



ELSEVIER

Computer Networks 34 (2000) 905–914

COMPUTER
NETWORKS

www.elsevier.com/locate/comnet

End-to-end performance analysis with traffic aggregation[☆]

Tiziana Ferrari

Italian National Institute for Nuclear Physics (INFN-CNAF), v. le Bertini Pichat 6/2, I-40127 Bologna, Italy

Abstract

The provisioning of end-to-end services in the presence of traffic aggregation is evaluated in the framework of the Differentiated Services (diffserv) QoS architecture. The effect of stream multiplexing on delay and jitter-sensitive traffic is analyzed through an experimental approach. Aggregation is evaluated in three different scenarios: with different aggregate traffic loads, with a variable number of flows multiplexed in the same class, and with different packet sizes. Two different scheduling algorithms are considered: Priority Queuing (PQ) and Weighted Fair Queuing (WFQ). © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Quality of service; End-to-end performance; Differentiated services; Traffic aggregation; Performance measurement

1. Introduction

The *Differentiated Services* (diffserv) QoS architecture, specified in RFC 2475, is based on the aggregation of streams into classes and on the provision of QoS to the class instead of the single flow [1,2,4,5]. The per-class packet treatment is defined by the *Per-Hop Behavior* (PHB), which describes the packet handling mechanism a given class is subject to [3]. Packets are classified by access routers or hosts through traffic filters: the class to which a packet is bound is identified by a specific code in the *diffserv field* called DSCP (DiffServ Code Point) as specified in RFC 2474. Classification and marking are performed at the ingress of a given domain. Only edge routers have access to the

policy for the binding between packets and classes. Core diffserv nodes only manage classes; i.e., no per-flow information is deployed in the core. This is to move the network complexity from the core to the edge.

In this paper, we focus on one aspect of diffserv: the provision of end-to-end services in the presence of traffic aggregation. While the mixing of different streams into one class is an inherent property of diffserv and is fundamental for improved scalability of the architecture, the interference between packets of multiple flows – that are treated in an undifferentiated way within a class – can have an influence on the end-to-end performance of the single stream.

While aggregation is of relevance to diffserv, this issue does not arise in QoS architectures, such as *ATM* and *intserv* [7–9], that provide finely grained QoS by supporting quality of service to the flow: both the stream and the reservation profile are advertised through a signaling protocol so that streams are treated independently. The problem of per-flow performance under aggregation is a topic

[☆] The results reported in this article are part of work being carried out by the joint DANTE/TERENA task force, TANTANT [25,27], in the framework of the EC-funded QUANTUM [26] project.

E-mail address: Tiziana.Ferrari@cnaif.infn.it (T. Ferrari).

that still needs more research for better understanding. In this work, we address the problem through an experimental approach by running tests over a diffserv test network. Given a reference stream to which measurement applies, the end-to-end performance of one stream in the class is evaluated in different scenarios: for different aggregation degrees, different loads, and different packet sizes.

In order to keep the end-to-end behavior of a flow in accordance with the contract, policing, shaping or other forms of traffic conditioning may be adopted. In this work, we call *stream isolation* the capability of a network to preserve the original stream profile when it is transmitted over one or more diffserv domains in an aggregated fashion. The maximum degree of traffic isolation is similar to that achieved when a dedicated wire carries each stream. In this work, we estimate the traffic isolation supplied in different traffic and network scenarios in order to evaluate the feasibility of the QoS architecture being tested and to identify the conditions for maximum isolation.

2. Traffic aggregation

The behavior of a diffserv network is related to the type of packet treatment in use, in particular to the scheduling algorithm adopted. Given the large variety of schedulers, we restrict our study to two well-known solutions: *Weighted Fair Queuing* (WFQ) [15–21] and *Priority Queuing* (PQ) [22, 23, 29]. We develop the study of PQ in detail and compare the performance of the two.

According to the WFQ algorithm, packets are scheduled in increasing order of *forwarding time*, which is an attribute computed for each packet upon arrival. It is a function of both the packet size and the weight of the queue the datagram belongs to. On the other hand, with PQ [8, 9], every time the ongoing transmission is over, the priority is checked: the priority queue is selected for transmission if it contains one or more packets, while the transmission control is released only when/if the priority queue is empty. While WFQ provides a fair distribution of link capacity among queues, PQ is starvation-prone.

Given the difference between the two algorithms, their performance is compared under different traffic scenarios to identify the most suitable approach for the provision of delay, jitter and packet loss guarantees. The goal is to identify configuration guidelines for the support of the Expedited Forwarding (EF) [6] PHB for the support of services. End-to-end performance measurement is applied to a single stream in a class, which we call a *reference stream* in this paper. We assume that the class the reference stream belongs to applies for the EF PHB, since the goal of configuration tuning is the limitation of one-way delay, jitter and packet loss. Nevertheless, the results of this work can be applied to any traffic class that is subject to PQ or WFQ.

Given a diffserv node, aggregation occurs when bundles of streams, coming from two or more different input interfaces, belong to the same PHB or PHB group and are sent to the same output interface. It can happen that in the long or short term – depending on the packet rate of the streams – arrival times of packets belonging to different bundles are close to each other, so that *packet collision* occurs. Collision can produce packet accumulation, i.e., instantaneously non-empty queues as illustrated in Fig. 1. As stated in [6, 10], given a diffserv node, packet clustering in the queue can be avoided if the minimum *departure rate*, i.e., the output rate of a given traffic class, is strictly greater than the maximum *arrival rate*, i.e., the maximum instantaneous input rate, when

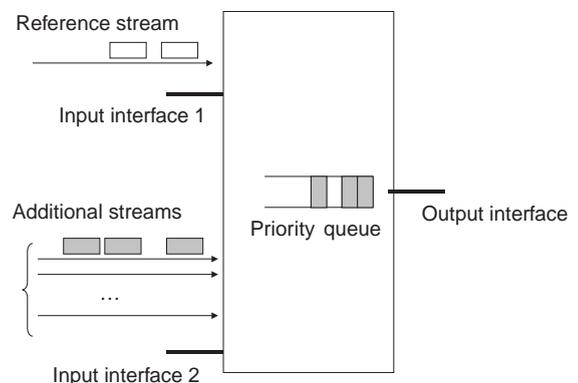


Fig. 1. Aggregation of streams and packet clustering in the output queue.

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات