

Differentiation and interaction of traffic: a flow level study

Eeva Nyberg and Samuli Aalto

Networking Laboratory
Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, Finland
{eeva.nyberg,samuli.aalto}@hut.fi

Abstract. We study what kind of differentiation can be achieved using DiffServ without admission control using a relative services approach, i.e. the rate of the flow should be in proportion to the contracted rate. We model analytically differentiation mechanisms proposed in such proposals as AF and Simple Integrated Media Access (SIMA) with emphasis on modelling the interaction between various differentiation mechanisms and traffic flows of type TCP and UDP. We first review main results of an earlier and more detailed packet level model and then introduce more abstract flow level models. As a result we show how the flow level models are powerful models for explaining how marking thresholds and mechanisms determine the differentiation in a network. The flow level models are able to produce results similar to the packet level models, and can be used as an abstraction of a DiffServ network and give system level guidelines for both network and differentiation design.

Keywords: DiffServ, proportional differentiation, TCP, flow level.

1 Introduction

New service concepts have been designed to overcome the problems of converged networks: delivering real time applications in the best-effort Internet. One of these concepts is Differentiated Services (DiffServ). DiffServ is termed a scalable architecture, as inside the network quality treatment is only offered to aggregate classes not to each individual traffic flow. The scalability is achieved at the price of losing strict quantitative guarantees and bounds on the level of service.

For the purpose of our evaluation, we divide traffic differentiation into two categories: *assured services* and *relative services*. In assured services the flow should receive a rate at least equal to the contracted rate, while in relative services the rate of the flow should be in proportion to the contracted rate.

We then study what kind of differentiation can be achieved using DiffServ without admission control. Specifically, previous work on DiffServ modelling and especially on the Assured Forwarding (AF) service proposal [4] have shown that assured services cannot be achieved without admission control, [13], [15], [7], [14], [3] and [12]. We challenge this claim by stating that a more valid service requirement in DiffServ is relative services, where the received rate does not

have to be equal to the contracted rate, but where the link bandwidth should be divided in proportion to the contracted rates.

In [13] and [15] the analytical relationship between the contracted rate and the transmission rate of TCP flows is studied. The works include a model for TCP and a model for the priority marker, under the assumption of deterministic loss rates for different priorities, without an explicit buffer model for how the loss probabilities depend on the TCP rates. In [7] a model for assured service that includes a buffer model with different discarding levels for different priorities is studied, but the paper does not include a TCP model nor a marking model for the traffic. A similar model is given in [14] and [8], with some considerations on modelling TCP flows, the paper however still lacks combining both a TCP model and a buffer model.

In a previous paper [10] and in [9] we give a detailed packet level model and a closed loop modelling of the dependency of the flow sending rate on the metering, marking, dropping and scheduling mechanisms. Furthermore, we studied the interaction between various differentiation mechanisms and traffic flows of type TCP and UDP, only done previously in simulation studies by [3] and [12].

In the present paper we continue the analytical modelling of differentiation mechanisms proposed in such proposals as AF and Simple Integrated Media Access (SIMA) [6] with emphasis on modelling the interaction between various differentiation mechanisms and traffic flows of type TCP and non-TCP, e.g. UDP. We first review the main results of the packet level model and then introduce flow level models to compare to our packet level models of [10]. The flow level model is a more abstract model than the detailed packet level model. On the flow level the emphasis is on modelling the interaction between traffic flows assuming some general models of the differentiation mechanisms, while the packet level models were based on detail models of the mechanisms themselves and the response of traffic flows to the feedback signals of the network.

The paper is organized as follows. In section 2 we discuss the marking, forwarding and discarding mechanisms employed for differentiating traffic. We then review the results for the packet level model, in section 3, and in section 4 present the flow level model. Section 5 compares the models through numerical results and section 6 concludes the paper.

2 Mechanisms for differentiating traffic

Throughout the paper we study how delay and priority aggregates can be used to achieve relative differentiation of flows and to protect traffic with different service requirements, e.g. delay and drops, from each other. We identify the main QoS mechanisms used to achieve differentiation in AF and SIMA and formulate a generic DiffServ architecture. We thus do not explicitly model existing DiffServ proposals, but study what mechanisms are needed at different parts of the network to achieve differentiation.

If the service model is of type assured services, then as long as the flow sends at a rate less than the assured rate, the flow is marked in profile and

thus only two or three priority levels are needed. If the service model is of type relative services, then the priority can be determined based on the ratio of the sending rate to the assured rate, thus the more priority levels, the more flexible the marking is in terms of covering a wider range of sending rates. The main difference between the service concepts is the division of excess capacity and the division of capacity in overload situations.

