Differentiated Services Networks: Pricing and Software Agents[1]

Peter Marbach

Department of Computer Science
10 King's College Road
University of Toronto
Toronto, ON
M5S 1A4
Canada
e-mail: marbach@cs.toronto.edu

Abstract: We consider packet-based networks that use the Differentiated Services (DiffServ) proposal of the Internet Engineering Task Force (IETF) to provide quality of service. For our study, we model DiffServ as a priority service. Users are free to choose the priority that they attach to their packets, however are charged accordingly. For this service, one would expect that users will choose an allocation (of priorities to packets) to satisfy their service requirements at the lowest possible cost. In this paper, we propose such an allocation strategy and study how it can be implemented by software agents. The paper is experimental, where we use numerical studies to investigate and illustrate this problem.

[1]This research was supported by a contract with Alcatel Bell, Belgium.

1 Introduction

In this paper, we study how pricing can be used for resource allocation in differentiated services (DiffServ) networks [1]. We model DiffServ as a priority service, and consider the situation where users are free to choose the priority of their data packets, but are charged accordingly by the network. We proposed an allocation strategy which could be implement through software agents and allows users to dynamically update the allocation of packets to priorities to optimize their own net benefit. We investigate the properties of the resulting algorithms through simulations.

DiffServ has been proposed by the Internet Engineering Task Force (IETF) as a mechanism to provide Quality-of-Service (QoS) in packet-based networks (such as the Internet) [1]. The basic idea behind DiffServ is the following. Networks deploying DiffServ offer a (small) number of different service classes. Users accessing the network assign (mark) their traffic to one of these classes, and the network forwards packets at each node according to class-specific rules (called per-hop behaviors) [13]. This mechanism can be used to provide differentiated QoS. For example, DiffServ could be used to implement a premium and a best-effort service, where premium traffic always gets preferential treatment over best-effort traffic [14]. As a result, premium traffic will obtain a better QoS when congestion in the network increases, i.e. premium traffic experiences less delay and is less likely to be dropped. A range of other service classes are also possible and have been proposed [7, 8, 9]. An important feature of DiffServ is that by aggregating a multitude of flows into a small number of service classes, DiffServ eliminates the need to recognize and store information about each individual flow in core routers; this makes DiffServ scalable.

In the DiffServ framework, it is natural to assume that users are charged based on the service classes that they use, i.e. a high-quality (premium) service is more expensive than a low-quality (best-effort) service. Also, one would expect that users choose service classes according to their service requirements, i.e. users try to meet their service requirements at the lowest possible cost. In this paper we analyze this situation. In particular, we are interested in the question whether the resulting interaction among users lead to a stable operation of the network.

For our analysis, we model DiffServ as a priority service. At each node, packets are served according to a priority rule, so that packets with high priorities will experience less delay and are less likely to be dropped. Users are free to choose the priorities for their traffic, but are charged accordingly by the network. Naturally, high priority packets will be more expensive than low priority packets. The fact that users are charged for every submitted (rather than only for transmitted) packets, makes this scheme easy to implement: the network forwards packets according to their priorities, and there is no need to track dropped packets (or link congestion levels), and send pricing information back to the users. We will show that charging users for all submitted packets will provide the right incentives (back-pressure) for users to reduce their rate during periods of congestion. An additional important feature of this model is that the prices the network charges for the different priorities are static and do not change (on a fast time-scale) to respond to changes in the network load; this makes the implementation of this scheme much simpler compared with dynamic pricing. However users can change their allocation dynamically (on a fast time-scale) to maximize their net benefit from using the network. These features allow to keep the operation of the network as simple as possible ("dumb networks"), and pushing the complexity towards the edge of the network ("smart users/end hosts").

For our study, we adopt an economic framework where we characterize the users' service requirements by utility functions. Roughly, submitting more high priority packets will increase the utility of an individual user. However, doing so will also increase the cost the user has to pay to the network provider. It is then natural to assume that users will choose an allocation of packets to priorities to maximize their net benefit, i.e. their utility minus cost. As the traffic load in packet-based networks, such as the Internet, is constantly changing due to arrival, and departures, of users, users will need to constantly monitor the congestion levels in the different priority classes, and to adapt their allocation accordingly. The goal of this paper is to formulate, and study, a simple allocation strategy that users may adopt to do this. In particular, we are interested in allocation mechanisms that can be implemented through software agents.

The above priority scheme does not prescribe a particular allocation strategy that customers have to employ for updating their allocation in response to changes in the traffic load of the network, or how to start up a new session. This is different from the flow control scheme of the Transmission Control Protocol (TPC) currently deployed in the Internet. Under TCP, all users must change/adapt their transmission rate according to a predefined rule; in particular, all users start up a new connection by slowly increasing their rate. Compared with TCP, the above priority scheme has the advantage that it is more flexible and allows customers to choose a strategy which suits best their service requirements. For example, users with delay sensitive, real-time traffic may want to ramp up their transmission rate quickly (to achieve a fast response time), whereas users with delay-insensitive traffic may slowly increase their rate. However as we will show, this could potentially lead to oscillations in the aggregate network load and packet transmission probabilities. While users with QoS-insensitive traffic might be able (or willing) to tolerate this, users with QoS-sensitive traffic might be adversely affected by such oscillations. One focus of this paper is to investigate this issue, and to study whether there exists a simple mechanism which allows users with QoS-sensitive traffic to isolate themselves from such oscillations.

The paper extends the work in [12], where the above priority scheme was modeled as a non-cooperative game. For the single link case, it is shown in [12] that there exists a (static) equilibrium for the corresponding game, as well as how pricing decision affect the properties of an equilibrium. In addition, it is shown in [12] that there exists a simple relation between link congestion and the prices of the different priority classes. However, the work in [12] does not provide or investigate a dynamic allocation strategy under which users actually converge to such an equilibrium.

The general framework that we consider here is closely related to the Paris Metro Pricing (PMP) proposal [15, 16] for providing QoS in the Internet. PMP partitions the network into several logically separated channels where channels differ in the prices paid for using them. This can be implemented by using the priority scheme, combined with pricing, as described above. However, [15, 16] do not analyze the existence of an equilibrium and the role of pricing. [3] provides simulation results to demonstrate that priorities can be an efficient tool for providing QoS in packet-based networks. In particular, [3] illustrates that using priorities, combined with pricing, achieves a higher a social welfare than a best-effort/flat-rate scheme. Gupta et. al. investigate in [5] a priority service similar to the one proposed here, however they consider the situation where the network dynamically changes the prices associated with the different priority classes in order to track a socially optimal allocation. This approach is theoretically appealing, but expensive (complex) to implement. The scheme considered in this paper is simpler as prices associated with the different service classes are fixed. Park et. al. also consider in [2, 17] a service where packets get marked according to customer's QoS requirements. However, there is no price associated with the different traffic classes, i.e. the costs incurred to customers are purely performance related. They study the situation where users are free to choose their traffic class [17], as well as where users indicate their QoS requirements and a network controller assigns network resources [2].

The priority service that we consider here can be interpreted as an approximation to the smart market proposal of MacKie-Mason and Varian [10, 11]. While in the smart market users can attach any bid $u \in \Re_+$ to their packets, in the above priority service users are limited to choose a bid $u \in \{u_1, ..., u_N\}$. Furthermore, while in the smart market users pay only for packets that get transmitted, in the above priority service users are charged for all submitted packets. Finally, while in the smart market users are charged a price that is equal to the highest bid among the dropped packets, here users are charged the bid price that they attached to their packets. Despite these differences, which make the priority service described above much easier to implement than the smart market, we will show that users (with elastic traffic) pay the same price per transmitted packet, and achieve the same average throughput, under the two service mechanisms.

The rest of the paper is organized as follows. After defining the situation that we are going to study in this paper in Section 2, we present in Section 3 the traffic and link model that we use for our analysis. In Section 4 we study the case where all users accessing the link measure quality of service solely as a function of the average throughput they obtain, define the non-cooperative game that we use to model the interaction among users, and carry out a static stability analysis by showing that there always exists an equilbirium allocation. In Section 5, we consider the case where users dynamically update their allocation of packet to priorities to maximize their net benefit, and propose a dynamic allocation strategy. In Section 6, we investigate the proposed algorithm using a case study.

2 Problem Formulation

Consider a packet-based network which offers a finite set $\mathcal{I} = \{1, ..., N\}$ of priority levels, where 1 is the lowest priority, 2 the second lowest, and so on. Users are free to choose the priorities for their traffic, but are charged accordingly by the network. In particular, the network charges a price u_i per unit traffic submitted with priority i, where for $i, j \in \mathcal{I}$ we have

$$u_i < u_j, \qquad \text{when } i < j.$$

At each node, packets are served according to a strict priority rule, so that high priority packets are less likely to be dropped during periods of congestion. A user accessing the network can then decide on

(a) its transmission rate, i.e. the total number of packets per unit time, and

(b) its traffic allocation, i.e. the priorities of the submitted packets.

