

# Efficient adaptation to dynamic pricing communicated by ECN marks: Scenarios for experimental assessment

Panayotis Antoniadis<sup>a</sup>, Costas Courcoubetis<sup>a,b</sup>, George Margetis<sup>b,c</sup>,  
Vasilios A. Siris<sup>b,♦</sup>, and George D. Stamoulis<sup>b,c</sup>

<sup>a</sup>Department of Informatics, Athens University of Economics and Business, Greece

<sup>b</sup>Institute of Compute Science (ICS), Foundation for Research and Technology - Hellas (FORTH)  
P.O. Box 1385, GR 711 10, Heraklion, Crete, Greece

<sup>c</sup>Department of Computer Science, University of Crete, Heraklion, Greece

## ABSTRACT

We present our research activities and initial results, regarding both simulation and actual test-bed implementation, that aim to demonstrate the feasibility and to evaluate the advantages and performance of a market-based management scheme based on a simple feedback mechanism that informs users of the congestion their traffic is experiencing. The feedback is based on Explicit Congestion Notification (ECN) marking. The aforementioned objectives are investigated using two service provisioning scenarios. The first scenario involves an application provider that offers discrete Quality of Service (QoS) classes at different prices, whereas the second scenario offers a wider range of QoS, price pairs. The experiments we report refer to the implementation of a modified version of the TCP algorithm, which provides service differentiation based on a sender's willing-to-pay, and of a packet marking scheme that provides early warnings of incipient congestion.

**Keywords:** congestion pricing, flow control, TCP, explicit congestion notification, packet marking

## 1. INTRODUCTION

The current Internet lacks mechanisms that allow efficient and stable network operation and healthy growth in order to meet increasing demand. A new field of research has recently emerged<sup>7,9,10</sup>, investigating how these objectives can be met by a network with a simple feedback mechanism that informs users of the congestion cost (shadow price) their traffic is incurring. Such feedback signals can be communicated using Explicit Congestion Notification (ECN) marking, which has been proposed for the Internet<sup>15</sup>. By coupling the above mechanism with a small fixed price per mark, users will have both the necessary information and the incentive to respond to the congestion signals in a way that guides the system (network and users) to efficient and stable operation. This is due to the fact that users are free to respond to congestion signals in any manner that suits their needs and objectives. This contrasts the current Internet, whose health depends to a large extent on all end-systems using a similar form of congestion adaptation, as that performed by TCP.

The overall objective of our research is to demonstrate the feasibility and evaluate the advantages and performance of the above market-based scheme in both a simulated environment and an actual test-bed implementation. The feasibility and performance of the above scheme depend on the following two functions that are implemented at the end-systems:

- at a low level (reaction in a fast time-scale), the adaptation of the traffic rate based on the ECN marks received by the network and
- at a higher level (reaction in a slow time-scale), the guidance of the rate adaptation function to achieve some objective, which can be expressed by a utility function.

We will refer to the first function as rate control and to the second function as dynamic price handling. The rate control function is performed at the sender-side and is similar to the functionality of the TCP algorithm. Indeed, it has been suggested<sup>7,4</sup> that relatively simple changes to the TCP algorithm can enable it to offer different performance to different users. This can be achieved by adapting the congestion window based on the price paid by each user. Thus, (price-based)

---

♦ Correspondence: e-mail: [vsiris@ics.forth.gr](mailto:vsiris@ics.forth.gr), tel.: +30 81 391726, fax: +30 81 391601

rate adaptation can run autonomously, and is only fine-tuned (from time to time) by the dynamic price handler. The latter function can be performed by an intelligent agent, acting on behalf of the user, and is typically located at the paying party side. Nevertheless, as we latter discuss, in the case where the end-customer is the paying party, it can be possible (and even desirable) to move some of the intelligence to the provider-side. Among our goals are to investigate rate control procedures and procedures for intelligent agents implementing various strategies. The ability of such procedures to effectively adapt to dynamic prices will ensure the feasibility and applicability of the proposed market-based scheme from the user's perspective.

In addition to the two functions, rate control and dynamic price handling, that are implemented at the end-systems, another component that affects the behavior and performance of the proposed market-based approach is the packet marking algorithm implemented in the routers. This algorithm is responsible for detecting congestion, and subsequently marking packets to signal this congestion to the end-systems.

The previous objectives will be investigated using two service provisioning scenarios. In the first scenario, an application provider offers discrete Quality of Service - QoS<sup>1</sup> classes (e.g. "Gold", "Silver") for each type of service it provides (e.g. web browsing, streaming video/audio). The QoS class is selected by the human user at the connection set-up phase, and can be changed at any intermediate point during service delivery. In the second scenario, the provider offers a more elaborate service where the user can have further control of the QoS being delivered. In this scenario, there is more intelligence at the user side; this intelligence is incorporated in the dynamic price handler.

In this paper we present initial experimental results, regarding both simulation and actual test-bed implementation, that deal with two of the three components discussed above, namely rate control and packet marking. Regarding the former, we consider a modified version of the TCP algorithm, which provides service differentiation based on a sender's willing-to-pay, and regarding the latter, we consider a packet marking scheme that provides early warnings of incipient congestion.

The rest of this paper is organized as follows. In section 2 we discuss each of the above two scenarios in detail, along with architectural issues of where the rate control and dynamic price handling functions are performed. In section 3 we present the window-based flow control algorithm and packet marking algorithm that we have initially implemented in our simulation and test-bed environment, and in section 4 we present initial results from our experimental investigations. Finally, in section 5 we conclude the paper discussing research directions that we are currently pursuing.

