How to Achieve Fair Differentiation

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 present a simple packet level model to show how marking at the DiffServ boundary node and scheduling and discarding inside a DiffServ node affect the division of bandwidth between two delay classes: elastic TCP flows and streaming non-TCP flows. We conclude that only per flow marking together with dependent discarding thresholds across both delay classes is able to divide bandwidth fairly, according to the load of the network, and in a TCP friendly way.

Keywords: DiffServ, TCP, fairness, TCP friendliness

1 Introduction

The main arguments against differentiation are the waste of network resources and the difficulty to guarantee fair bandwidth allocation between priority classes. More research in this field has to be done, to be able to settle the dispute. The Internet research also lacks efforts in coupling the packet level QoS mechanisms of DiffServ [1], e.g. Assured Forwarding (AF) [2], to flow level analysis. On the other hand, flow level bandwidth allocation and fairness research, e.g. [3], [4], continue to assume that weighted fair bandwidth allocations between flows in different service classes are somehow achieved and evade the question of how to do so without flow control or per flow scheduling.

In [5] we introduced both packet and flow level models to study how bandwidth is divided among flows using packet level differentiation mechanisms of the Simple Integrated Media Access (SIMA) proposal [6]. In the present paper we continue the packet level modelling approach to investigate the key factors of two DiffServ schemes, AF and SIMA. Following Roberts [7], we assume two forwarding classes based on delay requirements: elastic TCP traffic and streaming non-TCP traffic. As a result, we present the role of the conditioning and forwarding mechanisms in dividing bandwidth consistently across delay classes.

2 DiffServ Network Model and Its Analysis

The main elements of DiffServ are traffic classification and conditioning at the boundary nodes and traffic forwarding through scheduling and discarding at the DiffServ interior nodes. In addition, congestion control mechanisms designed for

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1178–1183, 2002.

the Internet, such as TCP, and active queue management algorithms, such as RED, may be used for QoS in the Internet. Figure 1 summarizes the components.

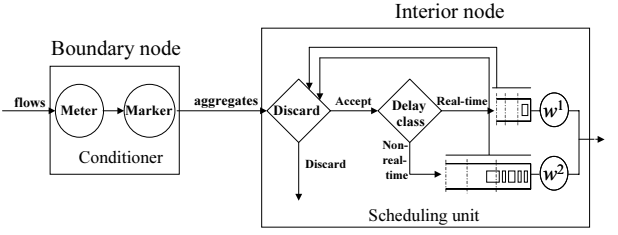

Fig. 1. Components of a DiffServ network

Network model. Consider a DiffServ network with a single bottleneck link, which is loaded by a fixed number of flows. Assume two delays classes, $d = 1, 2$, and I precedence levels, $i = 1, \ldots, I$. Delay class 1 refers to non-TCP flows, and delay class 2 to TCP-flows. Precedence level I refers to the highest priority, i.e. flows at that level encounter the smallest packet loss probability, and level 1 to the lowest priority. Note that this is just opposite to, e.g., the definition given in [2]. Therefore, we rather use the term priority level here.

Each flow is given a weight ϕ that reflects the value of the flow. A natural objective of any traffic control algorithm is to allocate bandwidth as fairly as possible. Here fairness refers to weighted fairness in a single link, i.e. the throughput θ of any flow should be proportional to its weight ϕ. For networks with DiffServ architecture it is not clear how to achieve this objective, since there are no per flow mechanisms available in the core network.

At the conditioner, the packets of a flow are marked to priority levels according to the measured traffic rate compared to the weight of the flow. More specifically, let ν denote the measured packet arrival rate of a flow. As in [6], we assume that the priority level pr of the flow depends on ν and ϕ as follows:

$$pr = \max\left[\min\left[\left\lfloor I/2 + 0.5 - \frac{\ln \frac{\nu}{\phi}}{\ln 2}\right\rfloor, I\right], 1\right]. \tag{1}$$

Thus, the priority level is decreased by one as soon as the traffic rate doubles.

