Two-Differentiated Marking Strategies for TCP Flows in a Differentiated Services Network

Sung-Hyuck Lee, Seung-Joon Seok, Seung-Jin Lee,
Sung-Kwan Youm, and Chul-Hee Kang

Department of Electronics Engineering, Korea University
5-Ka, Anam-dong Sungbuk-ku,136-701 Seoul, Korea
{starsu, ssj, linuz, skyoum}@widecomm.korea.ac.kr
chkang@korea.ac.kr

Abstract. The saw-tooth like behaviors of TCP impact Assured Forwarding Service flows in a differentiated services network. Therefore, we argue the use of TCP-friendly building blocks(or modules) and fairness modules in the Differentiated Services architecture regarding this issue, and propose *Two Markers System(TMS)* that is able to properly mark packets and fairly share the bandwidth to each flow for their targeted sending rates. TMS has two marking modules that are placed on the source and the edge of a differentiated services network. For sources of the network, *the virtual source making modules* play important roles of reducing TCP impacts in the assured services and suitable marking packets. Next, in the edge of the network, *the edge embedded marking module* conducts new fairness policy based on the marking rate of flows from sources, so called *"marking rate-based fairness"*. Finally, we present simulation results to illustrate the effectiveness of TMS scheme over several parameters. That is, Two Markers System reduces TCP impacts over assured service and fairly shares the bottleneck link bandwidth of a network.

1 Introduction

The commodity Internet is based on the best-effort service model. In this model the service provider allocates bandwidth among all of the instantaneous customers as best it can, that is, all user packets compete equally for network resources. The service provider also attempts to serve all of them without making any explicit commitment as to rate or any other service quality. The best-effort model has been successful till now because the majority of the traffic on the Internet is TCP-based. The TCP end-to-end congestion control mechanism forces traffic sources to back off whenever congestion is detected in the network[8]. However, such dependence on the end systems' interaction is increasingly becoming unrealistic. Given the current best-effort model with FIFO queuing inside the network, it is relatively easy for non-adaptive sources to gain greater shares of network bandwidth and thereby starve other, well-behaved, TCP sources. A greedy source, for example, may simply continue to send at the same rate when faced with congestion while other TCP sources back off.

M. Smirnov et al. (Eds.): QofISs 2001, LNCS 2156, pp. 207-221, 2001

The diffserv approach is based on a set of simple mechanisms that treat packets differently according to the marking of the DS field in the IP header. Before entering in a DS domain, the field is marked with a certain value(or codepoint) that determines the treatment that should be supplied to the packet inside the domain. However, because of the limited amount of bits available for use in the DS field, the IETF's Diffserv Working Group has defined a small set of building blocks, called per-hop behaviors(PHBs) which are used by routers to deliver a number of services. Among the initial PHBs being standardized are the Expedited Forwarding(EF) and the Assured Forwarding(AF) PHBs. The EF PHB specifies a forwarding behavior in which packets see a very small amount of loss and a very low queuing delay. In order to ensure every packet marked with EF receives this service, EF requires every router to allocate enough forwarding resources so that the rate of incoming EF packets is always less than or equal to the rate which the router can forward them. The AF PHB group, on the other hand, specifies a forwarding behavior in which packets see a very small amount of loss, and consists of four, independently forwarded classes which have two or three drop preference levels. The idea behind AF is to preferentially drop best-effort packets and packets which are outside of their contract when congestion occurs.

In this paper, we consider a form of a better-than-best-effort service called the "Assured Service". The Assured Service follows expected capacity profiles which are statistically provisioned. Packets are treated preferentially according to the dropping probability applied to the best-effort queue. The assurance of service comes from the expectation that the traffic is unlikely to be dropped as long as it stays within the negotiated capacity profile. The building blocks of this service include a traffic marker at the edge of the domain, and a differentiated dropping algorithm in the core of the network. A packet of a flow is marked IN(in profile) if the temporal sending rate of the arrival time of the packet is within the contract profile of the flow. Otherwise, the packets are marked OUT(out-of-profile). The temporal sending rate of a flow is measured using TSM(Time Sliding Window) or token bucket control module. A differentiated dropping algorithm such as RIO(Random Early Detection with IN/OUT) is provided in the core routers of the network. In particular, the OUT packets are preferentially dropped upon evidence of congestion at the bottleneck before the IN packets. After dropping all incoming OUT packets, IN packets are discarded. With this dropping policy, the RIO network gives preference to IN packets and provides different levels of service to users based on their service contracts.

In [12], authors presented that the use of a simple token bucket marker for the above assured service results in TCP realizing the minimum assured rate. The authors attributed the cause of such behavior to TCP's complex response primarily to packet losses. TCP reacts to congestion by halving the congestion window(cwnd) and increases the window additively when packets are delivered successfully. Exponential decrease(halving the congestion window) is required to avoid congestion collapse and TCP treats a packet drop as an indication congestion[8]. However, in the differv network these additive-increase and multiplicative-decrease make it hard to protect the reservation rate. When TCP reacts to an OUT packet drop by halving its congestion window and increases additively, it may not reach its reservation rate.