We divide the DiffServ mechanisms into two categories: classification and conditioning of flows at the boundary node and forwarding of flow aggregates through packet level mechanisms inside the DiffServ node. Figure 1 summarizes the components, each discussed separately below.

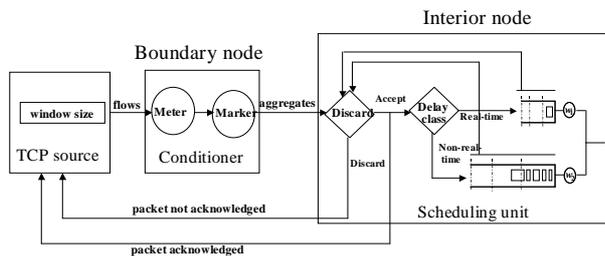


Fig. 1. Components of a DiffServ network including feedback to TCP sources

2.1 Conditioning flows at the boundary nodes

Consider a DiffServ network with a single bottleneck link, which is loaded by a fixed number of flows. We divide flows into two delay classes, real-time streaming traffic and non-real-time elastic traffic, $d = 1, 2$, respectively. We have I priority levels, $i = 1, \dots, I$. Level I refers to the highest priority, i.e. flows at that level encounter the smallest packet loss probability. Each flow is given a weight ϕ that reflects the value of the flow or the contracted rate of the flow, flows are grouped according to this weight. There are L^d different groups of flows of delay class d , each group l with a characteristic packet sending rate $\nu(l)$ and weight $\phi(l)$. Let \mathcal{L}^d denote the set of such flow groups. Finally, let $n(l)$ denote the number of flows in any group l . For TCP flows $\nu(l)$ depends on the network state, while for UDP flows $\nu(l)$ is fixed and does not change even if the network congestion level changes.

At the conditioner, the packets of a flow are marked and aggregated to priority levels. We adopt the proposal in [6], where the priority level $pr(l)$ of the flow depends on $\nu(l)$ and $\phi(l)$ as follows:

$$pr(l) = \min[i = 1, 2, \dots, I : \nu(l) \geq t(l, i)]. \quad (1)$$

Corresponding threshold rates would then be

$$\begin{aligned} t(l, 0) &= \infty, \\ t(l, i) &= \phi(l)a(i), \quad i = 1, \dots, I - 1, \\ t(l, I) &= 0, \end{aligned} \tag{2}$$

where, based on [6], $a(i) = 2^{I/2-i-0.5}$. Note that $a(i-1)/a(i) = 2$ for all i . Note also that we adopt here the relative services approach by defining that a flow sending at its contracted rate receives the middle priority and not the highest priority. We presented in [10] two **metering and marking alternatives** to mark flows to priority levels: *Token bucket* (TB) and *Exponential weighted moving average* (EWMA). Independent of the marking scheme we say that a flow has priority $i = pr$ as given by equation (1).

TB The token or leaky bucket principle, referred to, e.g. in the AF specification, is a popular metering principle. For three priority levels the metering and marking may be performed with two token buckets for each group l , with rates $t(l, 1) > t(l, 2)$ and capacities c . Traffic of the flow passes the two buckets sequentially, starting from the bucket $[t(l, 1), c]$, and is marked to highest priority if there are enough tokens in all the buckets, to middle priority if only in the first bucket there were enough tokens and to lowest priority if the traffic is out-of-profile already in the first bucket. We illustrated in [11], using simulations, that TB can be modelled as :

- *Per packet marking*: Only those packets of a flow that exceed the marking threshold are marked to the lower priority level. The ratio of packets that have priority i is

$$\frac{\min[\nu(l), t(l, i - 1)] - \min[\nu(l), t(l, i)]}{\nu(l)}. \tag{3}$$

Cascaded token buckets split the flow into sub-streams $i = pr, pr + 1, \dots, I$. All the packets of sub-stream i have the same mark i corresponding to the ratio given in equation (3).

EWMA The measurement results of previous time instants are taken into account, but exponentially dampened according to a time parameter α and the time interval that has elapsed since the measurement was done. The marking is then performed based on predefined thresholds on the resulting measured arrival rate. The measured bit rate $mbr(k, j)$ of a flow k at the moment of transmission of the j :th packet is given in [6], it can be derived from the traditional discrete EWMA updated at fixed intervals of length $1/\delta$ time slots, with time scale α/δ . The j :th packet of flow $k \in l$ has priority i , if

$$t(l, i) \leq mbr(k, j) < t(l, i - 1).$$

We have illustrated in [11] that EWMA can be modelled as :

- *Per flow marking*: Once the measured rate of a flow exceeds a marking threshold, all packets of the flow are marked to the same priority level.

