude that in deadline-based networks, the delay performance largely depends on the deadline urgency. When competing with the same background traffic, an ADU or a flow of ADUs can raise their service priority, thus obtaining better delay performance by using more urgent deadlines. An important objective of our pricing and charging scheme is to prevent such greedy behaviours.

3 Delay Pricing in Deadline-Based Networks

In this section, we present the channel delay pricing and the packet and ADU charging scheme that we have developed. Some implementation issues are then discussed.

3.1 A Market-Based Approach to Delay Pricing

In deadline-based networks, each packet carries a deadline, which specifies the requirement on its delay performance. At each hop, the T/H value calculated indicates the delay requirement of this packet on this channel; namely, if the response time at this hop is less than or equal to T/H, and if every hop along the packet path manages to achieve so, then the packet will arrive at the receiver on-time. From an economic point of view, the finite capacity and the transmission service at each channel is the scarce resource sought by real-time packets. The packet T/H values reflect the *demand* on the resource; and the time it takes to service a set of packets signifies the capability, *i.e.*, the *supply* available at the channel. The goal at each channel is to utilize a pricing mechanism to urge the

adjustment of demand so that the difference between the supply and the demand can be kept minimal.

We take a market-based approach from the field of microeconomics, and determine a market price, called *channel delay price*, at each channel based on the relation between the demand and the supply. An iterative tatonnement process [11] is used. The channel delay price is updated every *price update interval*. During each update interval, for each departing packet, the following information is recorded and accumulated: (i) the packet T/H value, and (ii) the packet response time. The packet response time is defined as the sum of the queueing delay, the packet transmission time, and the channel propagation delay. Let D^T denote the total T/H value of all departing packets, and S^T be the total packet response time of all departing packets. At the end of the update interval n , the channel delay price p for the update interval $n + 1$ is defined as:

$$p_{n+1} = \{p_n + \sigma * (D - S)/S, 0\}^+ \quad (1)$$

where $D = 1/D^T$, and $S = 1/S^T$. σ is an *adjustment factor*, which can be used to trigger faster or slower responses of the channel price to the amount $D - S$. At system initialization, p_0 is set to zero. In addition, only positive channel delay prices are defined.

It should be noted that the channel price is higher (i) when the deadline urgency is higher, *i.e.*, when D^T is lower; and (ii) when the load is heavier, *i.e.*, when S^T is higher. We assume that the channel prices can be made available to network users. In response to the changes of the channel delay price, adaptive users with budget constraints may adjust their requirements in terms of the deadline urgency and the offered load. The end result is that at the channel with the heaviest load, the resource demand can be driven towards the amount of supply, and at every other channel, the demand is no greater than the supply. Under these conditions, good service quality can be achieved.

3.2 Calculation of Packet and ADU Charges

Using the channel delay pricing scheme presented above, we describe a method to calculate packet and ADU charges. Note that in this work, we focus on devising a *delay charging* scheme that aims at two objectives: (1) to provide an incentive for users to submit requests with the QoS requirement that best matches their need, and (2) to control network load so that good delay performance can be maintained. In general, network charging schemes usually contain certain charges in order to assure the return on investment; these charges may cover the cost for constructing, maintaining, and upgrading the network. In this paper, however, we do not consider these charges and focus on the delay charge only.

A per-packet per-channel charging scheme is developed in our framework. At each channel, upon each packet departure, the packet response time d_a is calculated: the queueing delay can be obtained by subtracting the packet arrival time from the current time, the transmission delay can be computed using the packet size and the channel capacity, the propagation delay is fixed and given.

Let p be the current channel delay price. The packet charge g at this channel is defined as: $g = p/d_a$. Define a new packet header field called “accumulated charge”. It keeps track of the total delay charge incurred by this packet at all channels along its path. If a packet arrives at the receiver on-time, the value of this field is retrieved and is taken as the packet charge. If an ADU is delivered on-time, its ADU charge is defined as the sum of all its packets’ charges. Late packets and ADUs are not charged.

3.3 Network Layer Issues

In our pricing scheme, the channel delay prices are updated periodically at constant time intervals. This can be easily implemented using either hardware or software timer interrupts. In general, the length of the update interval should not be too short, this way a good number of T/H value and response time samples can be collected to estimate the current resource demand and supply. A length that is much longer than the average packet transmission time should be used. The update interval should not be too long either, in this way the short-term traffic conditions can be accounted for.

Our pricing and charging scheme introduces some processing overhead inside routers. This includes packet response time calculation, and accumulation of the T/H values and the packet response times for all departing packets. However, because none of these operations depends on the queue size, we consider this overhead to be in-expensive in terms of implementation. The computation of delay prices only occurs once every update interval, which is much longer than the mean packet transmission time, therefore is not considered costly either. The “accumulated charge” header field can also be easily added using packet header options or similar mechanisms.