## 2. SERVICE PROVISIONING SCENARIOS

The scenarios we describe next involve three business entities: the end-customer, the application provider, and the Internet connectivity provider. The application provider offers various QoS classes for the services it provides, from which the end-customer (user) can choose. The service is delivered over an Internet connectivity provider, which charges end-customers for transport services using the dynamic pricing scheme described in the introduction. Other business models are possible, e.g. the Internet connectivity provider can charge the application provider, which in turn charges end-customers at a fixed rate. In this case, the dynamic price handler would be located at the application provider side.

### 2.1 Scenario 1: Discrete QoS classes at different prices

The goal of this scenario, depicted in figure 1, is to demonstrate that different QoS is provided when a different price is paid. In particular, we assume that the application provider offers two types of service, e.g. "Gold" and "Silver" charged  $w_g$  and  $w_s < w_g$  monetary units per time unit respectively. The network charges customers by the total number of ECN marks they receive multiplied by a fixed price per mark. The application provider (sender) adapts the traffic rate of each flow so that the flow's charging rate does not exceed the charging rate of the QoS class selected by the user of the flow.

Initially, there is low congestion in the network, and as a result users of both QoS classes experience similar performance. In the case of a web browsing service, performance is measured in terms of throughput, which is inversely proportional to the time for transferring Web pages. Next, congestion is created by inserting background traffic inside the network. As a result, the QoS experienced by a user that selected the "Gold" class will be better compared to the QoS experienced by a user who selected the "Silver" class. In the case of web browsing, users can be offered the capability to switch to another QoS class, using a "QoS Knob" that is located in their web browser. This switching of QoS classes motivates the need for an intelligent software agent to aid the user in coping with dynamically varying QoS. Such an agent has the important role of hiding all the complexities associated with dynamically varying prices and QoS, and can implement more complex strategies than the one of this scenario, which was to maintain a constant charging rate.

---

<sup>1</sup> By Quality of Service (QoS) here we refer primarily to throughput, the reason being that throughput affects the quality or performance of the service received by a user; e.g. higher throughput results in lower transfer delay of web pages or enables encoding of higher resolution in streaming video/audio.

In order to implement the above service, the application provider (sender) requires a rate control module that adjusts the flow's traffic rate so that the rate of received marks is approximately constant, and, when multiplied with the price per mark is equal to the charging rate for the selected QoS class.

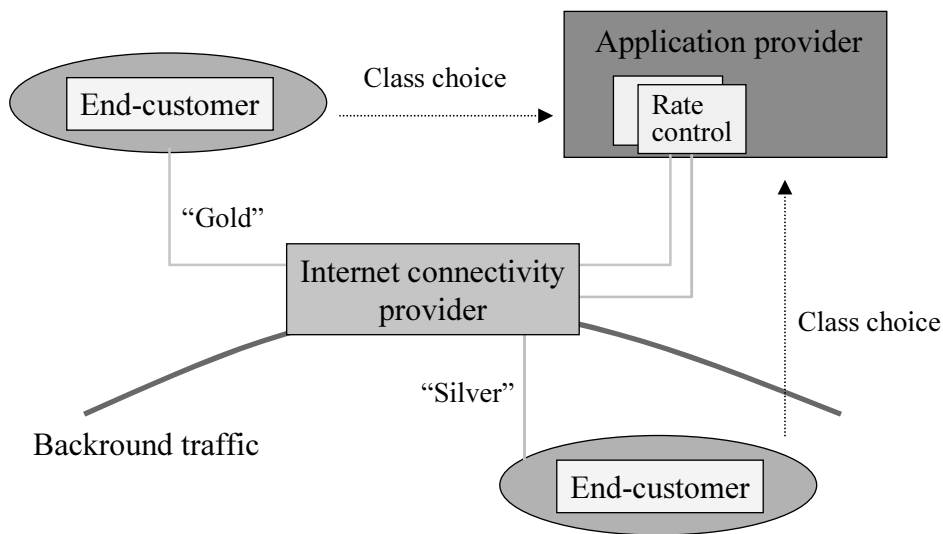


Figure 1. Scenario 1: discrete QoS class service offering

## 2.2 Scenario 2: QoS vs. price trade-off

In the discussion of Scenario 1, we motivated the need and importance for an intelligent software agent residing at the user-end to aid him in selecting the QoS class. The second scenario, depicted in figure 2, will demonstrate how such an agent, which we will refer to as dynamic price handler (DPH), can be useful for the user and investigate procedures for implementing various strategies. First, consider the objective of having constant throughput, while keeping the charging rate below some threshold. In the case of web browsing, a constant throughput results in a constant transfer time for web pages of the same size - the latter being a measure of QoS for a web browsing service. This service can be implemented by modifying the rate control module of Scenario 1 such that the sending rate is kept constant, provided the rate of received marks multiplied by the price per mark is less than a pre-defined threshold  $w$  on the charging rate (expressed in monetary units per time), which expresses the maximum willingness-to-pay of the user. When this threshold is hit, then the sending rate should be kept at a lower level, as long as the rate of marked packets is high.

More elaborate services can be offered, by adjusting the willingness-to-pay parameter  $w$  of the rate control module in a time scale longer than the time scale of rate control. This adaptation can be performed by the dynamic price handler (DPH), where the user's objective would typically be expressed in terms of a utility function. Furthermore, the above scenario can be extended by having the provider offer a wide variety of QoS classes (rather than only two), thus allowing a wide variety of values for the willingness-to-pay.