For those decision, users have the following incentives. On one hand, users will try to submit as many high priority packets as possible as to achieve a high transmission rate and quality of service. On the other hand, users will try to keep their cost low and submit as few high priority packets as possible. This means that users face a trade-off between achieving a high transmission rate and quality of service, and keeping their cost low. In the following we are going to model and analyze this trade-off.

Consider a given user denoted by r and assume that user r submits packet in priority i, $i = 1, ..., N$, at a rate $d_r(i)$. Furthermore, suppose that $P_{tr}(i)$ is the probability that a priority i packet of user r gets transmitted through the network (and is not lost). As packets are served according to a priority rule, we will have that

$$P_{tr}(i) \leq P_{tr}(j), \qquad \text{when } i < j.$$

The average throughput of user r is then given by

$$x_r = \sum_{i=1}^{N} d_r(i) P_{tr}(i).$$

Furthermore, suppose the user r perceives quality of service only as a function of the average throughput. For this case, we can characterize user r by a quality indicator (utility function) which depends on the average throughput: the higher the throughput, the higher the utility the of user r. This type of user is often called a user with elastic traffic. Examples of users with elastic traffic are users sending email, transferring files, and browsing the Web. Let $U_r(x_r)$ be the utility of user r under the average throughput. We can then model the above trade-off by the following maximization problem

$$\max_{d_r \in \Re^N} \left\{ U_r(x_r) - \sum_i d_r(i) u_i \right\},$$

where

$$x_r = \sum_{i=1}^{N} P_{tr}(i) d_r(i).$$

This means that given the transmission probabilities $P_{tr}(i)$, $i = 1, ..., N$, user r wants to find a balance between achieving a high utility and keeping the cost low, i.e. an allocation of packets to priorities that optimizes its own net benefit, which is given as utility minus cost.

Note that the solution to the above maximization problem will depend on the transmission probabilities $P_{tr}(i)$, which themselves depend on the overall traffic load in the network. Therefore, users will have to monitor the current traffic conditions (and estimate the transmission probabilities in the different priority classes), and then update the traffic allocations to optimize their net benefit. Of course it is desirable, this task would not be performed by the user itself, but be automated through some software agent.

Under this scenario, a population of software agents (one for each user/application) will simultaneously monitor the network, estimate the transmission probabilities in the different priority classes, and update their own allocation accordingly. Note that whenever a software agent changes its own allocation, this will affect the transmission probabilities in the network, which then may trigger other software agents to change their allocation.

Therefore, we end up with a dynamical system of software agents. Naturally, one is then interested in the question, whether such a system will lead to a stable operation of the network, or whether the above interaction will lead to an unstable operation where the overall network load oscillates. There are two types stability analysis one can carry out.

(a) Static: one can ask the question whether there exists an equilibrium for the above situation, i.e. a set of allocation, one for each user, such that under this allocation each user optimizes its net benefit, and therefore no user has the incentive to change their allocation. Note, once we reach such an equilibrium allocation and, no user has the incentive to change their allocation, and we will stay in equilibrium.

(b) Dynamic: one can study whether there exists a strategy for updating allocations which converges to an equilibrium allocation.

In the following, we wish to address these questions. In order to do that, we have to describe more precisely how users interact.

3 Traffic and Link Model

In this section we will describe the traffic and service model that we are going to use for our analysis. We will only consider the single link case.

We use a discrete-time model, where time is divided into slots. Let $C > 0$ be a given constant, and consider a single link with a fixed capacity of C packets per time slot, where packets can be arbitrarily small. There is no buffering and packets which do not get transmitted in a given time slot are dropped. The link supports a finite set $\mathcal{I} = \{1, ..., N\}$ of priorities. Assume that R users share the link, and assume that each user submits packets at a constant and deterministic rate. In particular, let $d_r = (d_r(1), ..., d_R(N)) \in \Re^N_+$, $r = 1, ..., R$, be the allocation of user r, where $d_r(i)$ is the number of priority i packets user r submits in each time slot, and let $(d_1, ..., d_R)$ be the corresponding allocation vector. Furthermore, let $d = (d(1), ..., d(N))$ be the aggregated allocation (over all users), where

$$d(i) = \sum_{r=1}^{R} d_r(i), \qquad i = 1, ..., N,$$

is the total number of packets submitted in priority i in each time slot. The link then serves packets as follows (see Figure 1 for an illustration). In each time slot, all packets with priority i such that

$$\sum_{k=i}^{N} d(k) < C$$

are transmitted. For priority i^* with

$$\sum_{i=i^*+1}^{N} d(i) < C \quad \text{and} \quad \sum_{i=i^*}^{N} d(i) \geq C,$$

only $\left(C - \sum_{i=i^*+1}^{N} d(i)\right)$ packets, which are chosen at random, get transmitted. All packets with priority $i < i^*$ are dropped. For the aggregated allocation $d \in \Re^N_+$, the (packet)

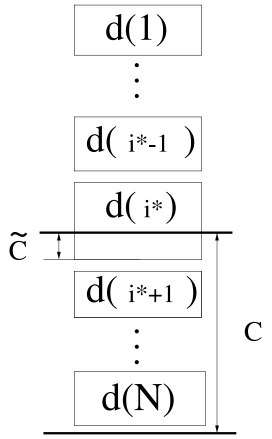

Figure 1: The link transmits all packets with a priority higher than i^*. From the packets with priority i^*, only $\tilde{C} = \left(C - \sum_{i=i^*+1}^{N} d(i)\right)$ packets are transmitted, which are chosen at random. All other packets are dropped.

transmission probability $P_{tr}(i, \tilde{d})$ in priority i is then given as follows,

$$P_{tr}(i, d) = \begin{cases} 1 & \text{if } C > \sum_{k=i}^{N} d(k), \\ \frac{C - \sum_{k=i+1}^{N} d(k)}{d(i)} & \text{if } \sum_{k=i}^{N} d(k) \geq C > ... \\ & > \sum_{k=i+1}^{N} d(k), \\ 0 & \text{otherwise.} \end{cases}$$

Let $P_{tr}(\tilde{d}) = (P_{tr}(1, \tilde{d}), ..., P_{tr}(N, \tilde{d}))$ be the corresponding probability vector. The average throughput of user r is then given by

$$x_r = \sum_{i=1}^{N} d_r(i) P_{tr}(i, d), \qquad r = 1, ..., R.$$

Users are charged a price u_i per submitted packet with priority i, where for $i, j \in \mathcal{I}$, $i < j$, we have $0 < u_i < u_j$. In addition, there exists a feedback mechanism that allows users to detect whether their packets were transmitted and to estimate the packet transmission probabilities in each priority.

The above model describes a priority service which charges a fixed price u_i is per submitted packet with priority i. However the price for actually transmitting a packet will depend on the total demand: in order to transmit a packet in priority i, users have to submit a packet (on average) $1/P_{tr}(i, d)$ times and the (average) price for transmitting a packet in priority i is then equal to $u_i/P_{tr}(i, d)$. In the following, we study this service.

4 Static Analysis

In this section, we carry out the static analysis for the above service. In Subsection 4.1, we consider first the case where a only one users optimizes the net benefit. In Subsection 4.2, we study the situation where all users simultaneously try to maximize their own net benefit. We model this situation as a non-cooperative game. For an introduction to the basic concepts of game theory we refer to [4]. Most results that we state in this section have already been presented in [12]; whenever this is the case, we will state the results without proofs.

4.1 The Single User Problem

Consider a fixed user r and assume that $d = (d(1), ..., d(N))$ is the current aggregate allocation vector, and that $P_{tr}(i, d)$, $i = 1, ..., N$, is the probability that a packet in priority i gets transmitted under the allocation d. Furthermore, assume that user r is able to (exactly) detect these transmission probabilities, and tries to find an allocation $d_r = (d_r(1), ..., d_r(N))$ that maximizes its own net benefit under this transmission probabilities, i.e. user r solves the following maximization problem

$$\max_{d_r \in \Re^N} \left\{ U_r(x_r) - \sum_i d_r(i) u_i \right\},$$

where

$$x_r = \sum_{i=1}^{N} P_{tr}(i, d) d_r(i)$$

is the average throughput under the allocation d_r. In addition, we have the following assumption.