The social fairness aspect of a pricing scheme is concerned with whether some users will be prevented from accessing the network only because of their inability to pay [3]. In our discussion so far, we have assumed that there is only real-time traffic in the network. In fact, best-effort traffic can easily be accommodated in our framework. All best-effort traffic can carry a deadline of infinity. At each outgoing channel inside the network, a certain amount of bandwidth can be allocated to best-effort traffic only. This can be implemented using a fair-queueing algorithm with two classes. The deadline-based scheduling is used only within the real-time class. The best-effort class can use a low flat-rate pricing and charging scheme. Those users who can not afford the delay charges of the real-time class can use the bandwidth that is allocated to the best-effort class.

4 Performance Evaluation

In this section, we evaluate the performance of our pricing and charging scheme by simulation. We used the performance model that is described in Section 2 and added our pricing and charging scheme implementation. The following values are chosen for algorithm parameters. The adjustment factor σ in Eq.(1) is set

to 0.06. The price update interval is 2 seconds' long. The simulation is run for 50 seconds. Corresponding to the two objectives of our pricing and charging scheme, we discuss two cases: differential charges based on deadline urgency, and price-based load control.

4.1 Deadline Urgency Differential Charges

In section 2, we have shown that it is possible for a user to gain higher service priority in deadline-based networks by specifying very tight deadlines. Our experiments show that this holds for both individual ADUs and for a flow of ADUs. Our solution to prevent such greedy behaviors is to introduce an ADU delay charge. In Tables 1 and 2, the last columns indicate the ADU charges in a *charge unit* (cu) using our pricing and charging scheme. In this paper, we do not associate the charge unit with any concrete monetary value, and leave this choice to network operators. It can be observed that when all traffic attributes but the deadline urgency are the same, the more urgent ADUs are charged more using our scheme. The absolute charge values depend on channel prices and packet response times along the path. We conclude that our scheme can be used to enforce differential charges based on ADU deadline urgency.

4.2 Price-Based Load Control

Our pricing and charging scheme may also aid in load control. This can be accomplished through a *delay pricing agent*. This agent is located between the users and the network. When a user submit a real-time ADU for transmission, this agent may utilize the current price information along the ADU path and the ADU deadline requirement to provide a *charge estimate* for this ADU. If the network is heavily loaded, the delay prices inside networks would be high, which may result in a high charge estimate. In this case, a user may choose not to submit the ADU for transmission until the price drops. In our simulation, we used a simple elasticity model to represent such user adaptation behavior. We assume that each sender has a fixed amount of budget for every price update interval. When an ADU is generated for transmission, the total allowable end-to-end delay is equally allocated to each hop, and the current highest channel price along the ADU path is used to compute an estimated per-hop packet charge. Cumulating the total number of hops along the path and the total number of packets in this ADU, an estimated ADU charge is obtained. If a sender has enough available budget to cover this charge, then the ADU is sent and the budget is decremented by this charge. Otherwise the ADU is refrained from entering the network.

In Figure 2, we plot the price dynamics at one bottleneck inside the network when without and with the user budget constraint. There are four graphs in this figure. The two on the left are ones when there is no budget constraint. The two on the right are ones when there is limited user budget. In our simulation, the value of budget is assumed to be 150000. The four curves in each graph are for four load levels. It can be observed that regardless of load, when with limited user budget, the price can be regulated to a fairly steady value. This is

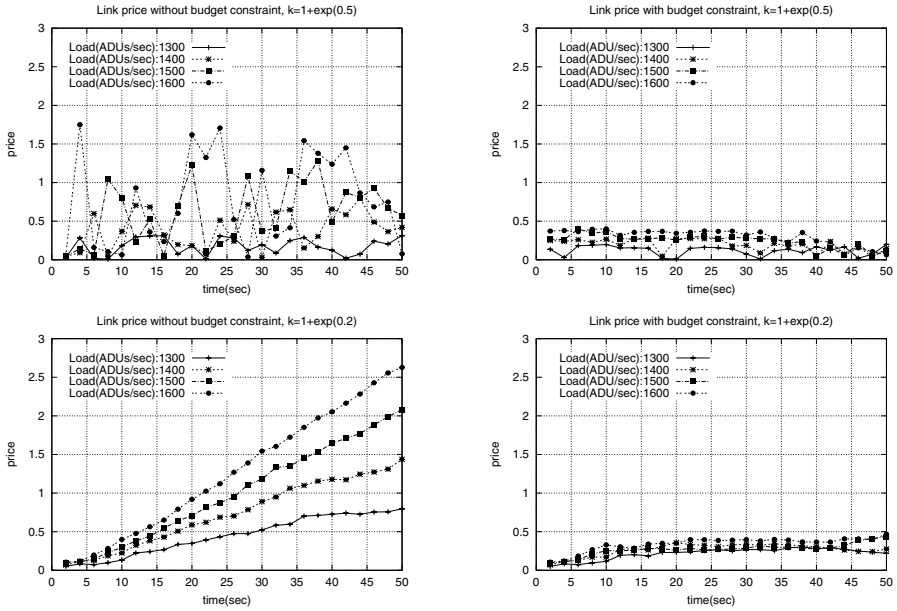


Fig. 2. Prices when without and with the budget constraint

because limited budget can limit the amount of load that enters the network. When demand and supply are approximately equal, the price becomes steady. This demonstrates the effectiveness of load control of our scheme when coupled with user budget constraints.