In the above example, the intelligence for QoS class selection was located in the dynamic price handler that resided at the end-customer. Another possibility is to distribute this intelligence among the end-customer (user) and the application provider: The knowledge of the user's utility will be located at the user agent. At the other side, the provider agent can offer some basic strategies that the user agent can synthesize in order to achieve a particular objective. Hence, referring to the example considered above, the functionality for a service with constant throughput (hence QoS), provided the charging rate was below some threshold, can be implemented entirely at the provider-side. An important advantage of adding more intelligence to the provider (sender) is the possibility of decreasing the communication between the user and the provider.

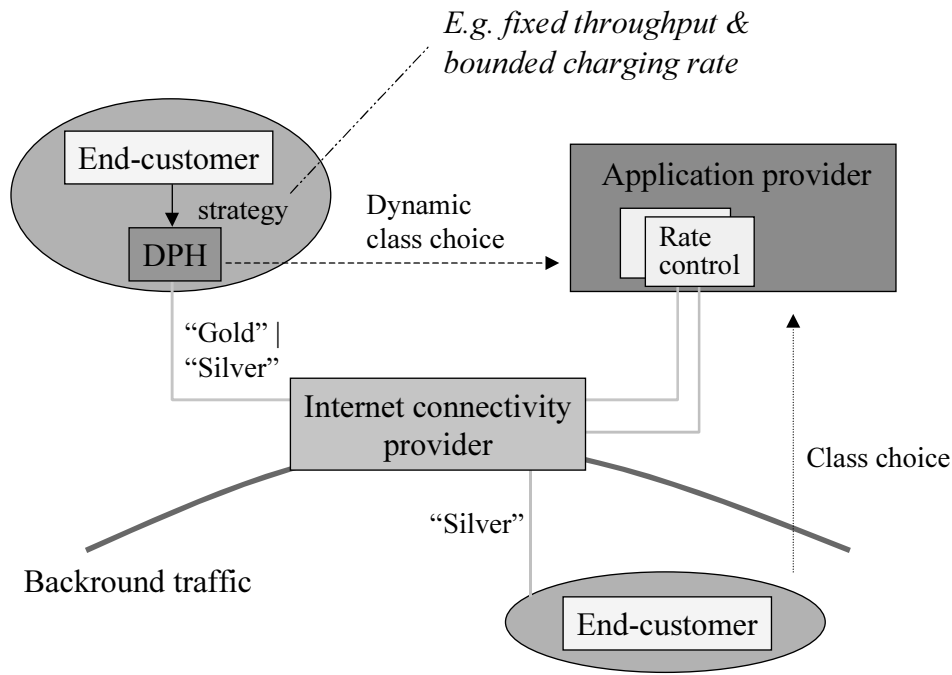


Figure 2. Scenario 2: dynamic QoS class selection

### 3. FLOW CONTROL AND PACKET MARKING

Two important components of the congestion pricing framework described in the previous sections are the rate control algorithm implemented at the end-systems (the end-customers and application provider in figures 1 and 2), and the packet marking algorithm implemented in the routers (which belong to the connectivity provider in figures 1 and 2). In this section, we describe the rate control and packet marking algorithms that we have initially implemented in both our simulation and test-bed environments. Related experimental results will be presented in the next section.

#### 3.1 The TCP/WTP rate control algorithm

The rate control algorithm that we have initially investigated is a simple modification of the TCP algorithm<sup>7</sup>, whereby a sender adjusts his sending rate so that the rate of the received ECN marks multiplied by the price per mark is (approximately) equal to his willingness-to-pay, the latter being expressed as a charge per unit of time.

The TCP (Transmission Control Protocol) is a window-based algorithm where the congestion window,  $cwnd$ , which gives the maximum number of unacknowledged segments that the TCP sender is allowed to have in transit, is adjusted based on the following four algorithms<sup>1</sup>:

- Slow start
- Congestion avoidance
- Fast retransmit
- Fast recovery

The slow start algorithm governs the initial increase of the congestion window at the beginning of data transmission, whereas the fast retransmit/recovery algorithms govern the behavior in cases of packet loss. According to the second algorithm, congestion avoidance<sup>8</sup>, the congestion window is increased by at most one segment per round trip time. An important feature of TCP is self-clocking: acknowledgements from the receiver are used by the sender to prompt a step

forward of the window update algorithm. Hence, since in the absence of congestion the sender receives  $cwnd$  acknowledgements in one round trip time, an increase of one packet per round trip time can be (approximately) achieved by increasing the congestion window by  $1/cwnd$  for each acknowledgment that does not contain an ECN mark. The ECN mark is contained in the ECN echo bit of the acknowledgment<sup>15</sup>. On the other hand, when packet loss is detected or when an acknowledgment containing an ECN mark is received, the congestion window is decreased by half. The modified congestion avoidance algorithm, which we will refer to as TCP/WTP, changes the above behavior so that upon the receipt of an acknowledgment the congestion window is adjusted based on the equation

$$cwnd_+ = \bar{k} \left( \frac{\bar{w}}{cwnd} - f \right), \quad (1)$$

where  $\bar{w}$  is the willingness to pay per round trip time,  $\bar{k}$  is a gain parameter that controls the rate of convergence, and  $f$  equals 1 if the received acknowledgment contains a mark or 0 if it does not contain a mark.