All the packets of the flow then have the same mark pr given by equation (1).

2.2 Forwarding and discarding packets inside the DiffServ nodes

Bandwidth between delay aggregates must be divided in terms of delay requirements. Bandwidth between priority aggregates, on the other hand, must be divided in terms of packet loss probabilities and result in a division according to flow weights assuming that the relative services approach is adopted. Furthermore, this must be done across delay classes. Low latency classes should not starve bandwidth from the other classes and the elastic delay aggregate with high priority should not be discarded before the low latency aggregate with lower priority.

Discarding We have a system with two delay classes, serviced by two separate buffers, where the buffer sizes are chosen according to the delay requirements of the delay aggregates. Both buffers have I discarding thresholds, one for each priority class. Consider two different **discarding mechanisms**:

- *Independent discarding*: Each buffer acts locally as a separate buffer, discarding appropriate priority levels according to its buffer content.
- *Dependent discarding*: The content of both buffers determines which priority level is discarded, in both buffers.

For example, when one buffer is heavily loaded and the other is almost empty, *independent discarding* would discard traffic belonging to different priority levels in different buffers, while *dependent discarding* would discard traffic of high priority also from the buffer that is almost empty.

On the packet level, discarding can be modelled using a buffer model. Let m^d denote the number of packets in the buffer of delay class d . The *independent discarding* is implemented by giving, separately for each delay class d , thresholds $K^d(i)$, where $K^d(I) = K^d$, the size of the buffer, and $K^d(0) = 0$. The minimum priority level accepted is then $pr_a^d = f_d(\frac{m^d}{K^d})$. The *dependent discarding*, proposed in [6], is implemented by giving a two-dimensional monotonic function that determines the minimum priority level accepted when in state (m^1, m^2) , $pr_a = f(\frac{m^1}{K^1}, \frac{m^2}{K^2})$, e.g.

$$pr_a = a + b \cdot \left(\frac{m^1}{K^1} + \frac{m^2}{K^2} \right), \quad (4)$$

$$pr_a = a + b \cdot \sqrt{\left(\frac{m^1}{K^1} \right)^2 + \left(\frac{m^2}{K^2} \right)^2}. \quad (5)$$

We use equation (5) as a basis for the discarding function in the packet level model. Note that equation (4) is similar to the discarding in a system with only one buffer.

Scheduling The traffic that is not discarded is placed in either one of the two buffers. We restrict our analysis of scheduling mechanisms to the different weights possible in the Weighted Fair Queuing (WFQ) scheduling principle. The capacity of the link is divided according to predetermined weights w^1 and w^2 , with $w^1 + w^2 = 1$, unless one of the buffers is empty, as then the other buffer has use of total link capacity.

3 Packet level model for TCP flows

In [10] we presented a packet level model, where it was assumed that UDP traffic is sent at a constant rate $\nu(l)$, classified to appropriate priorities, resulting in aggregate arrival intensities $\lambda(i)$. The TCP flows on the other hand respond to the congestion in the network, e.g. loss probability feedback signal, by adjusting their sending rate. For elastic traffic we solve the resulting equilibrium intensity using the fixed point method. We briefly present the equations in parameterized form needed to solve the equilibrium rate $\nu(l, x_l)$, for elastic flows $l \in \mathcal{L}^{\text{TCP}}$. Defining $\mathbf{x} = \{x_l, l \in \mathcal{L}\}$, where x_l is an auxiliary variable used for parameterization. For a detailed discussion see [9].

The TCP flows are characterized by their round-trip time. Let $RTT(l)$ denote the round-trip time of flows in group $l \in \mathcal{L}^{\text{TCP}}$. Let $q(l, \mathbf{x})$ denote the packet loss probability of flow group l and $p^d(i, \mathbf{x})$ the packet loss probability for delay class d at priority level i . Assuming that the dynamics of the buffer is faster than that of TCP we can use the steady state TCP throughput expression of, e.g. Kelly [5]. Then the equilibrium throughput $\nu(\mathbf{x}) = \{\nu(l, x_l), l \in \mathcal{L}\}$ amounts to solving the fixed point equation

$$\nu(l, x_l) = \frac{1}{RTT(l)} \sqrt{2 \frac{1 - q(l, \mathbf{x})}{q(l, \mathbf{x})}}, \quad l \in \mathcal{L}^{\text{TCP}}. \quad (6)$$

