Intra-Customer Admission Control for TCP flows in DiffServ Assured Forwarding

Albert Banchs, Sandra Tartarelli and Arnaud Descamps
Network Laboratories Heidelberg, NEC Europe Ltd.

Abstract—**With DiffServ Assured Forwarding, a customer contracts a certain bandwidth, which is shared among the flows sent by him. If the customer sends too many flows, the realized throughput for each will be very low, and the corresponding transfer is very likely to be interrupted by its user due to its prolonged completion time. Since the work done in transferring an incomplete document is generally wasted, interrupted transfers constitute a significant source of inefficiency in the utilization of the contracted bandwidth. In order to avoid this inefficiency, we propose the use of admission control to limit the number of flows of a customer to ensure that each receives a minimal acceptable throughput.**

I. Introduction

The Differentiated Service (DiffServ) architecture [1] has been proposed as a scalable way of providing Quality of Service (QoS) in the Internet. Scalability is achieved by moving complicated functionality toward the edge and leaving the core with very simple functionality. With DiffServ, packets are marked at the ingress of the network with a DiffServ codepoint (DSCP) and at the core they are given a forwarding treatment according to their DSCP. Each DSCP corresponds to a Per-Hop Behavior (PHB).

The IETF's DiffServ Working Group has defined two PHB groups: the Expedited Forwarding (EF) PHB [2] and the Assured Forwarding (AF) PHB [3]. This paper focuses on the latter. Typically, an ISP would use the AF PHB to provide a service where the packets of a customer are forwarded with a very high probability as long as the aggregate traffic from the customer does not exceed the target rate he has contracted, the Committed Information Rate (CIR).

Even though the AF PHB has become a standard, there are still some open issues that need to be understood. The most important question relates to the kind of end-to-end services that can be provided [4], [5]. There is the concern that AF should show some measure of predictability for paying customers. If charging is based on the CIR, then in a predictable system a customer would expect to receive a throughput at least equal to the CIR.

In order to achieve this target for one-to-one services (like IP-VPN), in [6] we proposed some configuration rules for DiffServ AF. Exhaustively evaluated via simulation [7], these rules proved to work well for UDP traffic, but we also showed that there was room for improvement in the case of TCP. In [8] we proposed the Random Early Marking (REM) scheme to improve the performance of AF customers sending TCP flows, by early notifying TCP sources of congestion. In this paper we address the issue of improving the performance of TCP flows by performing intra-customer admission control. In line with [6] and [8], in this paper we focus on one-to-one services (i.e. all flows of a customer traverse the same path) and TCP-only customers (all flows sent by a customer are TCP). We believe that the solutions proposed in [6], [8] and the one we present here are complementary and can be combined to optimize performance.

TCP flows correspond to the transfer of some document (Web page, file, MP3 track,...) and adapt their rate to the available bandwidth. With DiffServ AF, the TCP flows that belong to the same customer share the capacity available for this customer. Let's assume that this capacity is equal to the CIR contracted by the customer. Then, if the traffic demand[1] of the customer is less than the CIR (*underload*), all transfers can be completed and quality of service is generally good. On the other hand, if demand exceeds the CIR (*overload*), congestion necessarily ensues.

If users[2] did not react on overload, the number of flows in progress of the customer would increase indefinitely as long as the overload lasted, and the per-flow throughput would tend to zero. In practice, however, as transfer times grow longer some users become impatient and interrupt their transmissions. Similarly, as bandwidth tends to zero and packet loss increases, higher layer protocols consider the connection to be broken and abandon. In either case, all the work done in partially transferring a document is wasted, and the useful throughput or goodput experienced by the customer is lower than his CIR.

In this paper we propose to perform admission control on the TCP flows of a customer to ensure that any accepted flow of the customer receives a minimal acceptable throughput such that it can be completed. We argue that intra-customer admission control is essential to avoid wasting the customer's CIR on incomplete transfers.

