Simple Performance Models of Differentiated Services Schemes for the Internet*

Martin May Jean-Chrysostome Bolot Alain Jean-Marie Christophe Diot

INRIA
2004 route des Lucioles – BP 93
06902 Sophia-Antipolis
France
{mmay,bolot,ajm,cdiot}@sophia.inria.fr

Abstract

Schemes based on the tagging of packets have recently been proposed as a low-cost way to augment the single class best effort service model of the current Internet by including some kind of service discrimination. Such schemes have a number of attractive features, however, it is not clear exactly what kind of service they would provide to applications. Yet quantifying such service is very important to understand the benefits and drawbacks of the different tagging schemes and of the mechanisms in each scheme (for example how much RIO contributes in the Assured scheme), and to tackle key performance and economic issues (e.g. the difference in tariff between different service classes would presumably depend on the difference in performance between the classes). Our goal in this paper is to obtain a quantitative description of the service provided by tagging schemes.

Specifically, we describe and solve simple analytic models of two recently proposed schemes, namely the Assured Service scheme and the Premium Service scheme. We obtain expressions for performance measures that characterize the service provided to tagged packets, the service provided to non-tagged packets, and the fraction of tagged packets that do not get the better service they were supposed to. We use these expressions, as well as simulations and experiments from actual implementations, to illustrate the benefits and shortcomings of the schemes.

1 Introduction

There has been a major effort these past few years aimed at augmenting the single class best-effort service of the current Internet to include services offering a variety of performance guarantees. Providing such services requires first to define the desired services, then to define and evaluate the appropriate admission control, scheduling, and/or signaling mechanisms required to provide these services, finally to implement these mechanisms in the Internet. For example, the IntServ (Integrated Services) IETF working group has identified a number of desirable services in an integrated services Internet [11], the ISSLL (Integrated Services over Specific Lower Layers) working group has specified appropriate admission control and scheduling algorithms for a variety of underlying network technologies, and the RSVP (Resource Reservation Protocol) working group has specified a signaling protocol [2].

An integrated Internet providing a variety of services ranging from the best effort to the deterministic guarantee services is a worthy goal. However, reaching that goal is difficult. For example, many services turn out to require complex associated admission control and scheduling mechanisms, which in turn implies a relatively complex interface between the user/application and the network. Furthermore, there are concerns about the costs associated with the wide deployment of complex scheduling and signaling protocols, and more generally with scalability issues when moving from a stateless to a non-stateless network architecture.

The difficulties above have been tackled in different ways. One way is to develop efficient and scalable algorithms for routers to support the complex policy-based forwarding schemes required in an integrated services Internet [15, 16]. Another way, which is that examined in this paper, is instead to rely on simple schemes and lightweight router support to provide services that extend (even slightly) beyond best effort, where "simple" refers to architectural, user interface, and implementation complexity. Of course, "simple" is still more costly than the current stateless, single class FIFO scheduling. However, the hope is to obtain schemes that provide the basic benefit of an integrated services network, namely some kind of service discrimina-

*To appear in *Proc. IEEE Infocom'99*, New York, NY, March 99.

tion, for a small cost. Examples of such schemes include non-tail-drop schemes such as the Random Early Detection (RED) scheme [9], and a variety of recently proposed "differentiated services" (diff-serv) schemes based on *packet tagging* [3, 5, 21]. The idea of RED is to drop packets with a probability that depends on the mean queue length in routers. This turns out to have a number of fairness advantages. Furthermore, appropriate drop policies can provide different tradeoffs between lower delay and higher loss rate. Extending RED to a per-flow RED can provide protection from malicious connections by punishing (i.e., preferentially dropping packets from) connections that behave in an overly aggressive fashion [9, 7, 17]. We focus instead in this paper on tagging based schemes that explicitly attempt to introduce service discrimination between different connections in the Internet.

The basic idea of diff-serv schemes is to tag some packets (using for example the IP precedence field or the TOS byte) and let tags indicate that the packet should receive preferential treatment. Of course, the idea is not new: packet tagging has been advocated and deployed in Frame Relay (FR) networking using the drop preference bit (DE bit), in ATM networking using the cell loss priority bit (CLP bit) [19, 20, 23], and in a variety of other networks (e.g., [1]). However, tagging more generally can be used to provide different kinds of service discrimination.

Clark [3] argued that a guaranteed, or at least expected, throughput is the one quality of service (QoS) feature of interest to most applications, and that tagging packets should be used to provide such a guarantee. Crowcroft, however, argued [5] that tagging should be used to discriminate between delay sensitive and non-delay sensitive applications and that a guaranteed or expected low delay would benefit applications that most suffer from the current state of the Internet, namely interactive applications such as interactive multimedia or DIS-like applications [22]. Nichols, Jacobson, and Zhang [21] extend Crowcroft's earlier proposal and argue for a service that would combine guaranteed low delay and best effort. Other proposals lean more one way or the other (e.g. [12, 26]).

In any case, diff-serv architectures based on tagging are attractive because they appear to be simple to implement (an indeed DE-bit discrimination has been available for quite some time in commercial Frame Relay products), and they do not rely on a heavyweight state architectures with complex connection admission control, scheduling, and signaling mechanisms. However, current proposals for diff-serv architectures do not *quantify* the service they would provide applications. Quantifying the quality of service offered by differential services is precisely the goal of the paper. This was mentioned in the December 1997 IETF meeting as one of the important "next steps" in the overall diff-serv discussion. And indeed, it is a crucial step to understand the performance benefits and shortcomings of the different schemes and of the different mechanisms in each scheme (for example, how much RIO contributes to the overall performance of the Assured Scheme), and on which to base decisions related to dimensioning (how much to allocate to Premium/Assured data and to best effort data), tariffing (the difference in tariff between different classes would presumably depend on the difference in performance between the classes), etc.

The rest of the paper is organized as follows. In Section 2, we briefly review recently proposed diff-serv architectures. In Section 3 and Section 4, we describe and analyze analytic models for two such proposals, namely the Assured Service (drop priority) scheme [4], and the Premium Service (delay priority) scheme [21]. For both schemes, we obtain analytic expressions for performance measures that characterize the service provided to tagged (i.e. priority) packets, the service provided to non-tagged packets, and the percentage of tagged packets that do not get the better service. We use these expressions, as well as simulations to illustrate the benefits and shortcomings of each scheme. Section 5 concludes the paper.