where, using the *per flow* marking mechanism,

$$q(l, \mathbf{x}) = p^d(\lfloor pr(l, x_l) \rfloor, \mathbf{x})(\lfloor pr(l, x_l) \rfloor + 1 - pr(l, x_l)) + p^d(\lfloor pr(l, x_l) \rfloor + 1, \mathbf{x})(pr(l, x_l) - \lfloor pr(l, x_l) \rfloor), \quad l \in \mathcal{L}^d, \quad (7)$$

and using the *per packet* marking scheme,

$$q(l, \mathbf{x}) = \sum_{j=1}^I p^d(j, \mathbf{x}) \frac{\min[\nu(l, x_l), t(l, j-1)] - \min[\nu(l, x_l), t(l, j)]}{\nu(l, x_l)}, \quad l \in \mathcal{L}^d. \quad (8)$$

The loss probabilities $p^d(i, \mathbf{x})$ are solved for the one buffer case assuming an $M/M/1/K$ model. For the two buffer case they can only be given numerically, see [9].

The priority level is $pr(l, x_l) = I$ in the range $0 \leq x_l \leq 2$, and from there on the parameterized form is

$$pr(l, x_l) = \begin{cases} -x_l + I + 1 + i, & 2i \leq x_l \leq 2i + 1, \quad i = 1, \dots, I - 1 \\ I - i, & 2i + 1 \leq x_l \leq 2i + 2, \quad i = 1, \dots, I - 2. \end{cases}$$

For $x_l \geq 2I - 1$, we have $pr(l, x_l) = 1$.

Finally, the vector form of the arrival intensity of priority level i is

$$\lambda^d(\mathbf{x}) = \{\lambda^d(i, \mathbf{x}) \mid i = 1, \dots, I\},$$

where the arrival intensity of priority level i is with *per flow* marking

$$\lambda^d(i, \mathbf{x}) = \sum_{l \in \mathcal{L}^d: |pr(l, x_l) - i| < 1} n(l) \nu(l, x_l) (1 - |pr(l, x_l) - i|). \quad (9)$$

and with *per packet* marking

$$\lambda^d(i, \mathbf{x}) = \sum_{l \in \mathcal{L}^d: pr(l, x_l) \leq i} n(l) (\min[\nu(l, x_l), t(l, i - 1)] - \min[\nu(l, x_l), t(l, i)]). \quad (10)$$

4 Flow level model

In the previous section we gave a detailed packet level model including scheduling and discarding functions. Let us now study the resulting bandwidth division β assuming that the detail packet handling can be modelled on the flow level as a Processor Sharing mechanism, which divides capacity equally among all flows, $\beta = 1/n$. Then we are able to study the conditioning mechanisms by modelling how the introduction of priority levels i and flow groups l affect the actual bit rate of a flow, $\beta(l, i)$.

The flows with the highest mark I have, in our flow level model, a strict priority over all the other flows. Among these high priority flows, the bandwidth is divided as fairly as possible, i.e. each receives an equal share unless this exceeds the corresponding threshold rate.

4.1 Differentiation mechanisms and resulting bandwidth shares

One buffer Assume two TCP flow groups, $\phi(1) > \phi(2)$ and one buffer. The flows or substreams of flows in the same priority level share bandwidth equally up to their threshold rate. Because the threshold rate for group 1 is higher than for group 2, group 1 flows share also among themselves the extra bandwidth left over by group 2 flows in the same priority level. If *per flow* marking is used, then the general rule to determine the bandwidth shares $\beta(l, i)$ for all the $n(l, i)$ flows in group l and at level i is, cf. [1],

$$\begin{cases} \beta(1, i) = \min\{\max\{\frac{C(i)}{n(i)}, \frac{C(i) - n(2, i)t(2, i - 1)}{n(1, i)}\}, t(1, i - 1)\}, \\ \beta(2, i) = \min\{\frac{C(i)}{n(i)}, t(2, i - 1)\}, \end{cases} \quad (11)$$

where $C(I) = C = 1$ and

$$C(i) = \max\{C(i + 1) - n(1, i + 1)t(1, i) - n(2, i + 1)t(2, i), 0\}$$

refers to the remaining capacity for the flows with mark i , $n(i) = n(1, i) + n(2, i)$.