For non-TCP flows we assume a fixed packet arrival rate ν, whereas for TCP flows it depends on the congestion level of the network. Let RTT denote the round trip time of a TCP flow and q the packet loss probability it encounters in the buffer of the bottleneck link. Following [8], we assume that

$$\nu = \frac{1}{RTT}\sqrt{2\frac{1-q}{q}}. \tag{2}$$

Assume that there are L^1 different groups of non-TCP flows, each group l with a characteristic packet arrival rate $\nu(l)$, and let $\mathcal{L}^1$ denote the set of such flow groups. Furthermore, assume that there are L^2 different groups of TCP flows, each group l with a characteristic round trip time $RTT(l)$, and let $\mathcal{L}^2$

denote the set of such flow groups. Finally, let $n(l)$ denote the number of flows in any group l.

At the boundary node all the traffic belonging to the same delay class and precedence level are aggregated. Let $\lambda^d(i)$ denote the aggregate packet arrival rate of delay class d and priority level i. Packets of the flow aggregates are then forwarded or discarded by a scheduling unit that includes two buffers, one for each delay class. Denote by K^1 and K^2 the sizes of the two buffers in number of packets.

DiffServ mechanisms. Traffic is conditioned at the boundary node by measuring the incoming traffic and, based on the metering result, by marking the packets of the flow. We consider two different **marking principles**:

- *Per flow marking:* Once the measured traffic rate of a flow exceeds a marking threshold, all packets of the flow are marked to the same precedence level.
- *Per packet marking:* Only those packets of a flow that exceed the marking threshold are marked to the lower precedence level.

The marking thresholds for flow group l, determined from (1), are $t(l,0) = \infty$, $t(l,I) = 0$, and

$$t(l,i) = \phi(l) \cdot 2^{I/2-i-0.5}, \ i = 1, ..., I-1. \tag{3}$$

Per flow marking gives the aggregate arrival intensity $\lambda^d(i)$ as

$$\lambda^d(i) = \sum_{l \in \mathcal{L}^d : pr(l) = i} n(l)\nu(l). \tag{4}$$

On the other hand, if *per packet* marking is applied, then

$$\lambda^d(i) = \sum_{l \in \mathcal{L}^d : pr(l) \leq i} n(l)(\min[\nu(l), t(l,i-1)] - \min[\nu(l), t(l,i)]). \tag{5}$$

But what are such **metering and marking mechanisms** that follow these principles? In [9] we demonstrated by simulation experiments that the *token bucket* scheme marks packets to precedence levels *per packet*, while the use of *exponentially weighted moving average* (EWMA) marks packets *per flow*. The token bucket scheme is referred to, e.g., in the AF specification. Packets are marked to I precedence levels by $I-1$ cascaded token buckets. The EWMA scheme was proposed, e.g., in the SIMA proposal.

Forwarding at the interior node is done to aggregates divided, in our case, into two delay classes. Before forwarding, traffic can be limited by discarding packets based on precedence levels. We consider two different **discarding mechanisms**:

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

Let m^d denote the number packets in the buffer of delay class d. The independent discarding is implemented by giving, separately for each delay class d, thresholds $K^d(i)$ that determine the minimum priority level accepted, PL_a, when compared to m^d. The dependent discarding, proposed in [6], is implemented by giving a two-dimensional monotonic function

$$PL_a = f(\frac{m^1}{K^1}, \frac{m^2}{K^2}) \tag{6}$$

that determines the minimum priority level accepted when in state (m^1, m^2). We apply the function introduced in [10].

The traffic not discarded is placed in the two buffers. Following the Weighted Fair Queuing (WFQ) principle, whenever one of the buffers is empty, the other buffer has use of total link capacity. Otherwise the capacity of the link is divided according to predetermined weights w^1 and w^2, with $w^1 + w^2 = 1$. We consider three different **scheduling scenarios**:

- *Priority queuing:* WFQ with weights $(w^1 = 1, w^2 = 0)$.
- *Unequal sharing:* WFQ with weights $(w^1 = 0.75, w^2 = 0.25)$.
- *Equal sharing:* WFQ, with weights $(w^1 = w^2 = 0.5)$.