2 Proposals for a Differentiated Services Internet

In this section, we describe the Assured Service scheme proposed by Clark and Fang [3] and the Premium Service scheme proposed by Nichols and Jacobson [21]. A number of other schemes have been proposed in the DiffServ working group. Most combine in some way or another the principles used in the Assured and Premium schemes; others rely on more sophisticated scheduling mechanisms, for example the User-Share Differentiation (USD) scheme [26] is based on the principle of link sharing. Refer to the diff-serv web page [6] for more complete information.

2.1 The Assured Service Scheme

Clark has argued in [3] that *i)* a guaranteed, or at least, expected throughput is the one quality of service feature of interest to most applications, and thus that, *ii)* there is a need for a mechanism that directly reflects users desires in terms of transfer time. Regarding point *i)*, the idea is that users/applications know the size of data to be transferred and the desired delivery time, which thus can be used to define a minimum transfer rate. This rate then becomes the service objective for the transfer. One way to achieve this objective is to define at the source a *service profile*, which defines how packets should be sent so as to meet the rate objective, and to incorporate a *profile meter*. The profile meter monitors the transfer in progress and tags each packet of the data stream. The tag is set to 1 if the packet is sent according to the profile (i.e., at a rate that conforms to the expected rate); a tagged packet is also referred to as an *In* packet. The tag is set to 0 otherwise; a non-tagged packet is also referred to as an *Out* packet.

Implementing the profile meter depends on how the profile has been defined. For example, a profile that specifies a mean rate and a maximum burst length can be implemented using a leaky bucket filter. Only those packets that conform to a leaky bucket output are tagged, others are not tagged. Tagging schemes using a leaky bucket as profile meter have been studied for ATM networks (see for example [19]). Another type of profile meter for bulk data transfers is the Time Sliding Window (TSW) [4], which provides a smooth estimate of the sending rate of TCP.

The tag information is used by the intermediate routers in case of congestion. *Out* packets are preferentially selected to receive a congestion pushback notification, which typically means that they are dropped. In practice, there are three popular ways to implement such selective drop scheme, using:

- A threshold mechanism: When buffer occupancy reaches a given threshold M, arriving *Out* packets are discarded; *In* packets are still admitted in the queue as long as the buffer is not full.

- A RED with *In* and *Out* packets (RIO) mechanism [4]: RIO extends RED to handle two classes of packets. Specifically, it includes two sets of drop parameters, one for *In* packets and one for the *Out* packets. Service discrimination between the two classes can be achieved in different ways. One way is to use two thresholds to decide when to begin dropping packets, the threshold for *Out* packets being lower than that for *In* packets. Another way is to use the same threshold for both classes but use drop probabilities that increase at different rates as the mean queue length increases.

- A pushout mechanism: An arriving *In* packet may enter a full queue if at least one *Out* packet is already in the queue. Then, one of the *Out* packets is discarded and the *In* packet joins the queue. *Out* packets cannot enter an already full queue and are dropped.

Note that the threshold and the RIO mechanisms differ from the pushout mechanism because they can drop *In* packets while *Out* packets are still in the buffer.

No matter which buffering/scheduling mechanism is implemented, it is intuitively clear that tagged/*In* packets will get a higher throughput than non-tagged packets. In Section 3, we quantify this intuition. Note, that in the absence of admission control, it is impossible to provide deterministic guarantees even to tagged packets. This is precisely the point of the proposal in [3], which argues that the goal is to obtain an expected rather than a guaranteed throughput, and thus that admission control and signaling mechanisms are not required. Instead the network should provide information about the actual usage across the network links to prevent the user from overoptimistic expectations and to help applications adapt to current conditions [8]. Thus, the only building blocks required for implementation are the profile meter (in combination with the mechanisms to determine the traffic profile) and the buffer management schemes (threshold, RIO, pushout) in the routers.

2.2 The Premium Service Scheme

The Premium Service scheme [21] augments the current best effort service with a Premium service that provides applications with a “virtual leased line”. This virtual network is allocated a small share of the bandwidth, and is expected to provide significant delay reduction to the subscribers.

Applications or flows that want to enter the Premium service must specify a desired peak rate and a burst size. The application is not allowed (or supposed to) exceed the peak rate; in return, the network guarantees a (contracted) bandwidth. First-hop routers (or other edge devices) filter the packets entering the network, tag the packets that match a Premium service specification, and perform traffic shaping on the flow thereby smoothing all traffic bursts before they enter the network.

To provide a Premium service, routers implement two levels of priority queueing. Tagged (Premium) packets are sent first; non-tagged (best effort) packets are queued and sent only when all tagged packets have been sent. Thus, in practice, the Premium Service is visible by applications as two "virtual networks": one which is identical to today’s Internet with buffers designed to absorb traffic bursts; and one where traffic is limited and shaped to a contracted peak-rate, but packets move through a network of queues where they experience almost no queueing delay.

There is consequently a very limited queue management at the network node levels. Admission control (if used) and policing are only done at the edge-routers. Then, the Premium class behaves as a flow aggregation, and does not require the control of each flow within the network.

3 Modeling and Evaluation of the Assured Service Scheme

In this section, we describe and solve a model of the Assured Service scheme, and we use the results to evaluate the performance of the scheme.

Modeling the Assured Service Scheme

We saw earlier that the Assured Service (AS) scheme uses in the Internet environment concepts widely used (or at least widely studied, if still little used) in the ATM environment; in particular concepts such as tagging (*In/Out* packets in the the AS scheme, CLP bit in ATM) and policing (profile meter in the AS scheme, leaky bucket policing in ATM). Thus, it is natural to refer to the vast literature on CLP-based tagging schemes and leaky bucket profiling; see for example Elwalid and Mitra [19], Kröner *et al.* [14], and Roberts *et al.* [23]. We do not use the complex (but accurate) model of [19] which is only amenable to numerical solutions. Instead, we use an approach and models similar to those in [14, 23], however we add to those a model for RIO. The resulting model is simple and it yields closed form expressions for performance measures of interest.