If *per packet* marking is used, we consider substreams of flows instead of entire flows, as a flow has substreams in different priority levels. The general rule to determine the bandwidth shares $\beta(l, i)$ is, cf. [1],

$$\begin{cases} \beta(1, i) = \min\{\beta(1, i+1) + \max\{\frac{C(i)}{s(i)}, \frac{C(i) - s(2, i)\delta(2, i-1)}{s(1, i)}\}, t(1, i-1)\}, \\ \beta(2, i) = \min\{\beta(2, i+1) + \frac{C(i)}{s(i)}, t(2, i-1)\}, \end{cases} \quad (12)$$

where $\beta(l, I+1) = 0$, $C(I) = C = 1$ and

$$C(i) = \max\{C(i+1) - s(1, i+1)\delta(1, i) - s(2, i+1)\delta(2, i), 0\}$$

refers to the remaining capacity for the substreams with mark i , $\delta(l, i) = t(l, i) - t(l, i+1)$, $s(l, i) = n(l, 1) + \dots + n(l, i)$ and $s(i) = s(1, i) + s(2, i)$.

Two buffers with dependent discarding Assume now that, instead of two TCP flow groups sharing the same buffer, we have one group in each delay class, so that group 1 consists of UDP flows, sending at a fixed rate of $\nu(l)$ and group 2 consists of TCP flows. Now the UDP flows in group 1 divide the remaining capacity among themselves, but never receive more than their boundary rate. The TCP flows in group 2 then divide among themselves the capacity that is left over by the flows in higher priority levels and by the UDP flows or substreams of the same priority level. Because we assume dependent discarding, UDP flows of lower priority levels than the TCP flows do not affect the bandwidth share of the TCP flows. The equations are then for *per flow* marking,

$$\begin{cases} \beta(1, i) = \min\{\frac{C(i)}{n(1, i)}, t(1, i-1), \nu(1)\}, \\ \beta(2, i) = \min\{\max\{\frac{C(i) - n(1, i)t(1, i-1)}{n(2, i)}, 0\}, t(2, i-1)\}, \end{cases} \quad (13)$$

where

$$C(i) = \max\{C(i+1) - n(1, i+1)t(1, i) - n(2, i+1)t(2, i), 0\}$$

as before.

If *per packet* marking is used, we again consider the substreams of flows. The equations are then of the form

$$\begin{cases} \beta(1, i) = \min\{\beta(1, i+1) + \frac{C(i)}{s(1, i)}, t(1, i-1), \nu(1)\}, \\ \beta(2, i) = \min\{\beta(2, i+1) + \frac{C(i) - s(1, i)\delta(1, i-1)}{s(2, i)}, t(2, i-1)\}, \end{cases} \quad (14)$$

where

$$C(i) = \max\{C(i+1) - s(1, i+1)\delta(1, i) - s(2, i+1)\delta(2, i), 0\}$$

as before. Note that now, $\delta(1, i) = \min\{\nu(1), t(1, i)\} - \min\{\nu(1), t(1, i+1)\}$ and $\delta(2, i) = t(2, i) - t(2, i+1)$.

4.2 Interaction of priority levels and bandwidth shares

To model the interaction between TCP and the marking mechanisms we need to determine the priority levels $pr(l)$ as a function of the number of flows $n(l)$ in each group l . These priority levels, in turn, determine uniquely the *network state* $\mathbf{n} = (n(l, i); l = 1, 2; i = 1, 2, \dots, I)$, namely

$$n(l, i) = \begin{cases} n(l), & \text{if } i = pr(l), \\ 0, & \text{if } i \neq pr(l). \end{cases}$$

As soon as the network state is known, the bandwidth shares can be calculated from equations (11) - (14), depending on the marking and buffer models used.

Consider a non-elastic CBR source using UDP that does not react to packet losses but tries to maximize the sending rate, then the bandwidth share does not necessarily equal the sending rate. The result will be that the packets have a low priority and high sending rate, but the bandwidth share of the flow may become very small. True TCP, instead, seems to behave smartly tending to optimize, in the first place, the bandwidth share of the flow and not the sending rate. It follows that the priority level of the UDP flows remains constant, whereas TCP sources experiment whether it is more useful to limit the sending rate and have a high priority or increase the sending rate and have a lower priority. More precisely, we set up a game between two flow groups with the following rules:

1. For UDP flows the priority levels are constant and determined from the sending rates. For TCP flows the initial priority levels are the highest ones, $pr(l) = I$.
2. The groups make decisions alternately.
3. Flows belonging to the UDP group do not adjust their sending rate, while at level $pr(l)$, TCP group $l \in \mathcal{L}^{\text{TCP}}$ decides to raise the level by one if $pr(l) < I$ and the resulting bandwidth share $\beta'(l, pr(l) + 1)$ is higher than the original one $\beta(l, pr(l))$. If the level is not raised, group l decides to lower the level by one if $pr(l) > 1$ and the resulting bandwidth share $\beta'(l, pr(l) - 1)$ is higher than the original one. All the bandwidth shares are calculated from equations (11) - (14).
4. The game ends whenever it is beneficial to all TCP groups to keep the current priority levels. The final bandwidth share $\theta(l)$ for a flow in group l will be $\beta(l, i)$ corresponding to the final level $pr(l) = i$.