Analysis. The scheduling unit with two buffers is modelled as two dependent $M/M/1/K$ queues with state dependent arrival intensities. When in state (m^1, m^2), the arrival intensity depends on the applied discarding function as follows: if PL_a is i, then the arrival rate for buffer d is $\lambda^d(i)+\ldots+\lambda^d(I)$. The packet transmission times are assumed to be exponentially distributed with mean $1/\mu$. Thus, if both buffers are non-empty, packet service rates are $w^1\mu$ and $w^2\mu$ for the two delay classes. This results in a two-dimensional Markov jump process, the stationary distribution of which can be solved numerically.

From the stationary distribution we can calculate the packet loss probabilities $p^d(i)$ for each traffic aggregate, i.e., for each combination of delay class d and priority level i. Thus, if per flow marking is applied, the packet loss probability, $q(l)$, for a flow in group $l \in \mathcal{L}^d$ becomes

$$q(l) = p^d(pr(l)). \tag{7}$$

On the other hand, if per packet marking is applied, then

$$q(l) = \sum_{j=1}^{I} p^d(j) \frac{\min\left[\nu(l), t(l, j-1)\right] - \min\left[\nu(l), t(l, j)\right]}{\nu(l)}. \tag{8}$$

For each TCP flow these packet loss probabilities can be used to determine iteratively the packet arrival rate ν from equation (2). Then these rates are again aggregated as in (4) and (5), and the aggregate rates are used to solve the stationary distribution of the resulting two-dimensional Markov process. By continuing this iteration, the traffic rates of TCP flows converge to some equilibrium values, which reflect the network state, i.e. the number of flows $n(l)$ in different classes l.

3 Numerical Results and Conclusions

We study the combined effect of the three degrees of freedom introduced in the text: marking, discarding thresholds and weighted capacity.

We have the following scenario in terms of the free parameters: $\mu = 1$, $K^1 = 13$, $K^2 = 39$ and $I = 3$. In addition we consider two flow groups, non-TCP flows in group 1 with $\phi(1) = 0.08$ and $\nu(1) \in \{0.039, 0.079, 0.16\}$, and TCP flows in group 2 with $\phi(2) = 0.04$ and $RTT(2) = 1000/\mu$. The three values of $\nu(1)$ are chosen so that, under the per flow marking scheme, the non-TCP flows have priorities $pr(1) = 3$, $pr(1) = 2$, and $pr(1) = 1$, respectively.

Each set of pictures depicted in figure 2 show the ratio $\frac{\theta(1)}{\theta(2)} = \frac{\nu(1)(1-q(1))}{\nu(2)(1-q(2))}$ between throughputs of flows as function of total number of flows, under the condition $n(1)/n(2) = 1/2$. The trajectories are solid, gray, and dashed for $\nu(1) = 0.039$, $\nu(1) = 0.079$, and $\nu(1) = 0.16$, respectively.