Recall that RIO drops incoming packets with a probability that depends on buffer occupancy and on whether packets are *In* or *Out* packets. Clearly, the router has to keep track of the average number of *In* packets (avg_{in}) and the average of the total number of packets in the buffer (avg_{total}). The drop probability of *In* packets then depends on avg_{in}, the drop probability of *Out* packets depends on avg_{total}. We now describe our model of a queue in the Assured Service scheme. Refer to Figure 1.

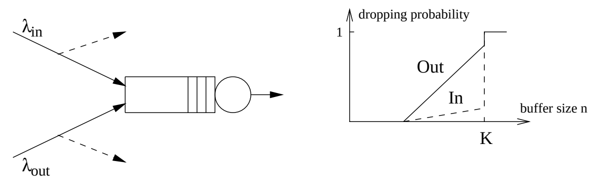

Figure 1: A queue with *In* and *Out* packets and a RIO buffer management scheme

Assume packets arrive in the queue according to a Poisson process with rate λ. Packets are *In* packets with probability p, and *Out* packets with probability $\overline{p} = 1 - p$. We assume that both types of packets require a service exponentially distributed with parameter μ. We denote the total offered load of the system by $\rho = \lambda/\mu$, and the buffer size in packets as K.

Given the recent results showing long range dependence being a salient feature of Internet traffic [27], the Poisson assumption has to be handled with care. The main reason we use it in our analysis is of course mathematical tractability. Indeed, the mathematical analysis of queues with long range dependent (LRD) input traffic is a thorny problem [28]; explicit results for queues with finite buffers are few, and we are not aware of any result with finite buffer queues and priority buffer management. Furthermore, simulation results (shown below) with long range dependent input traffic show good correlation with analytic results obtained with the Poisson hypothesis. Note that this is in agreement with recent results on the relevance of LRD vs. Markovian traffic models with finite buffer queues [10].

We model buffer management algorithms such as RIO using a function $\alpha(n)$. Specifically, let $\alpha^T(n)$ be the probability that an arriving tagged (*In*) packet is accepted given that n packets are already present in the queue. Likewise, let $\alpha^{NT}(n)$ be the acceptance probability for non-tagged (*Out*) packets when n packets are found in the queue. The probability that an arriving packets is accepted is $\alpha(n) = p\alpha^T(n) + \overline{p}\alpha^{NT}(n)$. If the total buffer capacity is K, then $\alpha^T(K) = \alpha^{NT}(K) = 0$.

Modeling different buffer management schemes can be done simply by choosing appropriate values for α^T and α^{NT}. We have examined three schemes, which we refer to as TAIL (tail drop, no specific buffer management), RIO (RED with tagged and non-tagged packets), and THRESH (threshold drop). There are defined as follows:

TAIL No specific buffer management scheme: $\alpha^T(n) = \alpha^{NT}(n) = 1,\ 0 \le n \le K-1$

RIO Accept all packets until the queue size is equal to $K/2$. Then drop with a probability that increases linearly to 10% for tagged packets, and to 90% for non-tagged packets. Specifically,

$$\left\{\begin{array}{l} \alpha^T(n) = 1 \\ \alpha^{NT}(n) = 1 \end{array}\right. \qquad n \le K/2$$

$$\left\{\begin{array}{l} \alpha^T(n) = 1 - 0.1(2n-K)/K \\ \alpha^{NT}(n) = 1 - 0.9(2n-K)/K \end{array}\right. \qquad K/2 \le n < K-1 \tag{1}$$

THRESH Accept all packets until the queue size reaches $K/2$, then drop all non-tagged packets but accept all tagged packets. Formally,

$$\alpha^T(n) = 1, \quad 0 \le n \le K-1$$

$$\left\{\begin{array}{ll} \alpha^{NT}(n) = 1 & n \le K/2 \\ \alpha^{NT}(n) = 0 & n > K/2 \end{array}\right. \tag{2}$$

Thus, THRESH is similar RIO except that low priority packets are dropped with probability 1. Refer to Figure 2.

We now get back to solving the model. Given the assumptions made, the number of packets in the buffer is a Markov chain. It is actually a birth and death process with birth rate in state n equal to $\lambda\alpha(n)$ and death rate equal to μ (if $n \neq 0$). Accordingly, the stationary distribution of the buffer contents is easily computed [13]:

$$\pi(n) \;=\; \pi(0)\ \rho^n \prod_{i=0}^{n-1} \alpha(i) \tag{3}$$

where

$$\pi(0) \;=\; \left[\sum_{n=0}^{K} \rho^n \prod_{i=0}^{n-1} \alpha(i)\right]^{-1}$$

Let π^T_{rej}, π^{NT}_{rej} and π_{rej} be the drop probabilities for tagged packets, non-tagged packets, and all packets, respectively. Using the PASTA (Poisson Arrivals See Time

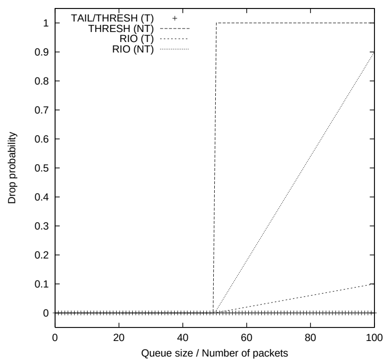

Figure 2: Drop probabilities for TAIL (tail drop), RIO (same thresholds for both classes, but different drop probabilities), and THRESH (simple threshold $th = 50$), buffer size $K = 100$. T stands for tagged packets, NT for non-tagged packets.