Assumption 1 *For each user $r = 1, ..., R$, the function $U_r : \Re_+ \mapsto \Re_+$, is increasing, bounded, strictly concave, and twice differentiable.*

Assumption 1 implies that for every user $r = 1, ..., R$, the function $U_r : \Re_+ \mapsto \Re_+$, can be written as

$$U_r(x_r) = \int_0^{x_r} G_r(x) dx,$$

where $G_r(x) : \Re_+ \mapsto \Re_+$ is bounded, strictly decreasing and differentiable with

$$G_r(0) > 0 \quad \text{and} \quad \lim_{x \to \infty} G_r(x) = 0.$$

As we will show below, the function G_r can be interpreted as the inverse demand of user r with utility U_r. Note that Assumption 1 does not require that all users have the same utility function.

We introduce some more notation. For $r = 1, ..., R$, let the function $D_r : \Re_+ \to \Re_+$ be defined such that $D_r(u)$ is the optimal solution to the maximization problem, i.e. we have

$$D_r(u) = \arg\max_{x \geq 0} \{U_r(x) - xu\}, \quad u \in \Re_+.$$

Under assumption 1, the above maximization problem has a unique solution. $D_r(u)$ can be interpreted as the demand function of user r, i.e. $D_r(u)$ is the number of packets users r submits when the price per packet is u and all packets are transmitted with probability

1. Note that the function $G_r(x)$ that we defined above is the inverse of the function $D_r(u)$. Finally, we use $D(u)$, $u \in \Re_+$, given by

$$D(u) = \sum_{r=1}^{R} D_r(u),$$

to denote the aggregated demand over all users under the price u. In the following, we will extensively use refer to the demand function D_r, and the inverse demand function G_r, of user r, as well as the total demand D.

We then have the following result.

Proposition 1 *Let Assumption 1 hold and let*

$$u_{r,\max} = G_r(0).$$

Furthermore, let $d^* \in \Re_+^N$ be a given aggregate allocation and let the priority $i^* \in \{1, ..., N\}$ be such that

$$P_{tr}(i, d^*) = 0, \text{ for all } i < i^*, \quad \text{and} \quad P_{tr}(i, d^*) > 0, \text{ for all } i \geq i^*.$$

Let $d_r^* \in \Re_+^N$ be the allocation of a given user $r \in \{1, ..., R\}$, and let $x_r^* = \sum_{i=1}^N P_{tr}(i, d^*) d_r^*(i)$ be the corresponding average throughput. Then $d_r^* \in \Re_+^N$ is an optimal solution to the maximization problem

$$\max_{d_r \in \Re_+^N} \left\{ U_r(x_r) - \sum_{i=1}^{N} d_r(i) u_i \right\},$$

where

$$x_r = \sum_{i=1}^{N} P_{tr}(i, d^*) d_r(i),$$

if and only if the following holds.

(1) For $i \notin \{i^*, i^*+1\}$, we have $d_r^*(i) = 0$

(2) When $d_r^*(i^*) > 0$, then $G_r(x_r^*) = \frac{u_{i^*}}{P_{tr}(i^*, d^*)}$ and $P_{tr}(i^*, d^*) \geq \frac{u_{i^*}}{u_{i^*+1}}$.

(3) When $d_r^*(i^*+1) > 0$, then $G_r(x_r^*) = u_{i^*+1}$ and $P_{tr}(i^*, d^*) \leq \frac{u_{i^*}}{u_{i^*+1}}$.

(4) When $d_r(i^*) = d_r(i^*+1) = 0$, then $u_{r,\max} \leq \frac{u_{i^*}}{P_{tr}(i^*, d^*)}$ and $u_{r,\max} \leq u_{i^*+1}$.

Furthermore, when $d_r^* \in \Re_+^N$ is an optimal solution to the above maximization problem, then we have that

(5) When $P_{tr}(i^*, d^*) > \frac{u_{i^*}}{u_{i^*+1}}$, then $d_r(i^*+1) = 0$.

We provide a proof for this proposition in Appendix A

Using Proposition 1 an optimal allocation d_r^* for user r under the transmission probabilities $P_{tr}(i, d)$, $i = 1, ..., N$, is then given as follows. Let the priority $i^* \in \{1, ..., N\}$ be such that

$$P_{tr}(i, d^*) = 0, \text{ for all } i < i^*, \quad \text{and} \quad P_{tr}(i, d^*) > 0, \text{ for all } i \geq i^*.$$

We distinguish between the following three cases. When $i^* = N$, then set

$$u^* = \frac{u_{i^*}}{P_{tr}(i^*, d)},$$

and choose

$$d_r^*(i) = \begin{cases} D_r(u^*) \frac{1}{P_{tr}(i^*,d)}, & i = N, \\ 0 & \text{otherwise.} \end{cases}$$

When $i^* < N$ and $P_{tr}(i^*, d^*) \geq u_{i^*}/u_{i^*+1}$ then set

$$u^* = \frac{u_{i^*}}{P_{tr}(i^*, d)},$$

and choose

$$d_r^*(i) = \begin{cases} D_r(u^*) \frac{1}{P_{tr}(i^*,d)}, & i = i^*, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, when $i^* < N$ and $P_{tr}(i^*, d^*) < u_{i^*}/u_{i^*+1}$ then set

$$u^* = u_{i^*+1},$$

and choose

$$d_r^*(i) = \begin{cases} D_r(u^*), & i = i^* + 1, \\ 0 & \text{otherwise.} \end{cases}$$

4.2 Formulation as a Non-Cooperative Game

In this subsection, we study the case where all users simultaneously try to maximize their own net benefit. We model this situation as a non-cooperative game.

For the analysis, we assume that users ignore how they influence the transmission probabilities when optimizing their own net benefit. This simplifying assumption corresponds to the standard competitive price-taking assumption of economic theory. In this context, it can be justified when

(a) many "small" users access the link so that the influence of an individual user on the transmission probabilities is negligible.

(b) it is impossible (or too expensive) for a customer to determine how the choice of an allocation may affect the transmission probabilities.

We then define an equilibrium allocation as follows.

Definition 1 *Let $(d_1^*, ..., d_R^*)$ be a given allocation vector such that $d_r^* \in \Re_+^N$, $r = 1, ..., R$, and let $d^* = (d^*(1), ..., d^*(N))$,*

$$d^*(i) = \sum_{r=1}^{R} d_r^*(i), \qquad i = 1, ..., N,$$

be the corresponding aggregated allocation. We call $(d_1^*, ..., d_R^*)$ an equilibrium allocation when for every user $r = 1, ..., R$, we have

$$d_r^* = \arg \max_{d_r \in \Re_+^N} \left\{ U_r(x_r) - \sum_{i=1}^{N} d_r(i) u_i \right\},$$

where

$$x_r = \sum_{r=1}^{R} d_r(i) P_{tr}(i, d^*).$$

Note that in the maximization problem given in the Definition 1, the transmission probabilities to compute the average throughput x_r do not depend on the decision variable d_r, but only on d^*; this reflects the assumption that users ignore how they influence the transmission probabilities when optimizing their own net benefit. Under an equilibrium allocation $(d_1^*, ..., d_R^*)$, the allocation d_r^*, $r = 1, ..., R$, maximizes the net benefit of user r. Therefore, no user has an incentive to change its allocation d_r^* as this would decrease the net benefit. This implies that once an equilibrium is reached, no user will deviate.

In the following, we show that there always exists an equilibrium allocation (Proposition 3). In addition, we characterize an equilibrium allocation; in particular, we show how pricing decisions influence the properties of an equilibrium (Proposition 2). For the analysis, we assume that the demand in the lowest priority level exceeds the link capacity and the total demand $D(u_i)$ for priority $i \in \mathcal{I}$ is strictly positive. The case where the link capacity exceeds the the demand in the lowest priority level leads to the (trivial) situation where all users choose in equilibrium the rate $x_r = D_r(u_1)$, $r = 1, ..., R$, and all packets are transmitted with probability 1. When the total demand $D(u_i)$ in priority i is equal to zero, then priority i will not be used in equilibrium, and we can ignore this priority.

Assumption 2 *We have*

$$D(u_1) > C$$

and

$$D(u_i) > 0, \qquad \text{for } i = 1, ..., N.$$

In the next proposition, we will derive the necessary conditions for an allocation vector to be an equilibrium allocation. This will allow us to characterize an equilibrium allocation and determine how pricing decisions influence the properties of an equilibrium.