In principle, it may happen that the game does not end, but remains in a loop consisting of a number of states. However, our numerical experiments suggest that such a unique final state is always achieved.

5 Numerical results

With the models presented above, we can now study, first on the packet level, how the sending rate of TCP flows affect the dropping probabilities of the network and vice versa, with the effect of UDP flows in the network taken into account.

We then compare the packet level results, to the flow level model. The interaction of traffic groups is modelled as a game, where TCP flows try to maximize the bandwidth share of the flow by adjusting their sending rate, UDP flows, on the other hand, have a constant sending rate. For the sake of comparison, we will also present flow level results for the case of two TCP flow groups.

In all cases, we evaluate the resulting differentiation under the different mechanism options based on the relationship between the contracted rate, i.e. the weight, of the flow and the share of bandwidth achieved by the flow belonging to a specific aggregate. Our main purpose is to study what mechanisms give elastic traffic, which adjusts its sending rate according to the state of the network, incentives to maximize primarily its bandwidth share and not its sending rate.

5.1 Packet level

Assume that the flows in group $l = 1$ are UDP flows that have a constant sending rate $\nu(1)$. Then the fixed point equation is only solved for the TCP flows in group $l = 2$. We study the ratio of achieved throughputs $\frac{\theta(1)}{\theta(2)} = \frac{\nu(1)(1-q(1))}{\nu(2)(1-q(2))}$ under three constant sending rates $\nu(1)$, chosen so that under *per flow* marking, the flows are in the highest (solid line), middle (gray line), and lowest (dashed line) priority $\nu(1) = 0.039$, $\nu(1) = 0.079$, and $\nu(1) = 0.16$, respectively. Our basic scenario for three priority levels is the following: $I = 3$, $\mu = 1$, $RTT = 1000/\mu$ and $(\phi(1), \phi(2)) = (0.08, 0.04)$, i.e. ratio of weights $k = 2$. Each set of pictures depicted in figures 2 (one buffer) and 3 (two buffers) show the ratio between throughputs of flows as a function of the total number of flows, under the condition $n(1)/n(2) = 1/2$.

One buffer with TCP and UDP traffic The case of one delay class and one buffer, with size $K = 39$ is depicted in figure 2. When the UDP traffic has the highest priority, there is no visible difference between the marking schemes and as long as $\nu(1) \leq 0.039$, there is no way of limiting its sending rate. For *per flow* marking we notice that the UDP flows do not always receive more bandwidth than the TCP flows. In fact, only when the fixed priority of the UDP flows is optimal in relation to the load of the network, the UDP flows receive more bandwidth than the TCP flows.

Two buffers with TCP traffic and UDP traffic For the two buffer model we can compare the result with the one delay class case, to see if our conjecture on the difference of marking schemes holds, and how the weighted capacity and discarding affects these results. These results for the two buffer case were presented in [10], and due to lack of space, we only replicate the comparison of discarding methods and marking alternatives for priority queuing ($w^1 = 1, w^2 = 0$) in figure 3. We notice that figure 2 depicting the same scenario for one delay class roughly corresponds to two delay classes with dependent discarding, i.e. in times of low load it is optimal to have low priority and in times of high load to have high priority. The TCP flow is able to adjust its priority to differ from

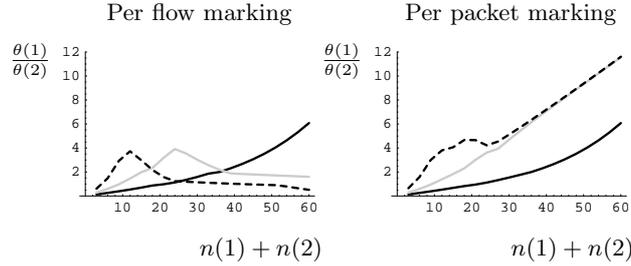


Fig. 2. Packet level model for one buffer with 66% TCP and 33% UDP traffic.

the priority of the aggressive flows and therefore only when the constant priority coincides with the load level of the network, does the aggressive flow get more than the TCP flow.