Averages) property, we have:

$$\pi^T_{\text{rej}} \;=\; 1 - \sum_{n=0}^{K} \alpha^T(n)\pi(n), \quad \pi^{NT}_{\text{rej}} \;=\; 1 - \sum_{n=0}^{K} \alpha^{NT}(n)\pi(n)$$

and

$$\pi_{\text{rej}} \;=\; 1 - \sum_{n=0}^{K} \alpha(n)\pi(n) \qquad (4)$$

Then, the effective throughput of packets, by class and globally, is given by:

$$\lambda^T_{\text{eff}} \;=\; \lambda p \sum_{n=0}^{K-1} \alpha^T(n)\pi(n), \quad \lambda^{NT}_{\text{eff}} \;=\; \lambda \overline{p} \sum_{n=0}^{K-1} \alpha^{NT}(n)\pi(n)$$

$$\lambda_{\text{eff}} \;=\; \lambda \sum_{n=0}^{K-1} \alpha(n)\pi(n) \qquad (5)$$

We can also derive the distribution of the response time for packets accepted in the queue when the scheduling discipline is FIFO. In that case, if n packets are found in the queue, the response time is the sum of $n+1$ exponentials with parameter μ. In particular, the expected delay in the queue is, for each class:

$$D^T \;=\; \frac{1}{\mu} \sum_{n=0}^{K-1} (1+n)\pi(n)\alpha^T(n)$$

$$D^{NT} \;=\; \frac{1}{\mu} \sum_{n=0}^{K-1} (1+n)\pi(n)\alpha^{NT}(n)$$

We next present numerical results obtained from the analysis above. We first examine how well or how badly the Poisson hypothesis fares in practice when compared with simulation results obtained with bursty input traffic. Then, recalling that the Assured Service scheme was designed to provide applications with an expected throughput, we examine λ_{eff} – and the related measure π_{eff} – for the different types of packets and the different buffer management schemes.

Validating the Poisson hypothesis

Figure 3 shows the loss probability for the tagged and non-tagged packets computed from the model (and thus with the Poisson hypothesis), and obtained using simulation. The simulation results shown in the figure correspond to three different traffic assumptions: 1) Poisson arrivals, 2) a superposition of 32 independent on/off sources with constant bit rate during On periods and exponentially distributed On and Off periods, 3) a superposition of 32 independent on/off sources with constant bit rate during On periods distributed according to a Pareto distribution with the parameter $\alpha = 1.4$ or Hurst parameter $H = 0.8$. In all cases, the value of p is 0.95 and $K = 100$ packets.

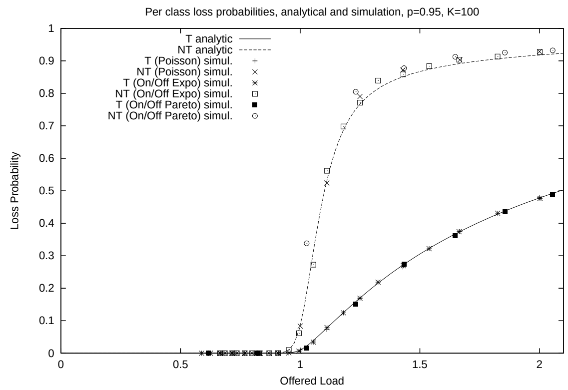

Figure 3: Comparing analysis and simulation: loss probability vs. load ρ, buffer size $K = 100$, fraction of tagged packets $p = 0.95$

Not surprisingly, we find that the simulation results with the Poisson traffic perfectly match the analytic results. However, we also observe very good correlation between the analysis and the simulation with long range dependent traffic. As expected, we also observe that the loss rate for non-tagged packets rapidly increases toward 1 when load gets close to 1, whereas the loss rate for tagged packets increases significantly only under extreme load ($\rho \gg 1$).

Figure 4 extends Figure 3 to consider a range of values for p. The graph on the left hand side shows the loss probability for tagged packets as a function of load and p; the graph on the right hand side shows the loss probability for non-tagged packets. Upon close look, it appears that each graph includes three surfaces: the top surface is that obtained when input traffic is a superposition of 32 Pareto On/Off sources, the medium surface is obtained when input traffic is a superposition of 32 exponential On/Off sources. The lower surface is obtained when using Poisson Arrivals. The three surfaces are very close to each other, which is confirmation of what we observed in Figure 3.

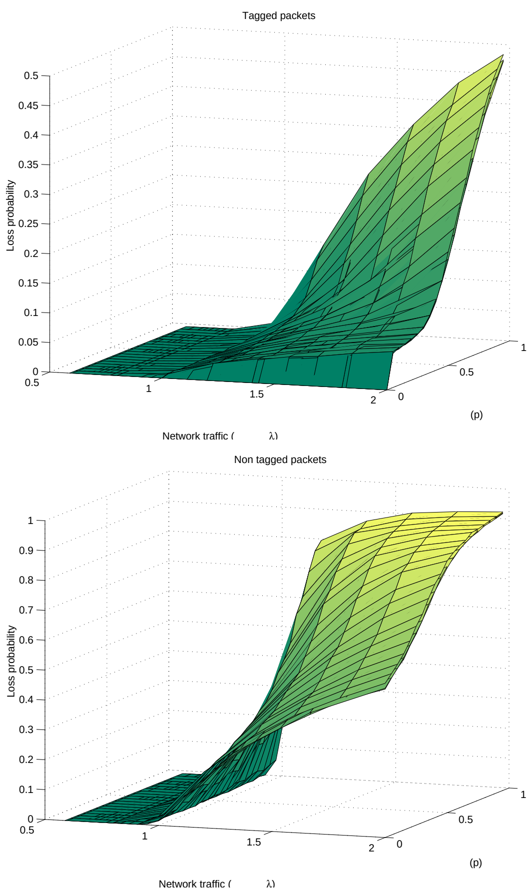

Figure 4: Loss probability for tagged packets (top) and non-tagged packets (bottom), vs load λ and fraction of tagged packets p. The three surfaces in each graph correspond to Pareto (top surface), exponential (medium surface) On periods, and Poisson Arrivals (bottom surface)

Throughput analysis and the impact of RIO

We now examine the effective throughput as a function of the load ρ and of the buffer management scheme. Figure 5 shows the effective throughput λ_{eff}, λ_{eff}^{T} and $\lambda_{\text{eff}}^{NT}$ for the different types of packets, as a function of the total offered load. In all cases, we have $p = 0.5$ and $K = 100$.