Proposition 2 *Let Assumption 1 and 2 hold. Consider an allocation vector* $(d_1^*, ..., d_R^*)$ *and let*

$$d^* = \sum_{r=1}^{R} d_r^*$$

be the corresponding aggregated allocation. Furthermore, let

$$x_r^* = \sum_{i=1}^{N} d_r^*(i) P_{tr}(i, d^*)$$

be the average throughput of user r, $r = 1, ..., R$ *under* $(d_1^*, ..., d_R^*)$ *, and define*

$$u_{N+1} = \max_{r=1,...,R} G_r(0).$$

Then the following holds. When $(d_1^*, ..., d_R^*)$ *,*$d_r \in \Re_+^N$, $r = 1, ..., R$, *is an equilibrium allocation, then there exists a priority* $i_0 \in \{1, ..., N\}$ *such that, for* $r = 1, ..., R$, *we have*

(a) $d_r^*(i) = 0$, *for all* $i \notin \{i_0, i_0 + 1\}$;

(b) $P_{tr}(i_0, d^*) \geq \frac{u_{i_0}}{u_{i_0+1}}$;

(c) $x_r^* = D_r(u^*)$ *where*

$$u^* = \frac{u_{i_0}}{P_{tr}(i_0, d^*)}.$$

Note that u_{N+1} is lowest price such that the total demand vanishes, i.e. we have that $D(u) > 0$, for $u \in [0, u_{N+1})$ and $D(u) = 0$, for $u \geq u_{N+1}$. Using Proposition 2, we can characterize an equilibrium allocation as follows.

(a) states that, under every equilibrium allocation, at most two priorities are used. This result has the following interpretation. Let $(d_1^*, ..., d_R^*)$ be a given allocation vector, let $d^* = \sum_{r=1}^{R} d_r^*$ be the corresponding aggregated demand, and let the priority i^* be such that

$$P_t(i, d^*) = 0, \text{ for all } i < i^*$$

and

$$P_t(i, d^*) > 0, \text{ for all } i \geq i^*.$$

Under the link model given in Section 3, packets with a priority lower than priority i^* will never be transmitted. As users are charged for every submitted packet, it is not economical to use these priorities. Furthermore, packets with a priority higher than $i^* + 1$ will be transmitted with the same probability as packets with priority $i^* + 1$, but at a higher cost; and these priorities won't be used either.

(b) states that when priority i is used under a given equilibrium allocation $(d_1^*, ..., d_R^*)$, then the probability that a submitted packet with priority i gets transmitted is lower-bounded by u_i/u_{i+1}. This result has the following interpretation. In equilibrium, the average price for transmitting a packet in priority i is equal to $u_i/P_{tr}(i, d^*)$. When

$$P_{tr}(i, d*) < \frac{u_i}{u_{i+1}},$$

then the average price for transmitting a packet in priority i is higher than for priority $i + 1$, i.e. we have that

$$u_i/P_{tr}(i, d^*) > u_{i+1}.$$

In this case it is more economical for users to switch their packets from priority i to priority $i + 1$, and therefore, Property (b) will hold in equilibrium.

(c) implies that the average throughput of user r is

$$x_r^* = D_r(u^*),$$

where

$$u^* = \frac{u_{i^*}}{P_{tr}(i^*, d^*)}.$$

Here, u^* can be interpreted as the average price users pay for transmitting a packet; and the average throughput of user r is equal to the demand D_r under u^*. This result then suggests that the pricing scheme is fair in the sense that all users receive a throughput according to the same price u^*.

Based on Proposition 2, one can derive sufficient conditions for an $(d_1^*, ..., d_R^*)$ to be an equilibrium allocation; which can then be used to construct an equilibrium allocation and to obtain the following result.

Proposition 3 *Let Assumption 1 and 2 hold. Then there exists an equilibrium allocation. Furthermore, when there exists a priority $i_0 \in \mathcal{I}$ such that*

$$D(u_{i_0}) > C > D(u_{i_0+1}),$$

then the equilibrium is unique.

Proposition 3 states that we have a unique equilibrium allocation when there exists a priority $i_0 \in \mathcal{I}$ such that

$$D(u_{i_0}) = \sum_{r=1}^{R} D_r(u_{i_0}) > C > \sum_{r=1}^{R} D_r(u_{i_0+1}) = D(u_{i_0}).$$

When there exists a priority i_0 such that

$$D(u_{i_0}) = C,$$

then it can been shown that there is an infinite number of equilibrium allocations

4.3 Discussion

Property (b) of Proposition 2 implies that we can use pricing to lower-bound the transmission probabilities in each priority. This result suggests that pricing can be used for congestion control through a similar mechanism as in the congestion avoidance algorithm of TCP. Under both schemes, users receive a (congestion) feedback which indicates whether their packets have been transmitted or not. However, while TCP assumes that users will reduce their rate when packet loss is detected, the priority service that we study here uses pricing to ensure that users indeed react appropriately to congestion. In this sense, this priority service provides a congestion avoidance mechanism that can be implemented in a non-cooperative environment where users behave in a selfish manner and are only interested in maximizing their own net benefit. Note that this (monetary) back-pressure is obtained because users are charged for every submitted packet.

Property (c) and (d) of Proposition 2, imply that there exists a unique price u^* such that for every user $r = 1, ..., R$, the average throughput is given by

$$x_r^* = D_r(u^*).$$

It is interesting to observe that the price u^* is equal to the congestion price users would pay under the smart market proposal by MacKie-Mason and Varian [10, 11]. Therefore, despite the differences between the smart market and the priority service that we consider here (see Section 1), users pay (in equilibrium) the same (congestion) price per (transmitted) packet, and obtain the same average throughput.

5 Allocation Strategies

In the previous section, we carried out a static analysis and showed that there always exists an equilibrium and characterized how pricing decisions influence the characteristic of an equilibrium. In this section, we go a step further and consider the situation where users dynamically update their allocation. We will propose a specific dynamic allocation strategy, and study through experiments whether the allocations under this strategy will converge to an equilibrium allocation.

Let us first consider the following allocation strategy. Suppose that users constantly monitor the transmission probability in each priority class. This could be done, for example, by submitting a small number of packets in each priority class to probe the network. Having obtained the transmission probabilities for a given time slot, users then choose for the next time slot an allocation which is optimal (i.e. maximizes the net benefit) for the measured transmission probabilities. As discussed in the previous section, such an optimal allocation can easily be computed. Given that the transmission probabilities do not change by too much between time slots, one would then expect that this strategy closely tracks an optimal allocation. Unfortunately, this might not be true. Moreover, when all users adopt this strategy, the resulting aggregate allocation might oscillate. The following example illustrates this.

Example 1 *Consider a link with capacity C which supports two priority classes $i = 1, 2$, and let the integer k, $= 1, 2, ...$, be the indicator of the time slot. Suppose that R users access the link, let D_r, $r = 1, ..., R$, the demand function of users r, and assume that we have that*

$$\sum_{r=1}^{R} D_r(u_2) < C$$

and

$$\frac{C}{\sum_{r=1}^{R} D_r(u_1)} < \frac{u_1}{u_2}.$$

Consider the time slot $k = 1$, let $d_r^1 = (d_r^1(1), ..., d_r^1(N))$, $r = 1, ..., N$, be the allocation of user r in this time slot, let d^1 be the corresponding aggregate allocation, and assume that

$$\sum_{r=1}^{R} d_r^1(1) = 0 \quad \text{and} \quad \sum_{r=1}^{R} d_r^1(2) < C.$$

The transmission probabilities in time slot $k = 0$ are then given by

$$P_{tr}(1, d^1) = P_{tr}(2, d^1) = 1.$$

Using the result of the previous section, to maximize the net benefit under the transmission probabilities $P_{tr}(1, d^1)$ and $P_{tr}(2, d^1)$, each each user will submit in time slot $k = 2$ $D_r(u_1)$ packets with priority 1 and no packets with priority 2. Using the assumptions on the demand functions D_r, $r = 1, ..., R$, it follows that in time slot $k = 2$ we have that

$$P_{tr}(1, d^2) = \frac{C}{\sum_{r=1}^{R} D_r(u_1)} < \frac{u_1}{u_2},$$

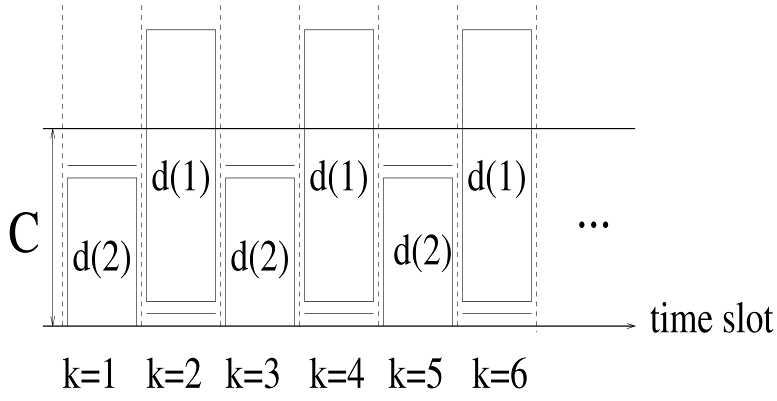

Figure 2: The aggregate allocation oscillates as at time $k = 1, 3, 5, ...$ all users use only priority class 2, and at time $k = 2, 4, 6....$ all users use only priority class 1.

and

$$P_{tr}(2, d^2) = 1.$$

This implies that for the (expected) cost $u_1/P_{tr}(1, d^2)$ for transmitting a packet in priority 1 is larger than for transmitting a packet in priority 2. To maximize the net benefit, in time slot $k = 3$, each user r will submit $D_r(u_2)$ packets with priority 2, and no packets with priority 1. By assumption, we have

$$\sum_{r=1}^{R} D_r(u_2) < C,$$

and in time slot $k = 3$ we encounter the same situation that we had at time $k = 1$; this cycle will repeat itself (see Figure 2) leading to oscillations.