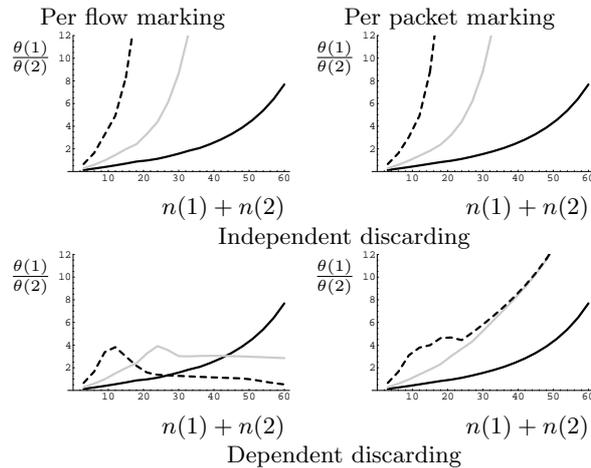


Fig. 3. Effect of marking and discarding for two buffer priority queueing 66% of the flows are TCP and 33% UDP.

When priority queueing is used to schedule traffic, only with *per flow* marking and dependent discarding TCP flows can be protected from bandwidth exhaustion by the UDP flows. It was shown in [10] that when independent discarding is used, the results are the same for one priority level and three priority levels, then bandwidth shares can only be controlled by the scheduler weights. However, there is no clear dependency between the ratio of weights of the scheduler (w^1/w^2) and the ratio of weights of the flow ($\phi(1)/\phi(2)$).

5.2 Flow Level

TCP and UDP traffic The flow level model seems to lead to similar results as the more detailed packet level model. Using both *per flow* marking and dependent discarding, gives a powerful incentive for the flows to be TCP friendly, whereas the combination of *per packet* marking and independent discarding, encourages the flows to behave selfishly. Figure 4 depicts this behavior.

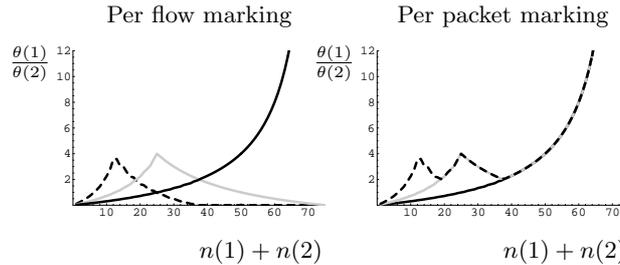


Fig. 4. Flow level model for two buffers with 66% TCP and 33% UDP traffic.

The packet level and flow level models differ quantitatively, but qualitatively the flow level model also shows that as the number of flows increases, i.e. the congestion of the network increases the bandwidth share for UDP flows is independent of their sending rate or priority level when *per packet* marking together with dependent discarding is used. With *per flow* marking and dependent discarding we again deduce that the elastic TCP flows gain from adjusting their rates to the network conditions as long as there are enough priority levels so that UDP traffic is not in the highest priority level.

From both packet level and flow level results, we notice that dependent discarding and *per flow* marking promote TCP friendliness [2]. When there are few users on the link, it is optimal for the streaming traffic to send at lowest priority $\nu(1) = 0.16$. As the number of users increases it would be optimal for the streaming traffic to drop its sending rate to, e.g. $\nu(1) = 0.079$, and as the number of users further increases it would be optimal to send at the highest priority level, with intensity $\nu(1) = 0.039$. Under all other mechanism, it is always optimal to send as much as possible, even if all or some of the packets are then marked to the lowest priority level.

TCP traffic As a final example, let us study, using flow level models, the bandwidth division in a link with only TCP traffic. Figure 5 shows the throughput ratios, when group 1 traffic is also TCP traffic.

In [9], we showed that also for the case of all groups having TCP traffic, the packet level and flow level models give similar results. In a graph, where the number of group 1 flows are on the x-axis and the number of flows of group 2 are on the y-axis, we can divide the plot of throughput ratios into 3–5 different

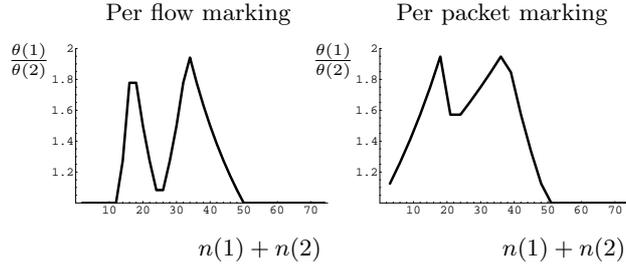


Fig. 5. Flow level model TCP flows, 66% are group 2 and 33% group 1 traffic.