Note that the effective throughput for both tagged and non-tagged packets is the same with the TAIL scheme. More generally, we find that the effective throughput for both tagged and non-tagged packets is little dependent on the buffer scheme when $\rho < 1$. However, the impact of each scheme is very clear when $\rho > 1$, i.e. at high load. With TAIL, tagged and non-tagged packets are not treated differently, and they share the bandwidth equally. With RIO, however, and even more so with THRESH, tagged

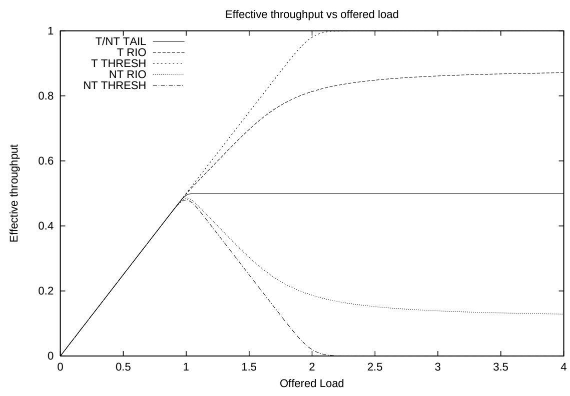

Figure 5: Effective throughput vs offered load, $K = 100$, $p = 0.5$. T stands for tagged packets, NT for non-tagged packets.

packets get a larger fraction of the total bandwidth. In fact, it appears that i) the share of the bandwidth grabbed by each type of packets converges to some value as the load increases, and ii) the share (NT, T) is (0% , 100%) with THRESH. To understand this better, let us go back to Equation (5). When $\rho > 1$, we find

$$\begin{aligned}
\lambda_{\text{eff}}^{T} &= \phi\,\frac{1}{\mu} + \frac{1}{\rho}\frac{p}{\mu\alpha(K-1)} \times \\
&\quad \left(\frac{\alpha^{T}(K-2)}{\alpha(K-2)} - \frac{\alpha^{T}(K-1)}{\alpha(K-1)}\right) + O(\frac{1}{\rho^2}) \qquad (6) \\
\lambda_{\text{eff}}^{NT} &= (1-\phi)\,\frac{1}{\mu} + \frac{1}{\rho}\frac{1-p}{\mu\alpha(K-1)} \times \\
&\quad \left(\frac{\alpha^{NT}(K-2)}{\alpha(K-2)} - \frac{\alpha^{NT}(K-1)}{\alpha(K-1)}\right) + O(\frac{1}{\rho^2}) \qquad (7) \\
\lambda_{\text{eff}} &= \frac{1}{\mu} + O(\frac{1}{\rho^K}) \qquad (8)
\end{aligned}$$

with

$$\phi = \frac{p\alpha^{T}(K-1)}{p\alpha^{T}(K-1) + \overline{p}\alpha^{NT}(K-1)}. \qquad (9)$$

Thus we see that at high load, both types of packets share the bandwidth μ, the tagged packets getting a fraction ϕ of it with ϕ given in Eq. (9) above. For TAIL, $\phi = 1/2$ and the bandwidth is shared equally; thus, there is no service discrimination. For THRESH, $\phi = 1$ and the tagged packets get all the bandwidth. For RIO, $0 \leq \phi \leq 1$; for the values used in the figure, we have $\phi = 451/510 \simeq 88.4\%$ which matches what we observe above.

We make two important observations about ϕ. First, Equation (9) shows that ϕ depends only on the probability of being accepted in the *last* buffer position, but *not* on the general shape of the drop function α. Second, Figure 5 shows that the effective throughput for tagged and non-tagged packets converges slowly as ρ increases to the steady state values ϕ/μ and $(1-\phi)/\mu$. One way then to have a *load-independent* sharing of the bandwidth between the two types of packets is, given the results in Equations (6) and (7), to choose the acceptance rates such that the ratios $\alpha^H(K-1)/\alpha^H(K-1)$ and $\alpha^H(K-2)/\alpha^H(K-2)$ are equal, i.e.

$$\frac{p\alpha^T(n)}{p\alpha^T(n) + \overline{p}\alpha^{NT}(n)} = \phi$$

An important design rule is thus:

To achieve a $(\phi, 1-\phi)$ load-independent sharing of the bandwidth between tagged and non-tagged packets, choose the RIO drop probability such that

$$\alpha^{NT}(n) = \frac{1-\phi}{\phi} \frac{p}{1-p} \alpha^T(n) \tag{10}$$

Note that the function $\alpha^{NT}(n)$ used so far for deciding whether to accept or drop an incoming packet depends on the the *total* number n of packets in the queue. The RIO scheme in [4] uses instead the number of tagged packets in the queue as a basis for dropping tagged/*In* packets. A little thought shows having α depend on the total number of packets or the number of tagged packets only does not matter with the TAIL and THRESH schemes since tagged packets are always accepted then. It turns out that the impact in the case of the RIO scheme is small as well. This can be shown using analysis or simulations. An analytic approach requires that we extend the basic model above, since we now have to track the type (tagged or non-tagged) of each packet in the queue. Thus, the state representation of the queue becomes $(c_0(t), c_1(t), c_2(t), \ldots, c_K(t))$, where $c_k(t)$ is the type of the packet in position i in the buffer (the packet currently served having index $i = 0$). Unfortunately, the state space now has size of order $\sim 2^{K+1}$. We have solved the extended model for up to $K = 16$ and compared the results with that of the previous model. Figure 6 shows the effective throughput of tagged packets for the two models: the continuous curves show the effective throughput computed with equations (5), the discrete points show the effective throughput computed with the extended model. Note that in both models non-tagged packets are accepted based on the total number of packets in the queue.

The graphs clearly show that our model is a good approximation to the scheme described by Clark, and that it does not matter whether the acceptance probability of tagged packets depends on the total number of packets or on the number of tagged packets in the queue, as suggested in [4].

We also note that in Clark's original paper, the author points out that the RIO parameters have to be carefully chosen. However, in their latest publication [4], the authors use a different set of parameters. An important question then is how sensitive are the results to the choice of specific parameter values. This is another case where our model can be useful. Figure 7 compares the difference in loss probability for tagged packets for two parameter sets, namely the parameter set proposed by Clark in [4] on one hand, with parameter values $min\_threshold = 10$, $max\_threshold = 30$, $max\_p = 0.2$ for non-tagged packets and $min\_threshold = 40$, $max\_threshold = 70$, $max\_p = 0.2$ for tagged packets; and another parameter set with the same thresholds for both classes ($min\_threshold = 50$ packets, $max\_threshold = 100$ packets), but $max\_p = 0.9$ for the non-tagged packets and 0.1 for the tagged packets. In both cases the buffer size is $K = 100$. The two surfaces in the plot are not close to each