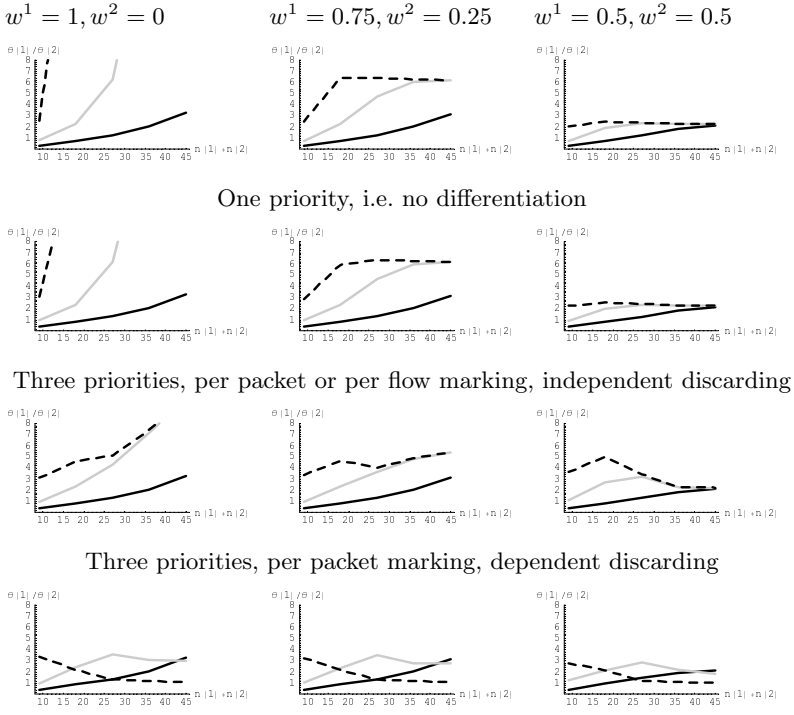

Fig. 2. Effect of marking and discarding when the minimum weights of the rt buffer and nrt buffer change. 66% are TCP flows and 33% non-TCP flows.

The lowest pair in figure 2 shows the effect of per flow marking and dependent discarding. Marking all packets of the flow to the same priority level encourages the TCP mechanism to optimize the sending rate according to the network state. Under congestion, the TCP flows attain a higher priority level by dropping their

sending rate. This also encourages the non-TCP traffic to adjust the sending rate accordingly. In all other cases, it is always optimal for the non-TCP flows to send as much as possible, even if packets are then marked to the lowest priority level. The use of per flow marking and dependent thresholds thus gives a powerful incentive for flows to be TCP friendly [11].

The use of dependent discarding controls the throughput of non-responsive flows better than independent discarding. With dependent thresholds, when the nrt buffer is congested packets in the rt buffer are also discarded to alleviate the congestion.

The effect of giving some minimum weight to the nrt buffer protects the TCP traffic from bandwidth exhaustion by the non-TCP flows. However, there is not a clear one to one relationship between the ratio w^1/w^2 of scheduler weights and ratio $\phi(1)/\phi(2)$ of flow group weights.

Further research has to be done in elaborating the TCP congestion control model to include slow start. Furthermore, to properly assess the mechanisms we need to extend the model to networks with more than one bottleneck link.

Acknowledgments. Eeva Nyberg's research is supported by the Academy of Finland and in part by a grant from the Nokia Foundation. The authors would like to thank Jorma Virtamo and Eemeli Kuumola for their cooperation.

References

1. Blake S., Black D., Carlson M., Davies E., Wang Z., and Weiss W., *An Architecture for Differentiated Service*, Dec. 1998, RFC 2475.
2. Heinanen J., Baker F., Weiss W., and Wroclawski J., *Assured Forwarding PHB Group*, June 1999, RFC 2597.
3. Kelly F., "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun*, vol. 8, pp. 33–37, 1997.
4. Massoulié L. and Roberts J., "Bandwidth sharing: Objectives and algorithms," in *Proceedings of IEEE INFOCOM*, 1999, pp. 1395–1403.
5. Nyberg E., Aalto S., and Virtamo J., "Relating flow level requirements to DiffServ packet level mechanisms," Tech. Rep. TD(01)04, COST279, Oct. 2001.
6. Kilkki K., "Simple Integrated Media Access," available at http://www-nrc.nokia.com/sima, 1997.
7. Roberts J., "Traffic theory and the Internet," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 94–99, Jan. 2001.
8. Kelly F., "Mathematical modelling of the Internet," in *Proc. of Fourth International Congress on Industrial and Applied Mathematics*, 1999, pp. 105–116.
9. Nyberg E., Aalto S., and Susitaival R., "A simulation study on the relation of DiffServ packet level mechanisms and flow level QoS requirements," in *Intl. Seminar, Telecommunication Neworks and Teletraffic Theory*, St. Petersburg, Russia, 2002.
10. Laine J., Saaristo S., Lemponen J., and Harju J., "Implementation and measurements of simple integrated media access (SIMA) network nodes," in *Proceedings for IEEE ICC 2000*, June 2000, pp. 796–800.
11. Floyd S. and Fall K., "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, Aug. 1999.