The TCP/WTP algorithm contains two more modifications that are related to the procedures for adding explicit congestion notification to TCP<sup>15</sup>:

- Upon detection of congestion, e.g. with the reception of an acknowledgment with the ECN echo bit set, the TCP sender should reduce the congestion window *by half only once* per round trip time. On the other hand, with the TCP/WTP algorithm the congestion window is adjusted, according to equation (1), *for every* acknowledgment received.
- When the TCP receiver receives an IP packet containing an ECN mark, it sets the ECN echo bit in all the acknowledgements it returns until it is notified by the TCP sender that the congestion window has been decreased. On the other hand, with TCP/WTP the receiver sends one acknowledgment with the ECN echo bit set for each packet received that contains an ECN mark.

Of course, the TCP/WTP algorithm described above represents one approach for modifying the TCP to achieve service differentiation based on the willingness-to-pay. Another alternative, which approximates the above behavior, includes keeping the TCP receiver behavior as that suggested for adding ECN to TCP<sup>15</sup>, and decreasing the congestion window by some quantity at most *once per round trip time*. If this decrease is equal to  $\bar{k}(\bar{w} - cwnd)/2$ , then the algorithm is equivalent to TCP/WTP when half of the acknowledgments in one round trip time contain marks. An important advantage of such an algorithm is that no modifications are required at the receivers, provided that they already support TCP/ECN. Only changes at the senders, i.e. at the application provider side referring to figures 1 and 2, are necessary. Hence, such a scheme lends itself to easier deployment. Other alternatives of TCP-like algorithms are discussed in references 5 and 11.

### 3.2 The virtual buffer marking algorithm

In addition to the sender and receiver flow control algorithms, another important component of the proposed congestion pricing framework includes the packet marking algorithm, which is implemented in the routers. One popular marking algorithm, already implemented in production routers, is RED (Random Early Detection)<sup>6</sup>, used in conjunction with ECN, i.e. packets are marked rather than dropped according to the probability given by the RED algorithm. In addition to containing a number of parameters, whose tuning depends heavily on the traffic characteristics<sup>3</sup>, the marking probability with RED is a function of the queue length, which is a rather ineffective indicator of congestion: for the same queue length, if the queue length is decreasing then there is no congestion, whereas if it is increasing then there is congestion. An alternative packet marking algorithm we have investigated, which presents an early warning of congestion, is the virtual buffer marking algorithm<sup>9</sup>. With virtual buffer marking, congestion is taken to be the loss of packets. The algorithm, for each output link in the router, maintains a virtual buffer having size  $\theta B$  and serviced at rate  $\theta C$ , where  $B$ ,  $C$  are the real buffer and capacity of the actual output link. The algorithm marks all packets that traverse the actual link from the time a loss occurs in the virtual buffer until the first time the virtual buffer becomes empty; this period is called the busy period of the virtual buffer. An important feature of the virtual buffer marking algorithm is that it differentiates flows based on their burstiness (see subsection 4.1). As a result, burstier flows receive a higher marking probability, which is consistent with the fact that such burstier flows are harder for the router to multiplex.

## 4. SIMULATION AND TEST-BED EXPERIMENTS

As already indicated, the two scenarios discussed in section 2 will be implemented in both a simulation environment and an actual test-bed. Our overall objectives are the following:

- Demonstrate the effects of dynamic price charging and rate adaptation.
- Assess the effectiveness of certain algorithms for rate control and strategies for dynamic price handling, and the dependence thereof on the QoS requirements of the application.

In this section we present and discuss our experimental results related to the rate control and packet marking algorithms described in section 3, which provide initial results towards achieving the aforementioned objectives.

### 4.1 Simulation experiments

Our simulation environment is based on the widely used UCB/LBNL/VINT ns-2 simulator<sup>14</sup>, which supports ECN feedback. On the end-system side, we have modified the congestion avoidance phase of the Tahoe version of the TCP algorithm (this version implements slow start, congestion avoidance, and fast retransmit, but not fast recovery), to implement the TCP/WTP algorithm described in subsection 3.1. Furthermore, because our objective at this stage is to investigate solely the behavior of the congestion avoidance phase, we have disabled the slow start algorithm. At the router side, we have implemented the virtual buffer marking algorithm described in subsection 3.2.

The experiments of this subsection refer to the network topology shown in figure 3. The traffic is produced by ftp sources, which can use as much bandwidth as the TCP algorithm allows them. The start times of the sources are selected randomly, and the aggregate results that we report are averages of 10 runs of the same experiment. Finally, since our objective is to investigate solely the behavior of the congestion avoidance phase, the virtual factor of the router's packet marking algorithm is such that there are no packet losses, namely  $\theta = 0.7$  as shown in figure 3.

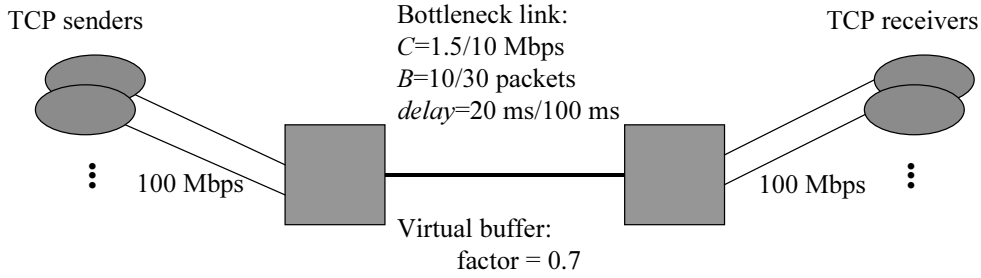


Figure 3. Simulation topology

The experimental results that we present next investigate the following issues:

- Throughput of TCP senders with different willingness-to-pay values, and effect of the packet marking algorithm.
- Effect of the gain parameter  $\bar{k}$  on convergence and stability.