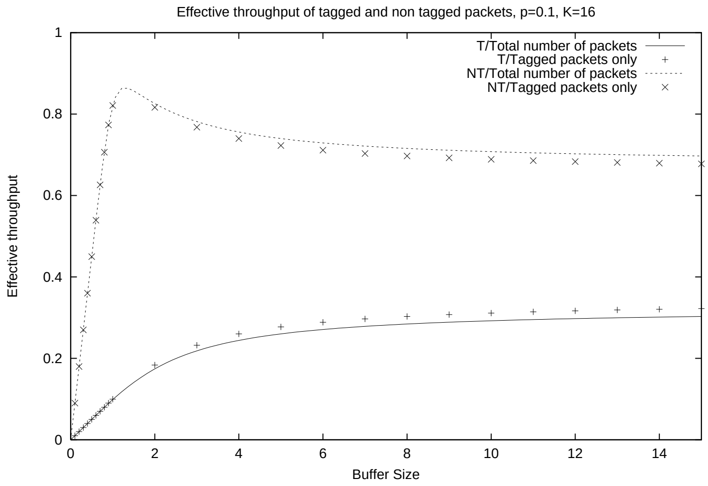

Figure 6: Effective throughput for tagged packets with RIO and drop probability depending on total number in queue (lines) and number of tagged packets in queue (points)(T stands for tagged packets, NT for non-tagged packets)

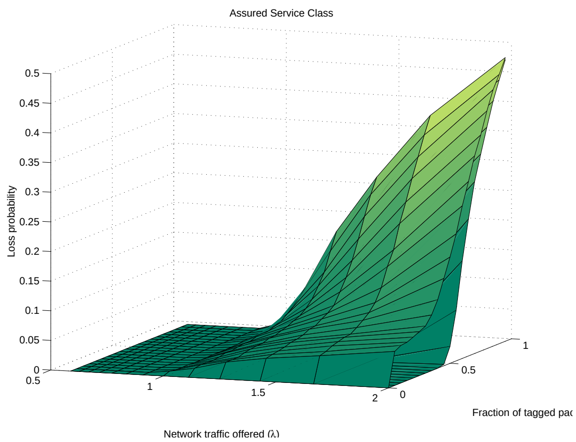

Figure 7: Loss probability of tagged packet with different parameter sets. The lower surface corresponds to the parameter set proposed in [4], the upper surface corresponds to the other parameter set we used in Figure 2

other over a wide range of parameter values. Thus we see that the choice of parameter values can have a clear impact on performance, and that our model can be used to find the parameter set needed to achieve a given behavior.

4 Modeling and Evaluation of the Premium Service Scheme

Modeling the Premium Service Scheme

We model a router in a network with Premium Service as follows: the router includes two separate queues, one with finite (and small) size K accessible only to tagged packets, the other infinite and accessible by non-tagged packets. The finite queue models the finite buffer of a shaping device (e.g. a leaky bucket shaper); thus the delay in the queue is the delay caused by the shaping of the tagged packets. Tagged packets are then served until the finite queue is empty. When that queue is empty, non-tagged packets get service. If the tagged queue is full, arriving tagged packets are discarded. Refer to Figure 8.

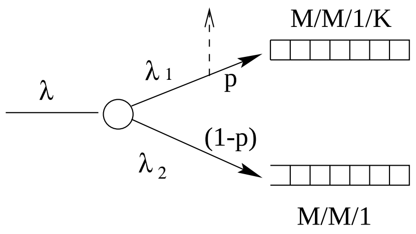

Figure 8: A model for the Premium Service scheme

We assume that the input stream is Poisson with rate λ, that the arriving packets are tagged (class 1) with probability p and non-tagged (class 2) with probability $1-p$. Alternately, we may consider that the input streams of the high and low buffers are independent Poisson processes with rates $\lambda_1 = p\lambda$ and $\lambda_2 = (1-p)\lambda$ respectively. Let $\rho = \lambda/\mu$ and let $\rho_1 = \lambda_1/\mu$ be the load factor of the tagged queue.

Our objective is to determine R_i, the response time of a class-i packet accepted in the system. We assume for simplicity that the service discipline is *preemptive*, and that the service times are exponential with parameter μ (however this can be easily extended to the case of a general distribution).

For the high priority customers we find that R_1 is the response time of customers in a $M/M/1/K$ queue with arrival rate λ_1. Let N_1 be the stationary number of customers in this queue. By Little's law and well known results [13], we have

$$ER_1 = \frac{EN_1}{\lambda_1} = \frac{1}{\mu - \lambda_1}\ \frac{1-(K+1)\rho_1^K + K\rho_1^{K+1}}{1-\rho_1^{K+1}} \ . \quad (11)$$

We also obtain the loss probability of the high priority class:

$$\pi_1 \ = \ \rho_1^K \ \frac{(1-\rho_1)}{1-\rho_1^{K+1}} \ .$$

Then we find the *stability* condition of the system: $\mu > \lambda_2 + \lambda_1\pi_1 = \lambda(1-p(1-\pi_1))$. From this condition, we find the maximum admissible offered load (with p fixed), or, for a fixed offered load, the minimal fraction p of packets that must have high priority.

The response time of the low priority customers is more difficult to obtain that R_1. We develop below bounds based on the analysis of [25] and [18]. Note that in any case, the response time of each customer of class 2 is less than what it would be if K were replaced by $K' > K$. This is clear because in the latter case, there would be more high priority customers accepted and the preemption periods would be longer. In particular, if $K = \infty$, our model reduces to a two-class $M/M/1$. The Laplace transform of the waiting time in this queue (actually, in the $M/GI/1$ queue) is known from [25], for both preemptive and non-preemptive priorities. In our case, we obtain

$$ER_2 \ \le \ ER_2^\infty \ := \ \frac{1}{\mu}\ \frac{1}{1-\rho_1}\ \frac{1}{1-\rho} \quad (12)$$