An alternative strategy for dynamically adapting the allocation is the following. Instead of immediately switching to an allocation that maximizes the net benefit for the measured transmission probabilities in a given time slot, users could "slowly" (or "gradually") update their allocation along a direction that improves their net benefit. In the next subsection, we propose such an algorithm.

5.1 A Gradient Algorithm

In this subsection, we propose an algorithm for dynamically updating the user's allocation. We use $k = 1, 2, ...$ as an indicator for the time slots.

For our formulation, we assume that at the end of each time slot $k = 1, 2, ...$, users are able to exactly determine the transmission probabilities $P_{tr}(i, d^k)$, $i = 1, ..., N$, and use this information to update their allocation. Let $d_r^k = (d_r^k(1), ..., d_r^k(N))$ be the allocation of user r in time slot k, and let $d^k = (d^k(1), ..., d^k(N))$ be the aggregated allocation over all users. We assume that user's use a local search algorithm, where they update the number of packets submitted with priority i as follows

$$d_r^{k+1}(i) := d_r^k(i) + \alpha_r f_{r,i}(d_r^k, P_{tr}(d^k)),$$

where $\alpha_r > 0$ is a small step size parameter, and $f_{r,i}(d_r^k, P_{tr}(d^k))$ is a (projected) gradient direction of the net benefit with respect to $d_r^k(i)$. The direction $f_{r,i}(d_r^k, P_{tr}(d^k))$ is then given as follows

$$f_{r,i}(d_r, P_{tr}(d)) = \begin{cases} 0 & \text{if } d_r(i) = 0 \text{ and } G_r(x_r)P_{tr}(i,d) - u_i < 0 \\ G_r(x_r)P_{tr}(i,d) - u_i & \text{otherwise} \end{cases}$$

where

$$x_r = \sum_{i=1}^{N} P_{tr}(i,d)d_r(i),$$

and $G_r(x)$ is the inverse demand function of user r. It follows that (for sufficiently small step sizes α_r), we have that

$$U(d_r^{k+1}, P_{tr}(d^k)) - \sum d_r^{k+1}(i)u_i \geq U(d_r^k, P_{tr}(d^k)) - \sum d_r^k(i)u_i$$

and the net benefit of user r with respect to $P_{tr}(d^k)$ tends to increase from time k to $k+1$.

Here, we assume that users are free to choose their step size parameter α_r. This has the advantage that it allows users to choose a step size that suits best their service requirements. However, it has the disadvantage that a "badly behaved" user, who chooses too large a step size, may causes oscillations in the aggregate allocation.

6 Simulation Results

In this section, we study the above algorithm using simulations. In particular, we wish to investigate whether under the above local search algorithm

(a) the allocations converge to an equilibrium allocation, and

(b) pricing can be used to control link congestion.

For our experiments, we use the link model of Section 3. In addition, we assume that at the end of each time slot, users are able to exactly determine the transmission probability for each priority class. This simplifying assumptions will allows us to eliminate effects caused by measurement noise and feedback delay, and to focus on the principle characteristic of the algorithm.

We use the following set-up. We consider a link that supports three different priorities $i = 1, 2, 3$. The prices for submitting a packet in the different priority classes is equal to $u_1 = 1$, $u_2 = 2$, and $u_3 = 3$, respectively. In addition, for all experiments we assume that four users $r = 1, ..., 4$ access the link and consider the scenarios where

(1) all users have the same utility function,

(2) users have different utility functions, and

(3) a user causes oscillations in the aggregate allocation by choosing too high a step size parameter.

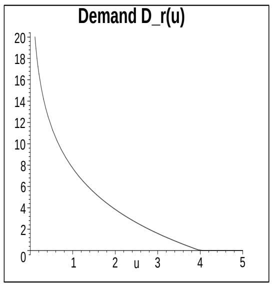

Figure 3: Demand $D_r(u)$, as a function of the price u, associated with the utility function U_r of user $r = 1, ..., 4$.

6.1 Users with Identical Utility Functions

For our first experiment, we consider the case where all four users have the same utility function U_r. Figure 3 illustrates the corresponding demand function $D_r(u)$, $r = 1, ..., 4$. We assume that all user start with an allocation $d_r^1 = (0, 0, 0)$; i.e. users don't submit any packets at time $k = 1$. Setting the step size parameters equal to $\alpha_1 = 0.45$, $\alpha_2 = 0.50$, $\alpha_3 = 0.60$, and $\alpha_4 = 0.75$, and the link capacity equal to $C = 18$ units, we obtain the simulation results given in the Figures 4 and 5. We make the following observations:

(1) The allocations converge to an equilibrium allocation, where each user obtains an average throughput of 4.5 (see Figure 4), which is 1/4 of the link capacity of $C = 18$; as predicted by Proposition 2.

(2) Under the equilibrium allocation, all user submit only packets in priority 1, and the probability that a packet of priority 1 gets transmitted is equal to $0.56 > u_1/u_2 = 0.5$ (see Figure 5).

(3) Throughout the updates, the transmission probability for priority 1 stays above 0.5, the ratio u_1/u_2 (see Figure 5).

For this example, the algorithm proposed above converges to an equilibrium allocation. In addition, pricing allows not only to control link congestion level in equilibrium, but also in the transient phase.

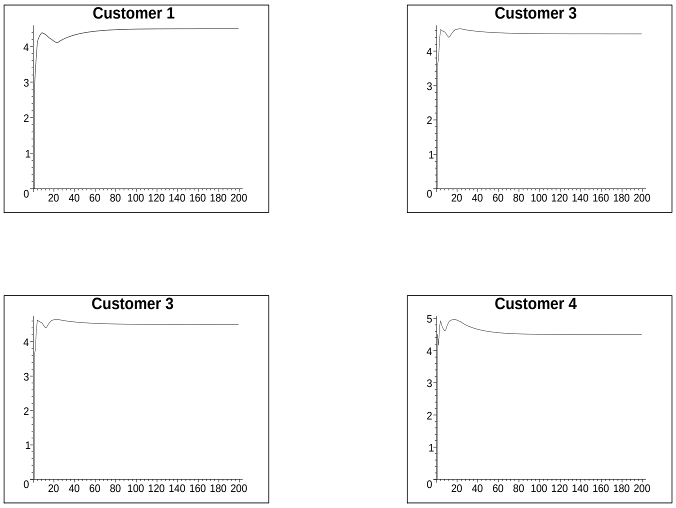

Figure 4: Trajectory of the average throughput of user 1,2,3, and 4. After 200 iteration steps, the average throughput converges to $x_1 = x_2 = x_3 = x_4 = 4.5$.

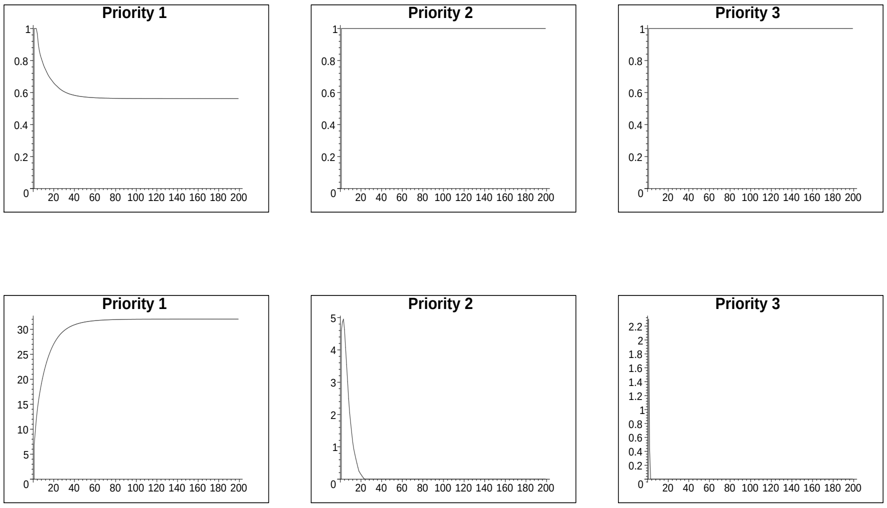

Figure 5: Trajectory of the transmission probabilities in the different priority classes (top row), and the total number of packets submitted in each priority class (bottom row).