#### 4.1.1 TCP senders with different willingness to pay

Figure 4 shows the congestion window  $cwnd$  as a function of time for two senders with willingness-to-pay  $\bar{w}_1=1$  and  $\bar{w}_2=4$ , respectively. This corresponds to scenario 1 discussed in subsection 2.1, with the charge per unit of time for the “Silver” class being  $w_s = \bar{w}_1 = 1$  and the charge per unit of time for the “Gold” class  $w_g = \bar{w}_2 = 4$ . As expected, the sender of the “Gold” class, which has a higher willingness-to-pay, has larger values for  $cwnd$ . Also, observe that a larger value of  $\bar{w}$  results in larger fluctuations of the congestion window.

Figure 5 shows the ratio of the average throughput as a function of the ratio of willingness-to-pay. Observe that for large values of  $\bar{w}_2/\bar{w}_1$ , the ratio of average throughputs is larger than the ratio of the willingness-to-pay. This behavior can be explained with the help of figure 6, which shows the ratio of marks as a function of the ratio of average throughput. This figure indicates that a smaller percentage of the packets sent by the sender with a larger willingness-to-pay are marked, i.e. the probability of packet marking is smaller for the sender with larger willingness-to-pay. This is due to the combination of the following two facts: first, the sender with a smaller willingness-to-pay sends a smaller number of segments in one round trip time and second, the segments are typically sent back-to-back; the latter being a property of any window-based control mechanism. As a result, the sender with a smaller willingness-to-pay produces a burstier traffic stream compared to the sender with a larger willingness-to-pay. Bursty traffic streams, however, are more difficult for a multiplexer (link) to handle, hence require proportionally more bandwidth than their average rate. The virtual buffer marking algorithm has the property of differentiating streams based on their burstiness. Hence, the algorithm marks a higher percentage of packets belonging to the burstier stream, which is the stream with smaller willingness-to-pay. A similar observation for the case of a slotted model with no buffering was made in reference 7. A marking scheme with the above property has the advantage of giving senders the incentive to shape their traffic. Other packet marking algorithms, such as RED, do not provide such differentiation; related results will be presented in an extended version of this paper.

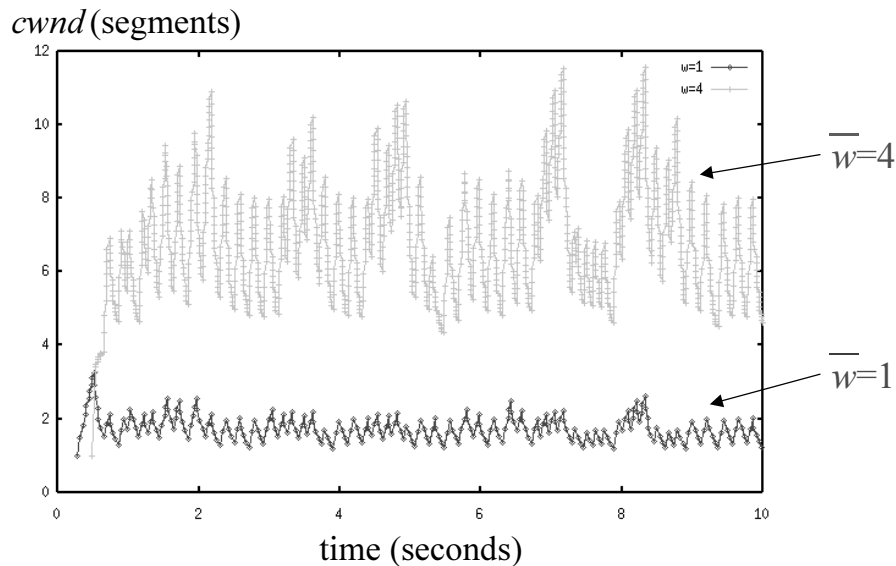


Figure 4. Congestion window for two senders with  $\bar{w}=1$  and 4.  $C=10$  Mbps,  $B=10$  packets,  $RTT=40$  ms, # of sources=10,  $\bar{k}=0.5$

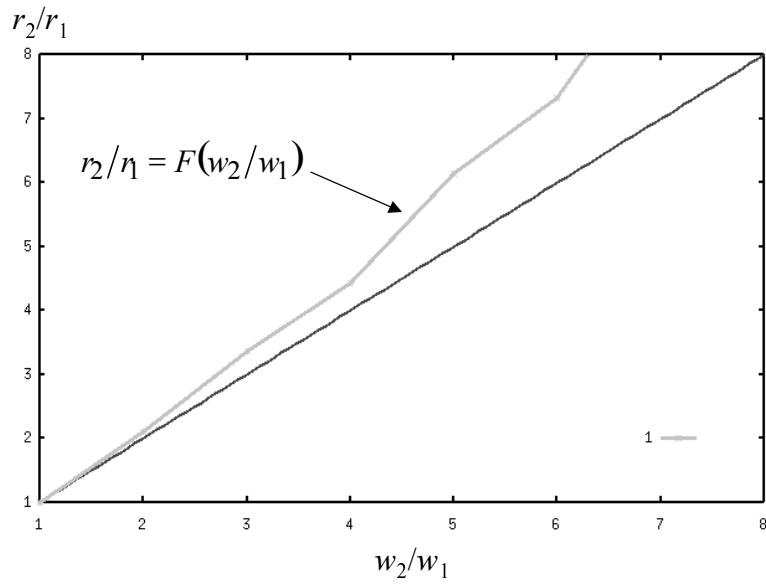


Figure 5. Ratio of average throughput as a function of ratio of willingness-to-pay.  $C=10$  Mbps,  $B=10$  packets,  $RTT=40$  ms, # of sources=2,  $\bar{w}_1=1$ ,  $\bar{k}_1 = \bar{k}_2=1$