The need for admission control for TCP flows had already been identified by Massoulié and Roberts [9]. The main difference between their approach and ours is in concept: while in [9] admission control is applied on a link basis in order to improve the efficiency of the link's utilization, in our approach admission control is applied on a customer basis in order to improve the efficiency of the customer's CIR. In addition, there are other important differences between [9] and our approach. In [9], per-flow admission control is performed in all links, which may result in scalability problems when applied to high speed core

[1] By traffic demand, we mean the flow arrival rate multiplied by the expected volume of data to be transferred.

[2] It is important to note the difference between a user and a customer. We refer to customer as the entity that contracts the DiffServ AF service. By user we mean the person originating the file transfer. A typical customer could be a web server that contracts the AF service to provide its clients with a high quality of service; the users of this customer are the clients that connect to the web server.

0-7803-7632-3/02/$17.00 ©2002 IEEE

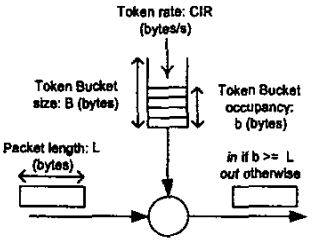

Fig. 1. Token Bucket algorithm.

routers. In contrast, in our approach admission control is only performed in lower speed edge routers. In [9] packets of rejected flows are dropped. In our approach, they are just marked with a lower level of drop precedence (they will be dropped only if they traverse a congested link).

The rest of the paper is structured as follows. In Section II we present our approach to perform admission control for TCP flows in AF. In the explanation of the approach, the admissibility criterion for new flows is left as an open issue. In Section III we study this issue analytically, and in Section IV we study it via simulation. Finally, Section V closes the paper with the conclusions.

II. Intra-Customer Admission Control

In this paper we assume that the Assured Forwarding PHB is implemented with the WRED algorithm [10] with two levels of drop precedence (*in* and *out*). The admission control scheme we propose, however, could be easily extended to three levels of drop precedence (*green*, *yellow* and *red*).

WRED works as follows. Packets entering the network are marked *in profile* or *out of profile* at the ingress according to the Token Bucket algorithm of Figure 1. In this algorithm, a token bucket of depth B is filled at the rate specified by the CIR, where B and CIR are part of the contract with the customer. Then, when a packet from the customer is received, it is marked *in* if there are enough bytes in the bucket for this packet, and it is marked *out* otherwise. In case of *in* marking, a number of bytes equal to the packet length is substracted from the bucket.

At core nodes, all packets, *in* and *out*, are put into the same buffer. This buffer is managed in such a way that in case of congestion *out* packets are dropped first. In this way, the WRED algorithm guarantees that, as long as the network is configured such that *in* packets alone do not cause congestion, *in* packets are never dropped. In [6] we provide configuration guidelines to achieve this behavior.

Based on the above explanation of the Token Bucket algorithm, we propose to perform admission control in the following way:

- Packets of accepted flows are marked *in* if there are enough bytes in the bucket and are marked *out* otherwise.
- Packets of rejected flows are all marked *out* and they do not consume bytes of the token bucket.

Admissibility of new flows should be performed according to the following criteria:

- The rate of rejected flows or blocking probability in *underload* should be negligible.
- A sufficiently high *in* marking rate for admitted flows should be mantained in *overload*.

With the above, we ensure that the customer's CIR is only spent on marking *in* packets of accepted flows. Since accepted flows will receive a satisfactory throughput, they will not be aborted, resulting this in an efficient use of the customer's CIR.

On the other hand, packets of rejected flows will be marked *out* and will most probably be dropped when traversing a congested link. A rejected flow traversing a congested link will most likely be aborted due to too low bandwidth; however, since this flow does not consume bytes from the customer's token bucket, this does not harm the efficient utilization of the CIR.

Note that in the case when some flows do not traverse congested links, *out* packets of those flows will not be dropped, and the customer will experience a throughput larger than the contracted CIR.

III. Analysis

In the admission control scheme described in the previous section, the issue of how to determine the admissibility of a new flow according to the desired criteria is still pending. In this section we investigate this issue by means of the following analytical model.

Consider a customer that has contracted a CIR equal to C (bits/sec). TCP flows of this customer arrive according to a Poisson process of intensity λ (flows/sec) and the sizes of the transferred documents constitute an i.i.d. sequence of exponentially distributed random variables with mean $1/\mu$ (bits/flow). Let ρ denote the load offered by this customer, $\rho = \lambda/C\mu$.