The difference between the top two and the bottom two graphs lies in the deadline urgency used. When deadline is less tight (see the top left graph where the mean deadline parameter is 1.5), although there is fluctuation, the demand is about equal to the supply, so the prices do not increase monotonically. When

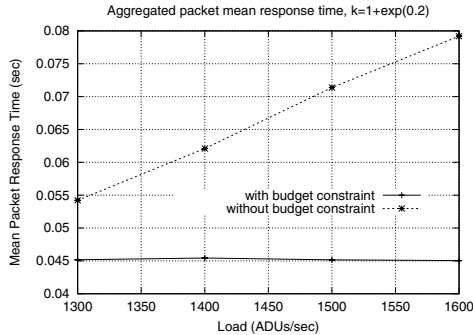


Fig. 3. Response time when with and without the budget constraint

deadline is tight (see the bottom left graph where the mean deadline parameter is 1.2), because there is no budget constraint to control the load, there is a clear mismatch between demand and supply. The ever increasing prices indicate that the supply does not keep up with the demand. Thus certain load control is needed.

Because of the effectiveness of load control of our scheme when with user budget constraints, the network delay performance can be significantly improved. In Figure 3, we plot the average response time when with and without the budget constraint. The average deadline parameter is 1.2. It can be observed that as the load increases, when without pricing and the budget constraint, the response time keeps increasing. When with pricing and the budget constraint, the response time can be kept very low regardless of the level of the offered load. We conclude that our pricing and charging scheme is effective in network load control when coupled with user budget constraints.

5 Related Work

Network pricing has been a popular subject of research. Except flat rate pricing, among dynamic pricing schemes, there are Paris Metro Pricing [9], priority-based pricing [1, 5], smart market pricing [8], competitive market pricing [4], DiffServ pricing and RNAP [11, 10]. Similar to some of above studies, in our study, we adopt a market-based approach to determining the channel price. However, differ from all above studies where pricing of bandwidth is concerned, in our work, we introduce the novel delay pricing concept. This is made available by the deadline-based framework in which each packet carries its delay requirement. The demand can easily be derived from this deadline information.

6 Conclusion

We have developed a novel delay pricing and charging scheme in deadline-based networks to support real-time data delivery. In our scheme, we make use of the concept of competitive market and determine a delay price based on delay demand and supply at each channel. Simulation results show that our scheme can incur different charges to users with different QoS requirements, and can aid in effective load control when user budget constraints are available. There are a number of interesting future work of this study, including the design of more sophisticated ADU charge estimation schemes, and the investigation of strategies to maximize the user utility.

Acknowledgment

This research was supported by the University of Manitoba, Canada, under the University Research Grant Program, and by the Natural Sciences and Engineering Research Council of Canada.

References

1. R. Cocchi, S. Shenker, D. Estrin, and L. Zhang. Pricing in computer networks: motivation, formulation, and example. *IEEE/ACM Trans. Netw.*, 1(6):614–627, 1993.
2. Internet end-to-end interest mailing list. Queue size of routers. <http://www.postel.org/pipermail/end2end-interest/2003-January/>.
3. M. Falkner, M. Devetsikiotis, and I. Lambadaris. An overview of pricing concepts for broadband ip networks. *IEEE Communications Surveys and Tutorials*, 3(2), 2000.
4. E. W. Fulp and D. S. Reeves. The fairness and utility of pricing network resources using competitive markets. *Computer Networks*, Feb. 2000. Submitted.
5. Alok Gupta, Dale O. Stahl, and Andrew B. Whinston. Priority pricing of integrated services networks. In L.W. McKnight and J. P. Bailey, editors, *Internet Economics*, pages 323–52, Cambridge, Massachusetts, 1997. MIT Press.
6. Y. E. Liu. *Deadline based network resource management*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2003.
7. Y. E. Liu and J. W. Wong. Deadline based channel scheduling. In *Proc. of the IEEE Globecom'2001*, pages 2358–2362, San Antonio, Texas, November 2001.
8. J. K. Mackie-Mason and H. R. Varian. Pricing the Internet. In *Int'l Conf. Telecommunication Systems Modelling*, pages 378–93, Nashville, TN, USA, Mar. 1994. available from URL <http://www.spp.umich.edu/papers/listing.html>.
9. Andrew Odlyzko. A modest proposal for preventing internet congestion. Technical report, Sept. 1997. available at <http://citeseer.ist.psu.edu/odlyzko97modest.html>.
10. X. Wang and H. Schulzrinne. Rnap: A resource negotiation and pricing protocol. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'99)*, pages 77–93, Basking Ridge, New Jersey, Jun. 1999.
11. X. Wang and H. Schulzrinne. Pricing network resources for adaptive applications in a differentiated services network. In *Proceeding of INFOCOM'2001*, Anchorage, Alaska, Apr. 2001.
12. J. W. Wong and Y. E. Liu. Deadline based network resource management. In *Proc. of ICCCN'2000*, pages 264–268, 2000.