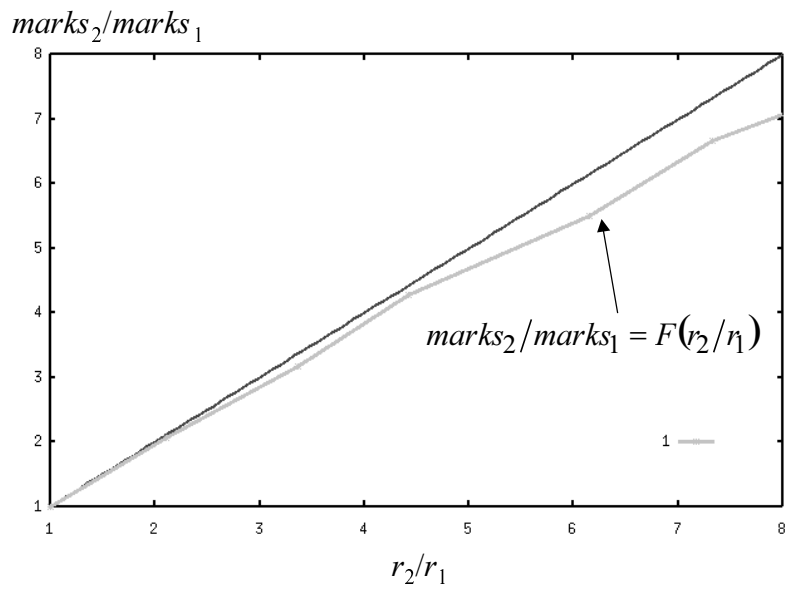


Figure 6. Ratio of marks received as a function of the ratio of average throughput.  $C=10$  Mbps,  $B=10$  packets,  $RTT=40$  ms, # of sources=2,  $\bar{w}_1=1$ ,  $\bar{k}_1 = \bar{k}_2=1$



#### 4.1.2 Effect of the gain parameter on convergence and stability

Next we investigate the effect of the gain parameter  $\bar{k}$  on convergence and stability. Figure 7 shows the convergence of the congestion window to a pattern repeated almost periodically, for different values of the gain parameter (all connections experience the same round trip time, equal to 200 milliseconds). Observe that smaller values of the gain parameter result in smaller fluctuations of the congestion window, but also in longer convergence times. As an example, for  $\bar{k}=0.05$  the convergence time is approximately 40 seconds, a time by which it is likely that most data transfers would be completed; for example, web server traffic has an average size of 8-10 KBytes and average duration 2-9 seconds<sup>5</sup>. This observation stresses the importance of the slow start phase. Moreover, the gain parameter will also determine how fast the congestion windows, hence the rates, adapt to changes of traffic or network conditions.

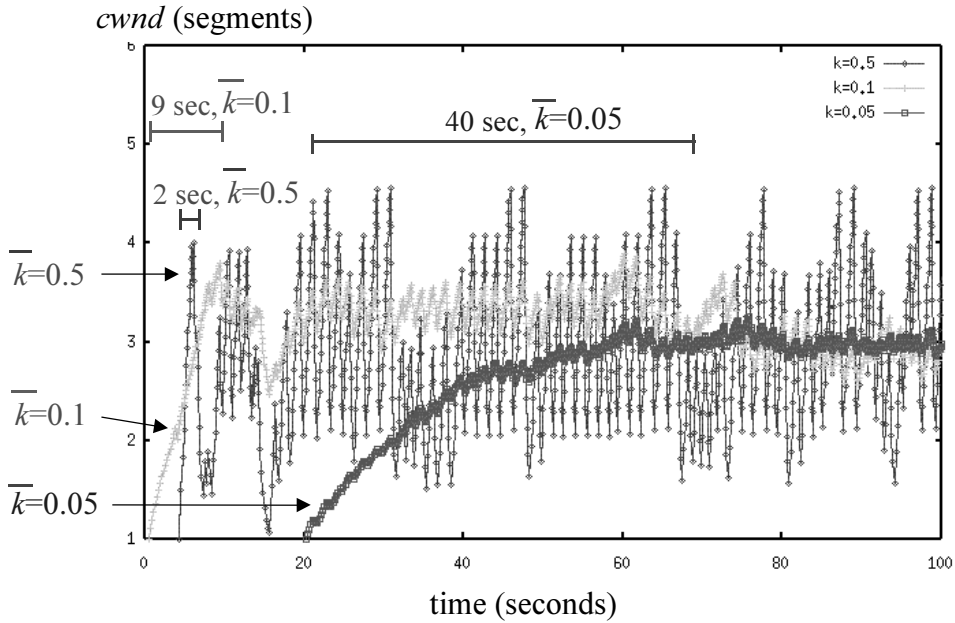


Figure 7. Convergence times for different values of the gain parameter  $\bar{k}$ .  $C=1.5$  Mbps,  $B=10$  packets,  $RTT=200$  ms, # of sources=11,  $\bar{w}=1$

It is interesting to note that, for  $\bar{w}=1$ , the value  $\bar{k}=1$  makes the TCP/WTP algorithm as aggressive as TCP currently is when *every* segment is acknowledged, whereas, the value  $\bar{k}=0.5$  makes the TCP/WTP algorithm as aggressive as TCP when *every other* segment is acknowledged (delayed acknowledgements).