In addition, let's assume that all the flows of the customer traverse congested links, such that the total capacity experienced by the customer is equal to his CIR. Let's also assume that this capacity is shared equally among the flows in progress of the customer.

The user of a flow may grow impatient and abort his flow before it is completed. We model the sequence of "patience durations" as an i.i.d. sequence of exponential random variables with mean $1/\nu$ (sec). We assume that admission control is implemented by rejecting additional flows when there are already N flows in progress from the customer.

In the above system, the number of (accepted) flows of the customer in progress is a Markov process with a stationary probability distribution defined by

$$\Pi_n = \frac{\left(\frac{\lambda}{\nu}\right)^n \frac{1}{\prod_{k=1}^{n}(k+\nu/\lambda)}}{\sum_{n=0}^{N}\left(\frac{\lambda}{\nu}\right)^n \frac{1}{\prod_{k=1}^{n}(k+\nu/\lambda)}} \qquad (1)$$

From the above, the throughput can be derived as $1 - \Pi_0$, and the blocking probability as Π_N. A formula for the goodput is given in [11][3].

Figure 2 illustrates the resulting goodput for the case of a customer with a CIR of 500 Kbps, a mean document size ($1/\mu$) of 10 Kbytes and a mean user patience duration ($1/\nu$) of 10 sec. Results are given for an admission control threshold N = 10, 20, 30 and no admission control.

[3][11] uses an analog model to the one presented here to analyze the goodput in a link when admission control is preformed on a link basis.

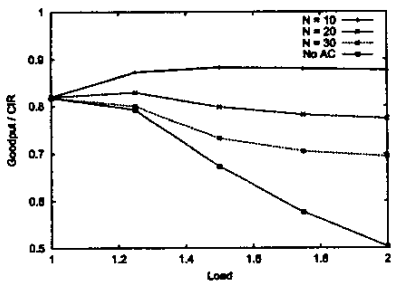

Fig. 2. Goodput normalized to the customer's CIR.

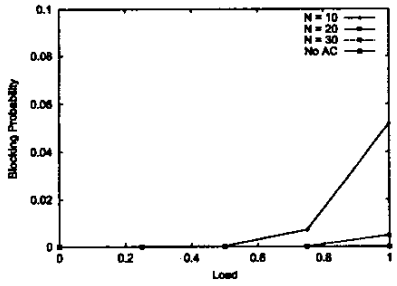

Fig. 3. Blocking Probability in underload.

It can be observed that, even though in all cases the throughput almost equals the CIR (the throughput is not shown for space reasons), the goodputs can be significantly lower due to user impatience (in the most extreme case, with an offered load of 2 and no admission control, the goodput is only one half of the CIR). However, with a proper admission control ($N < 30$) the goodput keeps high independent of the offered load.

Figure 3 illustrates the blocking probabilities with the above admission thresholds in underload. It can be observed that if the admission threshold is too low ($N = 10$), the blocking probability is undesirably high. From the above, we conclude that a reasonable admission threshold is $10 < N < 30$.

The above model is clearly very simple and imprecise. However, we believe that it does illustrate the following points:

- Useful work accomplished by a customer can be much lower than the CIR he has contracted.
- Applying intra-customer admission control is an effective means to maintain useful throughput.
- There is a tradeoff between useful throughput and blocking probability; the admission threshold should be chosen according to this tradeoff.

IV. Simulations

In this section we further investigate the issue of how to determine the admissibility of a new flow by means of packet level simulations[4]. For this purpose, we study the performance of two different algorithms to determine admissibility and we discuss on the simulation results obtained.

[4]Simulations were run by using ns-2 [12].

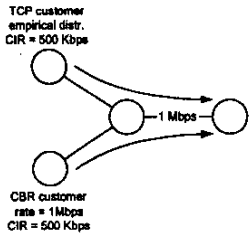

Fig. 4. Simulation scenario.

A. *Number of flows* scheme

The first scheme we studied is the one that we analyzed in the previous section, i.e. limiting the number of accepted flows to N.