A tighter bound is possible. Following [25], we argue that the response time R_2 is the sum of χ_2, the workload encountered by this customer upon arrival (including its own service time, because the discipline is preemptive), plus as many preemption periods as there were arrivals of high priority customers during the execution of this workload. We denote the number of such arrivals by $A(\chi_2)$. Thus,

$$R_2 \ = \ \chi_2 \ + \ \sum_{j=1}^{A(\chi_2)} B_j \quad (13)$$

Because of the finite buffer capacity, the distributions of χ_2, $A(\chi_2)$ and of the durations B_j are not easy to obtain. However, we note that χ_2 is less than R_2^∞, the stationary response time in the $M/M/1$ queue, which is known to be exponential with parameter $\mu - \lambda_1$. Thus, $A(\chi_2)$ is less than the number of arrivals of the Poisson process of rate λ_1 in the interval $[0, R_2^\infty]$. Finally, the duration of the preemption periods is less than the length B_K of the busy period in the $M/M/1/K$ queue. The expected values EB_K, which we denote by b_K, are given by [18]:

$$b_n \ = \ (b_{n-1} - \sum_{j=1}^{n-1} b_j p_{n-j}) \ / \ p_0, \quad (14)$$

$$p_k \ = \ \int_0^\infty (\lambda_1 t)^k / k! e^{-\lambda_1 t} d\sigma(t) \ = \ \frac{\rho_1}{(1+\rho_1)^{k+1}} \quad (15)$$

starting with $b_0 = 1/\mu$. Combining all the above, we obtain from (13) the bound:

$$ER_2 \ \le \ E\chi_2 \ (1 \ + \ \lambda b_K)$$

$$\leq \quad ER_2^{\infty}\ (1\ +\ \lambda b_K)\ , \qquad (16)$$

where, ER_2^{∞} is given in Eq. (12). The rigorous justification of Eq. (16) relies on methods of stochastic ordering [24].

It turns out that the bound in Equation (16) above is fairly tight. Since it is also very fast to compute, it is possible to envision the online control of the system (e.g. best choice of K or p) based on some criterion involving λ, ER_1, ER_2 and/or π_1.

Finally, note that the same approach as that used in this section may be applied to the *non-preemptive* case, and that it would be possible to extend it to generally distributed service times, and to analyze queue length and jitter *distributions*. using approaches such as in [23]. The details are omitted here for lack of space.

Validating the Poisson hypothesis

We carry out for the Premium Service the same type of validation we carried out for the Assured Service scheme. Specifically, we simulated the system with two different traffic sources which are a superposition of exponential On/Off sources and Pareto On/Off sources.

Figure 9 shows the mean delay of tagged packets with exponentially distributed interarrival times, with exponential On/Off sources, and with Pareto On/Off sources.

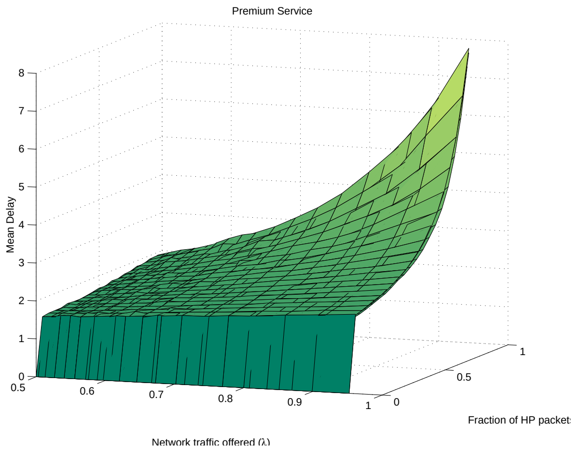

Figure 9: Mean delay of tagged packets vs. load with Poisson arrivals, exponential distributed, and Pareto distributed On/Off periods. Buffer Size $K = 100$.

The three surfaces are so close together that they are almost indistinguishable. This confirms what we observed in Figure 3, namely that the results of the model hold for exponential and bursty input traffic.

Delay analysis and experiments

Figure 10 shows the mean delay of tagged and non tagged packets for a buffer size $K = 100$ and service time μ =1. The total offered load λ varies from 0.5 to 0.95 on the y-axis, the fraction of traffic which is tagged p varies from 0 to 1 on the x-axis.

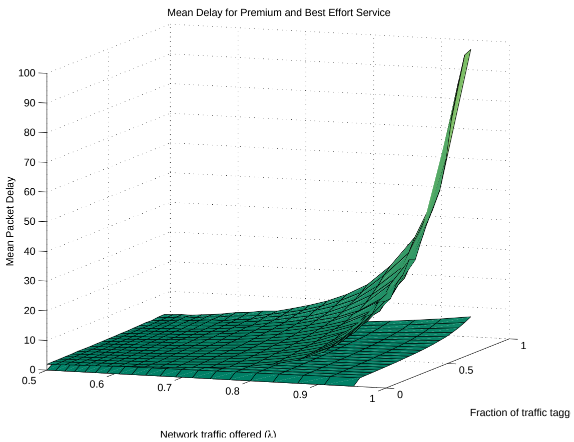

Figure 10: Mean delay of tagged (bottom surface) and non tagged (top) packets for $\lambda = 0.5..0.95/2.0$ and $p = 0..1$, with $K = 100$

Not surprisingly, the maximum delay experienced with Premium service is always smaller than the delay experienced with best-effort service. In fact, the delay for tagged packets remains very close to the minimum delay. On the other hand, the non-tagged packets suffer a very large delay because they have to wait for the busy periods of the tagged queue to get service. As the fraction of tagged packets in the total traffic increases, so does the difference in delay between the two service classes. Thus, the network provider has to decide for up to which fraction of tagged packets the increase in delay for the non-tagged packets is acceptable.

The buffer size of the tagged queue is another important performance parameter, which controls the trade-off between the loss probability of tagged packets and the delay of non-tagged packets. This is shown in Figure 11 for $\rho = 0.5$.

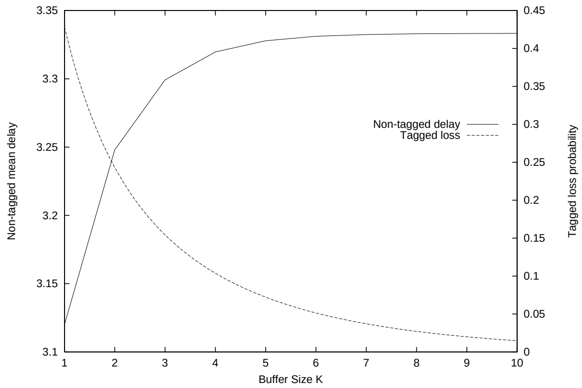

Figure 11: Tradeoff between delay for non-tagged packets and loss probability of tagged packets as a function of the Premium Service buffer size.