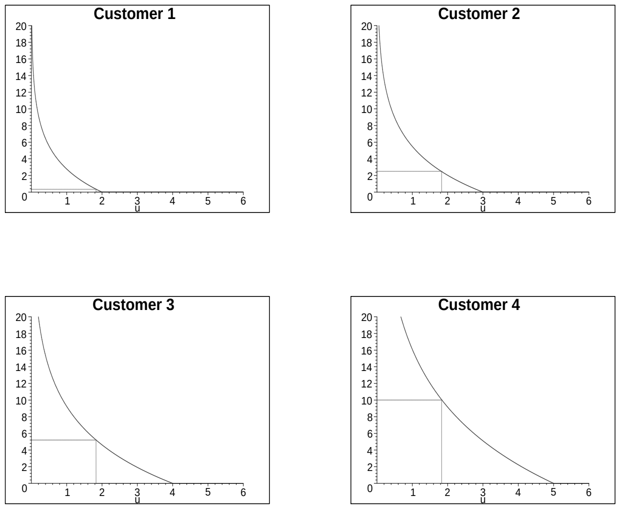

Figure 6: Demand $D_r(u)$, as a function of the price u, associated with the utility function U_r of user $r = 1, ..., 4$. All users see the same congestion price $u^* = 1.83$, but obtain a different average throughput of $x_1 = 0.3$, $x_2 = 2.5$, $x_3 = 5.2$, and $x_4 = 10.0$, respectively.

6.2 Users with Different Utility Functions

For the second experiment, we assume that users have a different utility functions. Figure 6 illustrates the corresponding demand functions $D_r(u)$, $r = 1, ..., 4$. Again, we assume that all user start with an allocation $d_r^1 = (0, 0, 0)$ at time $k = 1$. We use the same step size parameters as in the previous experiment: $\alpha_1 = 0.45$, $\alpha_2 = 0.50$, $\alpha_3 = 0.60$, and $\alpha_4 = 0.75$; and set the link capacity again equal to $C = 18$. Figure 7 gives the results we obtained. We make the following observations:

(1) The allocations converge to an equilibrium allocation where users obtain average throughputs equal to $x_1 = 0.3$, $x_2 = 2.5$, $x_3 = 5.2$, and $x_4 = 10.0$, respectively (see Figure 7).

(2) Although users obtain achieve different average throughputs, they all have the same "congestion price", which is equal 1.83 (see Figure 6).

(3) Again, pricing allows not only to control link congestion level in equilibrium, but also in the transient phase.

This experiment illustrates under an equilibrium allocation all users see the same average cost u^* for transmitting a packet, but may obtain a different average throughput depending on their utility functions.

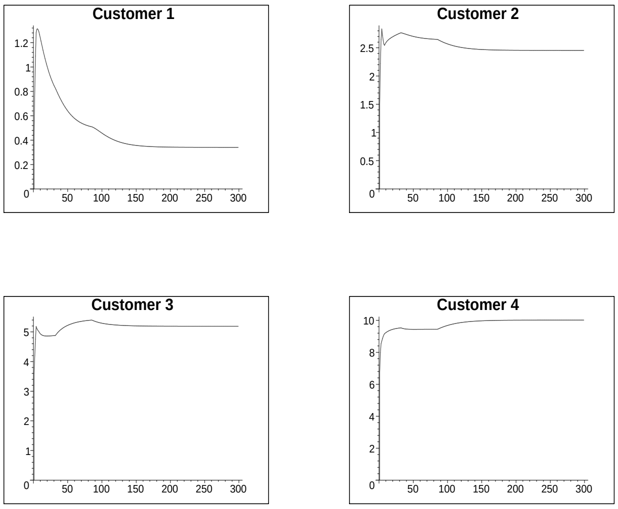

Figure 7: Trajectory of the average throughput of user 1,2,3, and 4. After 300 iteration steps, the average throughput converges and is equal to $x_1 = 0.3$, $x_2 = 2.5$, $x_3 = 5.2$, and $x_4 = 10.0$, respectively.

6.3 Congestion Control Through Pricing

In this experiment, we use the same set-up as in the first experiment, but reduce the link capacity from $C = 18$ to $C = 15$. We do this to investigate whether, in this situation, pricing can be used to control link congestion in the transient phase. Figure 8 illustrates the results we obtain. We make the following observation. Throughout the simulation, the transmission probability for packets submitted with priority 1 stays above $u_1/u_2 = 1/2$, and above $u_2/u_3 = 2/3$ for packets with priority 2 (see Figure 8). At the moment where the transmission probability for priority 1 falls below $1/2$, all users immediately switch their packets to the next higher priority 2. This examples illustrates that pricing is indeed an effective in controlling link congestion.

6.4 Avoiding Oscillations

In the previous experiments, we have chosen the step size parameters in such a way, that the allocations converge smoothly to an equilibrium allocation. However, as the scheme allows users to freely choose their step size parameter, a user might decide to chose a too large a step size parameter, causing oscillations in the aggregate allocation. While some users might be able (or willing) to tolerate this, others might be adversely affected by the resulting fluctuations in the expected transmission rate. Here, we want to study if there exists a simple algorithm which allows users to "isolate themselves" from such oscillations. The basic idea is the following. Based on the analysis in Section 4, the probability that a submitted packet with priority i gets transmitted should be lower-bounded by $\frac{u_i}{u_{i+1}}$. Therefore, when in a time slot the transmission probability for priority i falls below u_i/u_{i+1}, then this can be used as an indication that the aggregated allocation of

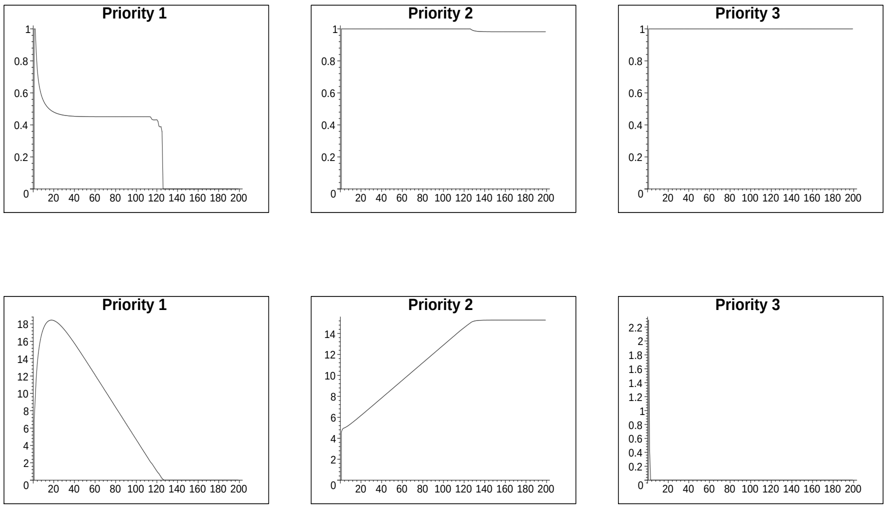

Figure 8: Trajectory of the transmission probabilities in the different priority classes (top row), and the total number of packets submitted in each priority class (bottom row). At the point where the transmission probability in priority class 1 falls below $u_1/u_2 = 0.5$, all users immediately switch to priority 2.

priority i oscillates. Consequently, users who wish to avoid oscillations could switch their packets to the next higher priority class. This leads us to consider the following "oscillation avoidance strategy".

Oscillation Avoidance Strategy
For time slot k, let i^* be such that, $P_{tr}(i^*, d^k) > 0$ and $P_{tr}(i, d^k) > 0$, for all $i < i^*$; let $f_{r,i}(d_r^k, P_{tr}(d^k))$, $i = 1, \ldots N$ be as in Section 5.1; and let $u_{\max} = \max_{r=1,\ldots,R} G_r(0)$.