In order to evaluate the performance of this scheme, we ran the following simulation (see Figure 4). A 1 Mbps link was shared by two customers, both with a contracted CIR of 500 Kbps. The first customer sent background CBR traffic at a rate of 1 Mbps. The second was the target customer, and sent TCP transfers with arrivals according to a Poisson process and sizes according to an empirical distribution[5]. The users of these TCP transfers had an infinite patience[6]. We considered the two cases for which the target customer sent respectively at a total aggregate rate of 90% and 140% of the CIR, in order to observe the behavior both in the *underload* and in the *overload* cases. As measures of performance we evaluated the distribution of the per-flow throughput and the blocking probability; for both metrics, average values and confidence intervals were measured. Based on the analytical results of the previous section, we studied the performance with values for the admissibility threshold in the range $N \in (10, 30)$.

Figures 5 and 6 show how the CIR of the target customer is spent on the different flows depending on the flows' experienced throughputs. The interpretation of these figures is as follows. If the bar corresponding to $1 < S < 10$ is of 0.1, this means that 10% of the CIR is consumed by flows that experience a flow throughput between 1 Kbps and 10 Kbps (where flow throughput = flow size / completion time of the flow).

As expected, the throughput distribution in underload (Figure 5) is always good: independent of N, almost all flows experience a throughput above 10 Kbps. This is because in underload all flows are able to finish within a relatively short completion time, even when admission control is not applied. In contrast, in overload (Figure 6) the throughput distribution highly depends on N. With N equal to 10 and 20, most of the customer's CIR is spent on flows that experience an acceptable throughput. Since these flows are not likely to be interrupted, we conclude that these values of N guarantee an efficient utilization of the CIR. $N = 30$ provides a worse throughput distribution, with an almost 10% of the CIR spent on flows that experience a throughput below 10 Kbps.

[5]The empirical distribution has been derived from the Internet traffic measurements given in [13].

[6]Note that, in case admission control is such that all accepted flows receive a minimal acceptable throughput so that they are not interrupted, then the user impatience only affects the rejected flows and has no impact into our measures.

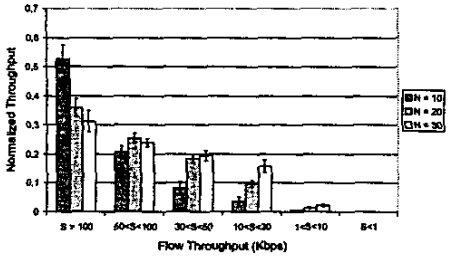

Fig. 5. Throughput distribution with the *number of flows* scheme (load 90%).

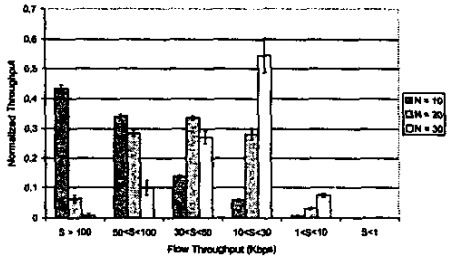

Fig. 6. Throughput distribution with the *number of flows* scheme (load 140%).

Figure 7 shows the blocking probability as a function of N and compares it with the theoretical ones according to the analytical model of the previous section. It can be seen that simulation and analytical results are fairly close, which confirms the analytical model used in the previous section[7]. With a 140% load, we observe that the blocking probability is fairly independent of the admission threshold N. In contrast, in underload there is a considerable dependency with N, with a non-negligible blocking probability for $N < 20$.

From the above simulation results, an admissibility threshold of $N = 20$ appears as a reasonable choice. With this threshold, all desired criteria stated in Section II are reasonably well met:

[7]Note that simulation results show slightly higher blocking probabilities than the analytical results. This is due to the heavy-tailed nature of the empirical distribution [13], which causes the arriving load to be more bursty than with the exponential distribution considered in our analytical model.

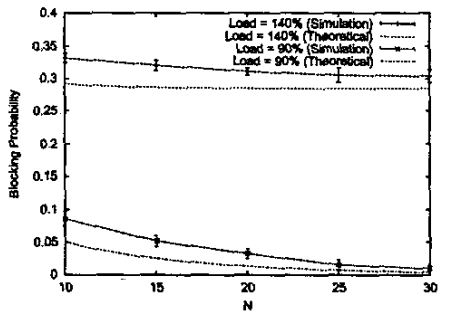

Fig. 7. Blocking Probability with the *number of flows* scheme.