Clearly, varying the buffer size from 1 to 10 packets controls the trade-off by decreasing the tagged packets' loss rate and increasing the mean delay of the non tagged packets. The buffer size we use in this plots can also be thought of as the fraction of buffer size reserved by an ISP for the Premium Service.

We have implemented the Premium service on a testbed at INRIA to gain experience with implementation complexity, and to obtain experimental results to complement and validate the analytic and simulation results above. PCs running Linux 2.0.25 in which we have modified the buffer management and scheduling discipline act as routers. Measurements tie in well with the analytic results in this paper.

5 Conclusion

We have analyzed analytic models for the Assured and Premium differentiated service schemes, and we have obtained expressions for performance measures that characterize the service provided to tagged and non-tagged packets and the fraction of tagged packets that do not get the desired service. The models are very simple, yet they tie in well with simulation results. They also bring out interesting features, such as the impact of RIO on the performance of the Assured Service scheme, and they can be used to derive dimensionning and configuration rules, such as the load insensitive sharing rule, also for the Assured Service scheme.

We are pursuing this work in several directions. One is to develop a model in which we could quantify the impact of service discrimination on a *per connection* basis, as opposed to a *per aggregate-of-connections* basis as is done in this paper, the goal being to derive a more accurate characterization of the service provided to an individual user. Another one is to extend the models for a multi-class Premium and Assured Service, and to develop and examine the impact of pricing schemes suitable for diff-serv networks.

References

[1] Bala, K., et al., "Congestion Control for High-speed Packet Switched Networks", *Proceedings of Infocom'90*, pp. 520-526, April 1990.

[2] Braden, R., et al., "RSVP: A New Resource ReSerVation Protocol", *IEEE Network*, September 1993.

[3] Clark, D., "Adding Service Discrimination to the Internet", LCS MIT Technical report, Sept. 1995. http://ana-www.lcs.mit.edu/anaweb/ps-papers/TPRC2-0.ps

[4] Clark, D., Fang, W.,"Explicit Allocation of Best Effort Packet Delivery Service ", Internet draft, Sept. 1997.

[5] Crowcroft, J., "All you need is just 1 bit", keynote presentation, *IFIP Conf. on Protocols for High Speed Networks*, Sophia Antipolis, Oct. 1996. http://www.cs.ucl.ac.uk/staff/jon/hipparch/dollarbit/

[6] DiffServ Working Group sites http://diffserv.lcs.mit.edu/ http://www.ietf.org/html.charters/diffserv-charter.html

[7] Fall, K., Floyd, S., "Promoting the use of end-to-end Congestion control cn the Internet", LBL technical report, Feb. 1998.

[8] W. Feng, D. Kandlur, D. Saha, K. Shin, "Adaptive packet marking for providing differentiated services in the Internet", *Proc. ICNP'98*, Austin, TX, Oct. 1998.

[9] Floyd, S. and Jacobson, V., "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397-413, August 1993.

[10] M. Grossglauser, J. Bolot, "On the relevance of long range dependence in network traffic", *Proc. ACM Sigcomm'96*, Stanfrod, CA, Sept. 1996.

[11] Integrated Services working group http://www.ietf.org/html.charters/intserv-charter.html

[12] Kilkki, K., "Simple Integrated Media Access", draft-kalevi-simple-media-access-01.txt, June 1997.

[13] Kleinrock, L., *Queueing Systems*, vol. 1, J. Wiley & sons, 1975.

[14] Kröner, H., Hébuterne, G., Boyer, P. and Gravey, A., "Priority Management in ATM Switching Nodes", *IEEE Journal on Selected Areas in Networking*, vol. 9, no. 3, pp. 418–427, April 1991.

[15] V. J. Kumar, T. V. Lakshman, D. Stiliadis, "Beyond best effort: Gigabit routers for tomorrow's Internet", *IEEE Communications Magazine*, May 1998.

[16] T.V. Lakshman, D. Stiliadis, " High speed policy-based packet forwarding using efficient multi-dimensional range matching", *Proc. ACM Sigcomm '98*, Vancouver, Canada, Sept. 1998.

[17] D. Lin, R. Morris, "Dynamics of random early detection", *Proc. ACM Sigcomm'97*, Cannes, France, Sept. 1997.

[18] Miller, L. W., "A Note on the Busy Period of an M/G/1 Finite Queue", *Op. Res. 23* (1975), pp. 1179–1182.

[19] Elwalid, A., Mitra, D., "Fluid Models for the Analysis and Design of Statistical Multiplexing with Loss Priorities on Multiple Classes of Bursty Traffic", *Proceedings of Infocom 92*, pp 415-425, 1992.

[20] Elwalid, A., Mitra, D., "Analysis and Design of rate-based congestion control of high speed networks, I: stochastic fluid models, access regulation ", *Queueing Systems*, Vol.9, 29-64, 1991.

[21] Nichols, K., Jacobson, V., Zhang, L.,"A Two-Bit Differentiated Services Architecture for the Internet", Internet draft draft-nichols-diff-svc-arch-00.txt, November 1997.

[22] Pullen, J., Myjak, M., Bouwens, C., "Limitations of Internet protocol suite for distributed simulation in the large multicast environment", Internet draft draft-ietf-lsma-limitations-04.txt, November 1998.

[23] Roberts, J., Mocci, U., Virtamo, O. (eds), *Broadband Network Teletraffic*, Lecture Notes in Computer Science, vol. 1155, Springer, 1996.

[24] Stoyan, D., *Comparison Methods for Queues and Other Stochastic Models*, J. Wiley and Sons, New York, 1984.

[25] Takács, L., "Priority Queues", *Op. Res. 12* (1964), pp. 63–74.

[26] Wang, Z., "User-Share Differentiation (USD): Scalable bandwidth allocation for differentiated services", draft-wang-diff-serv-usd-00.txt, November 1997.

[27] W. Willinger, M. Taqqu, R. Sherman, D. Wilson, "Self-similarity through high variability: Statistical analysis of Ethernet LAN traffic at the source level", IEEE/ACM Trans. Networking, Dec. 1996.

[28] W. Willinger, M. Taqqu, A. Erramilli, "A bibliographical guide to self-similar traffic and performance modeling"m in *Stochastic Networks*, F. P. Kelly, S. Zachary, and I. Ziedins (eds.), Oxford University Press, 1997.