Next we compare the values of the gain parameter shown in figure 7 with those suggested by theory for ensuring stability. Kelly<sup>9</sup>, under the assumption that the traffic produced by sources is Gaussian, obtains the following condition that is sufficient for achieving stability and non-oscillatory behaviour:

$$\bar{k} \left( 1 + \frac{2Q}{\sigma^2} \right) \leq \frac{1}{e}, \quad (2)$$

where  $\sigma$  is the standard deviation of the Gaussian sources and  $Q$  is the queue length at which marking begins. Taking  $Q/\sigma^2=10$  we obtain  $\bar{k} \approx 0.018$ , a value which would result in very long convergence times, as indicated in figure 7.

The above discussion prompts the need for further investigation, since the values of the gain and willingness-to-pay parameters that will be used in practice are likely to be much higher than the values for which theoretical results, under certain assumptions, can ensure stability and convergence.

## 4.2 Test-bed implementation

Our test-bed is comprised of FreeBSD workstations, figure 8. The end-stations include the TCP/ECN patch (distributed with ALTQ 2.1), which was subsequently modified to support the TCP/WTP algorithm described in subsection 3.1. The TCP/ECN patch implements the Reno version of the TCP algorithm. The choice of the TCP algorithm (TCP Reno/ECN or TCP/WTP), as well as the values of the parameters (willingness-to-pay and gain factor) are made via socket options. The middle workstation in figure 8 acts as a router. The kernel of this workstation is compiled with the Alternate Queueing (ALTQ) modules<sup>2</sup>, which support Class-Based Queueing (CBQ), RED, and ECN marking. The implementation of the virtual buffer marking algorithm of subsection 3.2 was added to these modules. Finally, various round trip delays are emulated using DummyNet<sup>12</sup>.

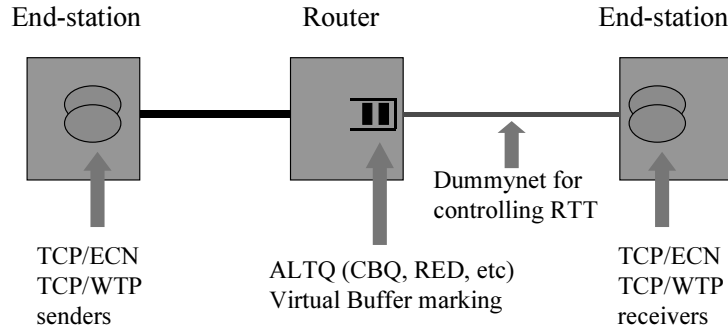


Figure 8. Congestion pricing based on ECN marks test-bed.

One of the reasons for proceeding with the implementation of rate control and packet marking algorithms in an actual test-bed was to investigate the behavior of congestion pricing in an actual environment and verify findings from simulation experiments. Towards this direction, figure 9 shows the ratio of average throughput as a function of the ratio of willingness to pay obtained with simulation and test-bed experimentation. This figure shows that the results obtained from test-bed experiments are generally in agreement with those obtained from simulation, with the most important observation being that the curve obtained with the test-bed is also above the diagonal.

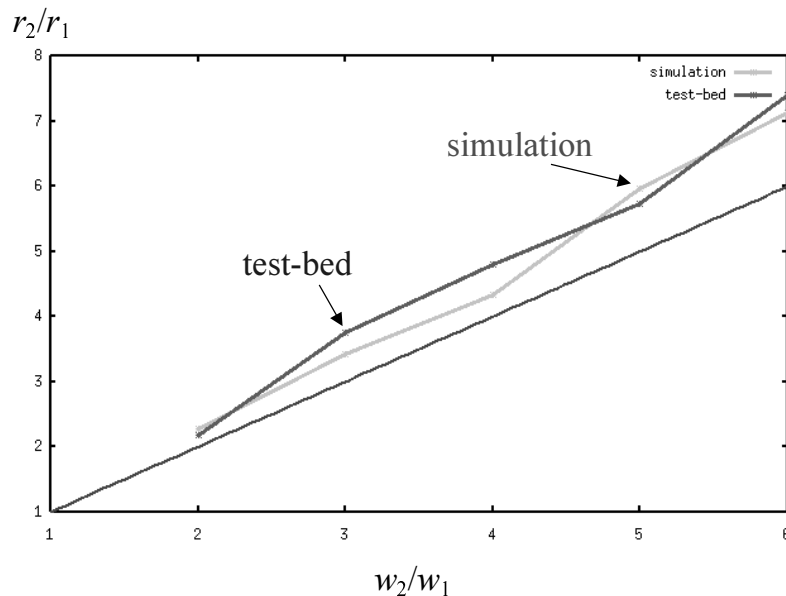


Figure 9. Ratio of throughput as a function of ratio of willingness-to-pay.  $C=1.5$  Mbps,  $B=30$  packets,  $RTT=200$  ms, # of sources=2,  $\bar{w}=1$ ,  $\bar{k}_1 = \bar{k}_1 = 1$