FOR $i = 1$ TO N DO $d_r^{k+1}(i) := d_r^k(i) + \alpha_r f_{r,i}(d_r^k, P_{tr}(d^k))$
IF $i^* < N$ THEN
    IF $P_{tr}(i^*, d^k) < \frac{u_{i^*}}{u_{i^*+1}}$ THEN
        $d_r^{k+1}(u^*) := 0$
        $x_r = \sum_{i=1}^N d_r^{k+1}(i) P_{tr}(i, d^k)$
        IF $x_r > D_r(u_{i^*+1}$ THEN
            $d_r^{k+1}(i^* + 1) := \max\{0, D_r(u_{i^*+1}) - \sum_{i=i^*+2}^N d_r^{k+1}(i)$
        END
    END
ELSE IF $P_{tr}(i^*, d^k) < \frac{u_N}{u_{\max,r}}$ THEN
    $d_r^{k+1}(N) := 0$
END

We test this algorithm using the same set-up as in the first experiment, except that the step size parameter of user $r = 0.40$ is now $\alpha_4 = 25.0$. Figure 9 shows that this causes the expected transmission rate of all users to oscillate. We then repeat the experiment, where

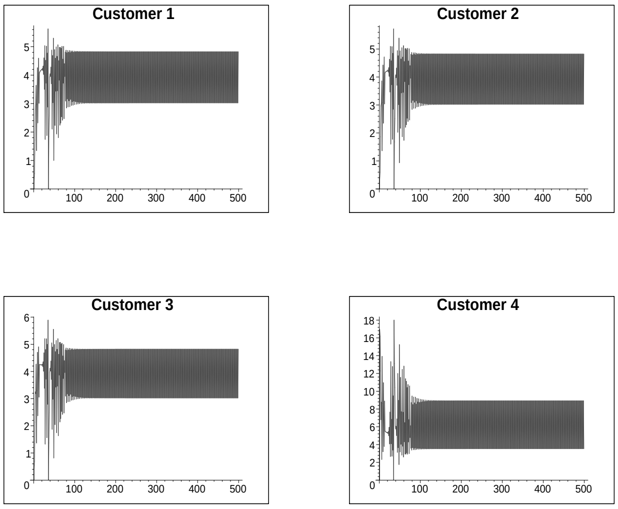

Figure 9: Trajectory of the average throughput of user 1,2,3, and 4, respectively.

user 1 now adopts the above proposed oscillations avoidance strategy. As a result, users 1 receives a much "smoother" rate than the other users (see Figure 10); however, at the expense of submitting packets with priority 2 and paying the corresponding higher price u_2 (see Figure 11).

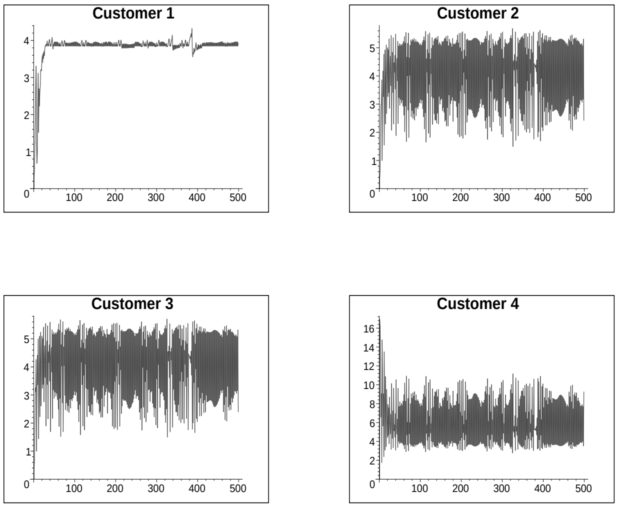

Figure 10: Trajectory of the average throughput of user 1,2,3, and 4, respectively. User 1 adopts the "oscillations avoidance strategy" and obtains a much smoother expected transmission rate.

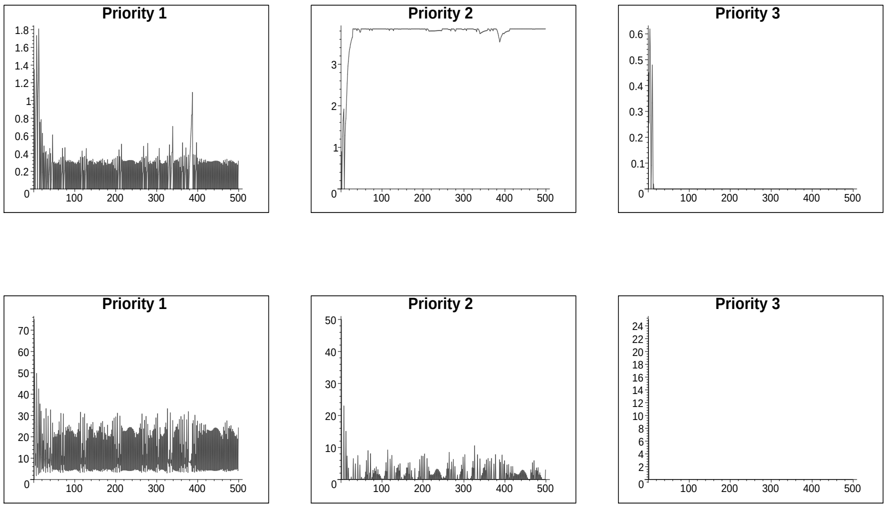

Figure 11: Trajectory of the allocation of packets to priorities of user 1 (top row), and user 2 (bottom row). User 1 submits most packets in priority class 2 to avoid the oscillations in priority 1.

7 Conclusions

The goal of this paper was to analyze the role of pricing in Differentiated Services (DiffServ) networks. For our study, we modeled DiffServ as a priority service, and assumed that users are free to choose the priorities they use and are charged accordingly. Too keep the analysis simple, we focused on the single link case, and used a bufferless, discrete-time link model. To capture the interaction among users, we modeled this situation as a non-cooperative game. For the case where only users with elastic traffic compete for the link capacity, we showed that there always exists an equilibrium allocation. Furthermore, this equilibrium allocation is unique, except for the (rare) case where there exists a priority $i \in \mathcal{I}$ such that the total demand $D(u_i)$ under the price u_i is equal to the link capacity. In addition, our analysis revealed a surprisingly simple relation between pricing decisions and the packet loss probabilities. In particular, when a packet with priority i is submitted under an equilibrium allocation $(d_1^*, ..., d_R^*)$, then the packet transmission probability $P_{tr}(i, d^*)$ is lower bounded by u_i/u_{i+1}, where $d^* = (d^*(1,), ..., d^*(N))$ is the corresponding aggregate allocation. This result suggest that in DiffServ networks, pricing is an effective tool to control congestion. In addition, we proposed an algorithm which allows users to dynamically update their allocation to optimize their net benefit, which we studied using simulations. The results of the experiments are promising as

(a) for a suitable choice of the step size parameters, the algorithm converges to an equilibrium allocation.

(b) pricing allows not only to control the congestion level in equilibrium, but also in the transient phase.

We assumed that customers are free to chose the step size parameters, and allowed the situation where a customers chooses too large a step size and causes oscillations in the aggregate allocation, and therefore in the packet transmission probabilities. For this situation, we proposed a simple algorithm which allows users to isolate themselves from such oscillations. This is important for users with tight quality of service requirements regarding packet loss and delay. As an alternative approach to avoid oscillations in the overall allocation, the network could assign customers a step size parameter, or define a range of admissible step sizes. This approach has the advantage that it allows to guarantee a stable network operation by suitably choosing the step size parameters. However in practice it might be (too) hard to efficiently choose step size parameters as the number of customers, and the corresponding traffic load, can vary significantly over time. As these changes are difficult to predict, one might have to settle for a conservative choice for the step size parameters which could result in a (prohibitive) slow convergence to an equilibrium allocation.

For future research, it would be interesting to study the theoretical properties of the algorithm that we proposed here, and to investigate its convergence properties. In addition, we would be interesting to carry a simulation study for the network case, involving a large set of users with different service requirements.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *"An Architecture for Differentiated Services* Internet Engineering Task Force, http://www.ietf.org/rfc/rfc2475.txt.

[2] S. Chen and K. Park, *"An Architecture for Noncooperative QoS Provision in Many-Switch Systems,"* Proc. IEEE INFOCOM '99, 1999.

[3] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang, *"Pricing in Computer Networks: Motivation, Formulation, and Example,"* IEEE/ACM Transactions on Networking, 1, 1993.

[4] R. Gibbons, Game Theory for Applied Economists, Princeton University Press, 1992.