- In overload almost all flows experience an acceptable throughput. Specifically, at a load of 140%, less than 5% of the CIR is spent on the flows that experience a throughput lower than 10 Kbps.
- In underload the blocking probability is small. Specifically at 90% load the blocking probability is lower than 5%.

However, even though the scheme studied in this subsection meets all the desired targets fairly well, it also suffers from a fundamental problem: if the accepted N flows send at a total aggregate rate below the CIR (e.g. the sending rate of a telnet session can be very low), then additional flows will be rejected even when they could be satisfactorily admitted. Therefore, a scheme based on the number of accepted flows cannot possibly guarantee an efficient utilization of the CIR, whereas a scheme based on the observed level of congestion must be used. In the following subsection we study the performance of such a scheme. The results presented in this subsection will be used as a reference to assess the performance results of the congestion-based scheme.

B. *Out marking ratio* scheme

The congestion-based scheme that we studied is based on the ratio of packets marked *out*, r_{out}. With this scheme, we monitor the percentage of packets of accepted flows that are marked *out*. Then, if this value exceeds a given threshold R, an incoming new flow is rejected, otherwise it is accepted.

For the estimation of the *out* marking ratio r_{out}, we measure it in fixed time intervals of δ seconds and apply exponential smoothing with parameter α. Specifically, let N_{in}^n and N_{out}^n be the number of packets of accepted flows marked *in* and *out*, respectively, in the interval $((n-1)\delta, n\delta]$, and let r_{out}^n be the *out* marking ratio estimate derived at time $n\delta$. Then, the *out* marking ratio r_{out} is updated every δ seconds according to:

$$r_{out}^n = (1-\alpha)\cdot r_{out}^{n-1} + \alpha \cdot \frac{N_{out}^n}{N_{out}^n + N_{in}^n} \qquad (2)$$

The values of δ and α are not highly critical to the accuracy of the method. Following a series of initial experiments, we set $\delta = 1$ and $\alpha = 0.1$.

In order to evaluate the performance of this scheme, we ran the same simulation as in the previous subsection. Figures 8 and 9 show the resulting throughput distribution and Figure 10 the blocking probability. From these simulation results, an admissibility threshold of $R = 15\%$ appears as a reasonable choice for the present configuration, since it provides a good tradeoff between the throughput distribution and the blocking probability. If we compare the simulation results obtained with this threshold with the ones obtained with the *number of flows* scheme in the previous subsection, we can observe that the performance of both schemes is very similar, both in terms of throughput distribution and blocking probability.

C. Discussion

In this section we have studied via simulation two different schemes to determine admissibility.

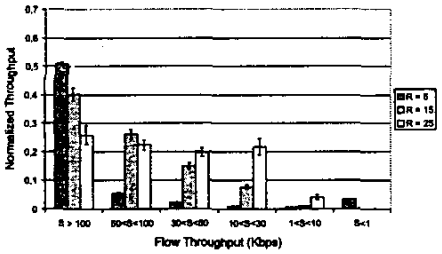

Fig. 8. Throughput distribution with the *out marking ratio* scheme (load 90%).

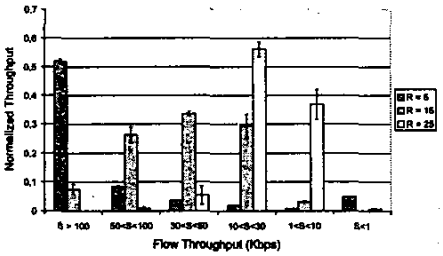

Fig. 9. Throughput distribution with the *out marking ratio* scheme (load 140%).

The first scheme prevents congestion by limiting the number of accepted flows. As a consequence, congestion never occurs and the throughput distribution is good. Unfortunately, this scheme is not applicable since it may reject flows also when rejection is not needed, i.e. in absence of congestion. However, the results from this scheme do show the potential of performing intra-customer admission control.