## 5. CONCLUSION AND FURTHER WORK

In this paper, we have described a simulation and experimental environment which aims to demonstrate the feasibility and evaluate the advantages and performance of congestion pricing based on ECN marking for controlling resource usage in a congested network. These objectives are investigated using two service provisioning scenarios. The first scenario involves an application provider that offers discrete Quality of Service (QoS) classes at different prices, whereas the second scenario offers a wider range of QoS, price pairs. We have presented our initial experimental results that involve the implementation of a modified TCP rate control algorithm, the TCP/WTP algorithm, which provides service differentiation based on the senders' willingness-to-pay, and the virtual buffer marking algorithm, which provides early warnings of congestion, where congestion is expressed as packet loss. The experiments presented in this paper are limited to the case where all TCP connections incur the same delay. When connections have different round trip times, the TCP/WTP algorithm, as is the case with legacy TCP, is biased against connections with longer round trip times. This bias can be corrected by adjusting the gain and increase factors of the TCP/WTP algorithm according to the round trip time<sup>7</sup>.

Ongoing work includes the investigation of other rate control, dynamic price handling, and packet marking algorithms. Regarding the former, algorithms we are currently investigating include rate-based algorithms that control the sending rate rather than the number of unacknowledged segments, as is the case with window-based algorithms such as TCP and the TCP/WTP algorithm investigated in this paper. Dynamic price handling algorithms we are currently investigating include algorithms for controlling the sending rate for file and web page transfers, including cases where there is a constraint on the total budget, as well as algorithms for controlling the average sending rate over an interval with duration a few round trip times, rather than the instantaneous sending rate. The latter algorithms are well suited for streaming video/audio applications, which produce bursty traffic. Finally, in the area of packet marking algorithms, we are currently experimenting with algorithms that detect congestion based on measurements of load and we are investigating the inter-working of the TCP/WTP algorithm with RED. An important issue here is the incentives for traffic shaping that the marking algorithms give (recall that the virtual buffer marking algorithm discriminates against bursty flows), and how the sender actions that arise from these incentives affect stability and convergence.

Important features to be added in both environments include the existence of both short-lived and long-lived flows, and a large number of independent and realistic bursty traffic streams. Short-lived flows exhibit the problem that they might be completed in a few round trip times, or even in less than one. One approach for tackling this problem is to use cached information from previous flows that shared the same path, and hence have experienced the same congestion prices.

For generating background traffic, we are currently using application level modeling and techniques for efficiently modeling a large number of independent sources. For example, web server-to-user traffic can be modeled as an on-off source whose "on" periods correspond to the transmission of web pages and "off" periods correspond to the user think time<sup>13</sup>. For the various parameters of such a model, e.g. page sizes and user think time, one can consider empirical distributions derived from actual traffic traces. In addition to modeling the traffic statistics, these traffic generators will include reaction to prices, typically modeled using some form of utility function.

## ACKNOWLEDGMENTS

This work is funded in part by EU's Fifth Framework Project "Market Managed Multiservice Internet" - M3I (RTD No IST-1999-11429).

## REFERENCES

1. M. Allman, V. Paxson, W. Stevens, "TCP congestion control", RFC 2001, April 1999.
2. K. Cho, "A framework for Alternate Queueing: Towards traffic management by PC-UNIX based routers", In *Proceedings of USENIX 1998 Annual Technical Conference*, June 1998. See also ALTQ for FreeBSD: <http://www.csl.sony.co.jp/person/kjc/software.html>
3. M. Christiansen, K. Jeffay, D. Ott, F. D. Smith, "Tuning RED for web traffic", In *Proceedings of SIGCOMM2000*, Stockholm, Sweden, August-September 2000.
4. J. Crowcroft and Ph. Oechslin, "Differentiated end-to-end Internet services using a weighted proportional fair sharing TCP", *ACM Computer Communications Review* 28 (1998), 53-67.
5. A. Feldmann, J. Rexford, and R. Caceres, "Efficient policies for carrying web traffic over flow-switched networks", *IEEE/ACM Transactions on Networking*, 1998, 673-685.
6. S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance", *IEEE/ACM Transactions on Networking*, 1 (1993), 397-413.

7. R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control", *Automatica* 35 (1999), 1969-1985.
8. V. Jacobson, "Congestion avoidance and control", *ACM Computer Communications Review*, 18 (1988), 314-329.
9. F. P. Kelly, "Models for a self-managed Internet", *Philosophical Transactions of the Royal Society*, London, 2000.
10. P. Key and D. McAuley, "Differential QoS and pricing in networks: where flow-control meets game theory", In *IEE Proceedings Software*, 146 (1999).
11. P. Key, D. McAuley, P. Barham, and K. Laevens, "Congestion pricing for congestion avoidance", Microsoft Research report MSR-TR-99-15. <http://research.microsoft.com/pubs/>
12. L. Rizzo, "Dummysnet: a simple approach to the evaluation of network protocols", *ACM Computer Communication Review*, 27 (1997), 31-41. See also: [http://www.iet.unipi.it/~luigi/ip\\_dummysnet/](http://www.iet.unipi.it/~luigi/ip_dummysnet/)
13. B. Mah, "An empirical model of HTTP network traffic", In *Proceedings of INFOCOM97*, Kobe, Japan, April 1997.
14. UCB/LBNL/VINT Network Simulator - ns (version 2). <http://www-mash.cs.berkeley.edu/ns/ns.html>
15. K. K. Ramakrishnan and S. Floyd, "A proposal to add Explicit Congestion Notification (ECN) to IP", RFC 2481, January 1999.