[5] A. Gupta, D. O. Stahl, and A. B. Whinston,*"A Stochastic Equilibrium Model of Internet Pricing,"* Journal of Economics Dynamics and Control, 21:697–722, 1997.

[6] A. Gupta, D. O. Stahl, and A. B. Whinston, *"An Economic Approach to Network Computing with Priority Classes,"* Journal of Organizational Computing and Electronic Commerce, 6:71-95, 1996.

[7] F. Baker, J. Heinanen, W. Weiss, and J. Wroclawski, *"Assured Forwarding PHB Group,"* Internet Engineering Task Force, www.ietf.org/rfc/rfc2597.txt.

[8] J. Heinanen and R. Guerin, *"A Two Rate Three Color Marker,"* Internet Engineering Task Force, www.ietf.org/rfc/rfc2698.txt.

[9] V. Jacobson, K. Nichols, and K. Poduri, *"An Expedited Forwarding PHB,"* Internet Engineering Task Force, www.ietf.org/rfc/rfc2598.txt.

[10] J. K. MacKie-Mason and H. R. Varian, "Pricing Congestible Network Resources," *IEEE Journal of Selected Areas in Communications*, vol. 13, no. 7, pp. 1141-149, Sept. 1995.

[11] J. K. MacKie-Mason and H. R. Varian, *Pricing the Internet*, in Public Access to the Internet, Brian Kahin and James Keller, eds. pp. 269-314, Cambridge: MIT Press, 1995.

[12] P. Marbach,*"Pricing Priority Classes in a Differentiated Services Network"*, Proceedings of the 37th Annual Allerton Conference on Communications, Control, and Computing, 1999.

[13] K. Nichols and B.Carpenter, "Definition of Differentiated Services Per-domain Behaviors and Rules for their Specification,", Internet Engineering Task Force, draft-ietf-diffserv-pdb-def-00.txt

[14] K. Nichols, V. Jacobson, and L. Zhang, *"A Two-bit Differentiated Services Architecture for the Internet,"* Internet Engineering Task Force, www.ietf.org/rfc/rfc2638.txt.

[15] A. M. Odlyzko, *"Paris Metro Pricing: The minimalist differentiated services solution,"* Proceedings of the Seventh International Workshop on Quality of Service, pp. 159-161, 1999.

[16] A. M. Odlyzko, *"Paris Metro Pricing for the Internet,"* Proceedings of the ACM Conference on Electronic Commerce, pp. 140-147, 1999.

[17] K. Park, M. Sitharam, and S. Chen, *"Quality of Service Provision in Noncooperative Networks: Heterogenous Preferences, Multi-Dimensional QoS Vectors, and Burstiness,"* Proc. International Conference on Information and Computation Economies, 1998.

[18] S. Shenker, *"Fundamental design issues for the future Internet",* IEEE J. Selected Areas Comm., 13 (1995), 1176-1188, 1995.

A Proof of Proposition 1

We will show that Properties (1)-(4) of Proposition refl: nrt 1 are necessary and sufficient conditions for d_r^* to be an optimal solution for the maximization problem given in the proposition.

We use the following notation. Let $d_r^* \in \Re_+^N$ be as given in the statement of the lemma, and for $d_r \in \Re_+^N$ let the function $F_r(d_r, d^*)$ be given by

$$F_r(d_r, d^*) = U_r(x_r) - \sum_{i=1}^{N} d_r(i) u_i,$$

where

$$x_r = \sum_{i=1}^{N} P_{tr}(i, d^*) d_r(i).$$

Furthermore, for $r = 1, ..., R$, let $f_{r,i}(d_r, d^*)$ be the derivative of $F_r(d_r, d^*)$ with respect to $d_r(i)$, i.e. we have

$$f_{r,i}(d_r, d^*) = \lim_{\alpha \to 0} \frac{F_r(d_r + \alpha e_i, d^*) - F_r(d_r, d^*)}{\alpha}, \qquad i = 1, ..., N,$$

where e_i is the ith unit vector. These derivatives exist as, by Assumption 1, $U_r(x_r)$ is differentiable on $[0, \infty)$. In particular, we have

$$f_{r,i}(d_r, d^*) = U_r'(x_r) P_{tr}(i, d^*) - u_i,$$

where U_r' is the derivative of U_r (with respect to x_r). The first order conditions for $d_r \in \Re_+^N$ to be an optimal solution for the maximization problem given in the statement of the lemma are then given as follows,

$$f_{r,i}(d_r, d^*) \leq 0, \qquad \text{for } i = 1, ..., N, \tag{1}$$

where this equation holds with equality when $d_r(i) > 0$. In the following, we show these first order conditions hold, if and only if the Properties (1)-(4) of Lemma 1 hold. Furthermore, note that that by Assumption 1, we have that the utility function U_r is convex in d_r and therefore the first order conditions are sufficient conditions.

Let the aggregate allocation $d^* \in \Re_+^N$ and the priority i^* be as given in the statement of the Lemma, i.e. we have that

$$P_{tr}(i, d^*) = 0, \text{ for all } i < i^*, \quad \text{and} \quad P_{tr}(i, d^*) > 0, \text{ for all } i \geq i^*.$$

Note that, for $i < i^*$ we have

$$f_{r,i}(d_r, d^*) = -u_i < 0, \qquad d_r \in \Re_+^N.$$

Furthermore, for $i < i^* + 1$ we have

$$f_{r,i}(d_r, d^*) = G_r(x_r) - u_i < G_r(x_r) - u_{i^*+1} = f_{r,i^*+1}(d_r, d^*), \qquad d_r \in \Re_+^N.$$

This implies the following. When

$$f_{r,i}(d_r, d^*) \leq 0, \qquad \text{for } i = i^*, i^* + 1,$$

then we have that

$$f_{r,i}(d_r, d^*) < 0, \qquad \text{for } i \notin \{i^*, i^* + 1\}.$$

Combining this result with the first order conditions (see Equation (1)), we obtain that when $d_r^* \in \Re_+^N$ is an optimal solution, then we have that

$$d_r^*(i) = 0, \qquad \text{for } i \notin \{i^*, i^* + 1\}.$$

This establishes Property (1).

To satisfy Equation (1) for $i = i^*$ and $i = i^* + 1$, we need the following. When $d_r(i^*) > 0$, then we have that

$$f_{r,i^*}(d_r, d^*) = G_r(x_r) P_{tr}(i^*, d^*) - u_{i^*} = 0$$

and

$$f_{r,i^*+1}(d_r, d^*) = G_r(x_r) - u_{i^*+1} \leq 0.$$

These two equations imply that

$$G_r(x_r) = \frac{u_{i^*}}{P_{tr}(i^*, d^*)} \leq u_{i^*+1}$$

and

$$P_{tr}(i^*, d^*) \geq \frac{u_{i^*}}{u_{i^*+1}}.$$

This gives Property (2). Similarly, when $d_r(i^* + 1) > 0$, then we have that

$$G_r(x_r) = u_{i^*+1} \leq \frac{u_{i^*}}{P_{tr}(i^*, d^*)}$$

and

$$P_{tr}(i^*, d^*) \leq \frac{u_{i^*}}{u_{i^*+1}},$$

which gives Property (3).

Finally, when $d_r^* \in \Re_+^N$ is an optimal solution with $d_r(i^*) = d_r(i^*+1) = 0$, then Property (1) implies that $x_r = 0$. Therefore, we need the following to satisfy Equation (1),

$$f_{r,i^*}(d_r, d^*) = G_r(0) P_{tr}(i^*, d^*) - u_{i^*} \leq 0$$

and

$$f_{r,i^*+1}(d_r, d^*) = G_r(0) - u_{i^*+1} \leq 0.$$

Combining these two equations, we obtain that $u_{r,\max} = G_r(0) \leq \frac{u_{i^*}}{P_{tr}(i^*,d^*)}$ and $u_{r,\max} \leq u_{i^*+1}$. This gives Property (4).

So far, we have shown that when the first order conditions given by Equation (1) are satisfied, then the Properties (1)-(4) hold. Similarly, one can show that when the Properties (1)-(4) hold, then the first order conditions are satisfied. We omit a detailed derivation here.

To show that Property (5) holds, we proceed as follows. Assume that $P_{tr}(i^*, d^*) > \frac{u_{i^*}}{u_{i^*+1}}$. Then we have that

$$f_{r,i^*}(d_r, d^*) = G_r(x_r) P_{tr}(i^*, d^*) - u_{i^*} > G_r(x_r) - u_{i^*+1} = f_{r,i^*+1}(d_r, d^*),$$

and the first order conditions given by Equation (1) imply that $d_r(i^* + 1) = 0$.