The second scheme falls into the family of congestion-based schemes, i.e. schemes that react by rejecting new flows upon detecting congestion. Since these schemes only act once congestion is already there, the unavoidable presence of congestion may lead to a worse distribution of the flow throughput than the first scheme. As a consequence, the simulation results obtained with the first scheme give a benchmark against which to assess the performance results of congestion-based schemes.

The specific congestion-based scheme that we have studied in this paper is based on detecting congestion by means of the *out* marking ratio. Since the results obtained with this scheme are very close to the reference results of the *number of flows* scheme, we conclude that the performance provided by the *out marking ratio* scheme is satisfactory.

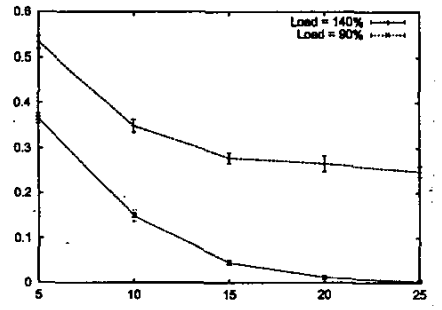

Fig. 10. Blocking Probability with the *out marking ratio* scheme.

V. Conclusions

The propensity of flows to be interrupted as their response time is prolonged due to congestion is a significant source of inefficiency in the utilization of the CIR contracted by a customer. The simple analytical model used in this paper illustrates that the proportion of CIR wasted on such interrupted transfers can be considerably high.

In order to avoid this inefficiency caused by interrupted flows, in this paper we propose the use of intra-customer admission control. Explicit admission control using signaling for each TCP flow would introduce an unacceptably high overhead. Instead, we propose to perform admission control implicitly, by simply marking *out* all the packets of a rejected flow.

With the proposed scheme, the customer's CIR is only spent on the flows that have been accepted by the admission control. Simulation results show that the use of a proper admission control provides these accepted flows with an acceptable throughput. Since flows with an acceptable throughput are not likely to be interrupted, we conclude that the proposed scheme is able to guarantee an efficient utilization of the customer's CIR.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, December 1998.

[2] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," RFC 2598, June 1999.

[3] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, June 1999.

[4] N. Seddigh, B. Nandy, and J. Heinanen, "An Assured Rate Per-Domain Behavior for Differentiated Services," Internet draft, February 2001.

[5] M. Brunner, A. Banchs, S. Tartarelli, and H. Pan, "A one-to-any Assured Rate Per-Domain Behavior for Differentiated Services," Internet draft, April 2001.

[6] S. Sato, K. Kobayashi, H. Pan, S. Tartarelli, and A. Banchs, "Configuration Rule and Performance Evaluation of Diffserv Parameters," in *Proceedings of 17th International Teletraffic Congress (ITC17)*, Salvador da Bahia, Brazil, December 2001.

[7] A. Banchs, S. Tartarelli, F. Orlandi, S. Sato, K. Kobayashi, and H. Pan, "Configuration of Diffserv routers for high-speed links," in *Proceedings of 2002 Workshop on High Performance Switching and Routing (HPSR 2002)*, Kobe, Japan, May 2002.

[8] S. Tartarelli and A. Banchs, "Random Early Marking: Improving the Peformance of TCP in Diffserv Assured Forwarding," accepted to IEEE ICC 2002.

[9] L. Massoulié and J. W. Roberts, "Bandwidth Sharing and Admission Control for Elastic Traffic," in *Proceedings of the ITC Specialists Seminar*, Yokohama, Japan, October 1998.

[10] CISCO SYSTEMS, "Congestion Avoidance Overview," `http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt3/qcfconav.htm`.

[11] L. Massoulié and J. W. Roberts, "Arguments in Favour of Admission Control for TCP Flows," in *Proceedings of 16th International Teletraffic Congress (ITC16)*, Edinburgh, Scotland, June 1999.

[12] UCB/LBNL/VINT, "Network Simulator (ns), version 2," `http://www.isi.edu/nsnam/ns/`.

[13] A. Feldmann, j. Rexfor, and R. Caceres, "Efficient policies for carrying Web traffic over flow-switched networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 673–685, December 1998.