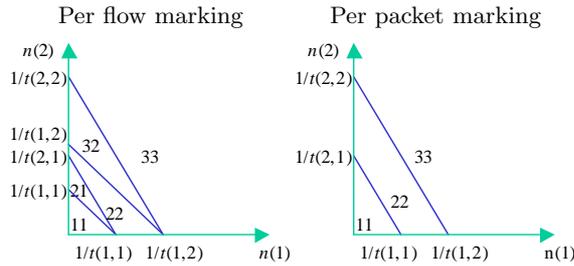


Fig. 6. The resulting priority levels ($pr(1), pr(2)$) as a function of the number of flows in the two groups, $n(1)$ and $n(2)$.

areas depicted in figure 6. The areas show how the TCP flows adjust their sending rate as the number of flows in the system change. They at the same time adjust their priority level to attain a better throughput. With per flow marking it is advantageous for flows in group 1 to move up in priority before flows in group 2. The flow model equations (11)–(14) thus give expressions that explains how and when the TCP flows adjust their sending rate.

6 Conclusions

Numerical results on both the packet level and flow level show that dependent discarding controls the throughput of non-responsive flows better than independent discarding. This is due to the fact that under the fixed threshold mechanism when the TCP buffer is congested, packets in the UDP buffer are also discarded to alleviate the congestion. Only when the UDP flows are sending at a rate low enough to attain highest priority are they able to use more than their fair share of the bandwidth. By having enough priority levels, i.e. more than three, this effect is also diminished. The use of *per flow* marking and dependent thresholds thus gives a powerful incentive for flows to adjust the sending rate according to the state of the network and thus be TCP friendly.

The paper also showed how the flow level models are powerful models for explaining how marking thresholds and mechanisms determine the differentiation

in a network. The flow level models are able to produce results similar to more detailed packet level models, and can be used as an abstraction of a DiffServ network and give system level guidelines for both network and differentiation design. Furthermore, being a simpler model, the flow level model can be used for more general network topologies, this we leave for an interesting further research topic.

References

1. S. Aalto and E. Nyberg. Flow level models of diffserv packet level mechanisms. In *Proceedings of the Sixteenth Nordic Teletraffic Seminar, NTS 16*.
2. S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, August 1999.
3. M. Goyal, A. Duresi, R. Jain, and C. Liu. Performance analysis of Assured Forwarding. IETF Draft October 1999.
4. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. *Assured Forwarding PHB Group*, June 1999. RFC 2597.
5. F. Kelly. Mathematical modelling of the Internet. In *Proc. of Fourth International Congress on Industrial and Applied Mathematics*, pages 105–116, 1999.
6. K. Kilkki. Simple Integrated Media Access. Available at <http://www-nrc.nokia.com/sima>, 1997.
7. M. May, J. Bolot, C. Diot, and A. Jean Marie. Simple performance models for Differentiated Services schemes for the Internet. In *Proceedings of IEEE INFOCOM*, pages 1385–1394, March 1999.
8. L. V. Nguyen, T. Eyers, and J.F. Chicaro. Differentiated service performance. In *Proceedings of Fifth IEEE Symposium on Computers and Communications ISCC 2000*, pages 328–333, 2000.
9. E. Nyberg. How to achieve fair differentiation: Relating flow level qos requirements to diffserv packet level mechanisms. Licentiate thesis, Helsinki University of Technology, Networking Laboratory, September 2002.
10. E. Nyberg and S. Aalto. How to achieve fair differentiation. In *Networking 2002*, Pisa, Italy.
11. E. Nyberg, S. Aalto, and R. Susitaival. A simulation study on the relation of diffserv packet level mechanisms and flow level QoS requirements. In *Intl. Seminar, Telecommunication Networks and Teletraffic Theory*, St. Petersburg, Russia, 2002.
12. P. Piedad, N. Seddigh, and B. Nandy. The dynamics of TCP and UDP interaction in IP-QoS Differentiated Service networks. In *3rd Canadian Conference on Broadband Research (CCBR)*, November 1999.
13. S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firoiu. On achievable service differentiation with token bucket marking for TCP. In *Proceedings ACM SIGMETRICS'00*, pages 23–33, June 2000.
14. S. Sahu, D. Towsley, and J. Kurose. A quantitative study of differentiated services for the Internet. In *Proc. IEEE Global Internet'99*, pages 1808–1817, 1999.
15. I. Yeom and A. L. N. Reddy. Modeling TCP behavior in a differentiated services network. *IEEE/ACM Transactions on Networking*, 9(1):31–46, 2001.