Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 29 (2006) 3679–3690
computer communications
www.elsevier.com/locate/comcom

A robust packet scheduling algorithm for proportional delay differentiation services

Jianbin Wei [a,*], Cheng-Zhong Xu [a], Xiaobo Zhou [b], Qing Li [a]

[a] Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA
[b] Department of Computer Science, University of Colorado at Colorado Springs, Colorado Springs, CO 80918, USA

Received 20 November 2004; received in revised form 15 June 2006; accepted 20 June 2006
Available online 12 July 2006

Abstract

Proportional delay differentiation (PDD) model is an important approach to relative differentiated services provisioning on the Internet. It aims to maintain pre-specified packet queueing-delay ratios between different classes of traffic at each hop. Existing PDD packet scheduling algorithms are able to achieve the goal in long time-scales when the system is highly utilized. This paper presents a new PDD scheduling algorithm, called *Little's average delay* (LAD), based on a *proof* of Little's Law. It monitors the arrival rate of the packets in each traffic class and the cumulative delays of the packets and schedules the packet according to their transient queueing properties in order to achieve the desired class delay ratios in both short and long time-scales. Simulation results show that LAD is able to provide predictable and controllable services in various system conditions and that such services, whenever feasible, can be guaranteed, independent of the distributions of packet arrivals and sizes. In comparison with other PDD scheduling algorithms, LAD can provide the same level of service quality in long time-scales and more accurate and robust control over the delay ratio in short time-scales. In particular, LAD outperforms its main competitors significantly when the desired delay ratio is large.

Keywords: Quality of service; Packet scheduling; Proportional delay differentiation; Little's law

1. Introduction

The past decade has seen an increasing demand for provisioning of different levels of quality of service (QoS) on the Internet to support different types of network applications and different user requirements. To meet this demand, two service architectures are proposed: Integrated Services (IntServ) [4] and Differentiated Services (DiffServ) [2]. IntServ requires to reserve routing resources along the service delivery paths using a protocol like Resource Reservation Protocol (RSVP) for QoS guarantee. Since all the routers need to maintain per-flow state information, this requirement hinders the IntServ architecture from widespread deployment. In contrast, DiffServ aims to provide differentiated services among classes of aggregated traffic flows, instead of offering absolute QoS measures to individual flows. It is implemented by stateless priority scheduling in the core routers, in collaboration of stateful resource management at the network edges. To receive different levels of QoS, application packets are assigned to different service types or traffic classes at the network edges [21]; DiffServ-compatible routers in the network core perform stateless prioritized packet forwarding, so-called "per-hop behaviors" (PHBs), to the classified packets. Due to its per-class stateless routing, the DiffServ architecture exhibits a good scalability.

Early PHB proposals of DiffServ focused on the construction of versatile end-to-end services with guaranteed QoS. Two examples are "expedited forwarding" [10] and "assured forwarding" [9]. An alternative to absolute DiffServ is a relative differentiated services model to quantify the difference of QoS between classes of traffic. In this model, the network traffic is divided into a number of classes

* Corresponding author. Address: Department of Mathematics and Computer Science, South Dakota School of Mines & Technology, Rapid City, SD 57701, USA. Tel.: +1 13135775147; fax: +1 13135771101.
E-mail addresses: Jianbin.Wei@sdsmt.edu (J. Wei), czxu@wayne.edu (C.-Z. Xu), zbo@cs.uccs.edu (X. Zhou), liqing@wayne.edu (Q. Li).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.06.009

with ordered QoS requirements in such a way that the traffic of a higher ranked class receives better (or at least no worse) services than the traffic of lower ranked classes, in terms of local (per-hop) metrics like queueing delay and packet loss [5]. The Internet traffic is classified by applications and users at the network edges according to various service-level cost/performance agreements and policy constraints. Due to the lack of admission control or resource reservation in the network core, relative DiffServ provides no QoS guarantee to services. However, with the support of server-side QoS adaptation, DiffServ-capable routers assure end-to-end relative service differentiation. Although absolute DiffServ is desired to Internet services like audio/video streaming applications that have hard time constraints, relative DiffServ with respect to delay is sufficient to soft real-time applications like e-Commerce transactions.

Recently, Dovrolis, et al. defined a proportional delay differentiation (PDD) model in support of relative DiffServ [6,7]. It ensures the quality spacing between classes of traffic to be proportional to certain pre-specified class differentiation parameters. Since then, many packet scheduling algorithms have been developed to implement the PDD model. Representatives of the PDD algorithms include backlog-proportional rate (BPR) [6], joint buffer management and scheduling (JoBS) [14], waiting-time priority (WTP) [7], adaptive WTP [12], hybrid proportional delay (HPD) [7], and mean-delay proportional (MDP) [18]. They demonstrated various characteristics in support of the PDD model in different class load conditions and different time-scales. Most of them are capable of achieving desired delay ratios, if the ratios are feasible, under heavy load conditions and in long time-scales. For example, HPD takes into account the delay of head-of-line packet of a backlogged class and the delay of departed packets simultaneously and achieves desired delay ratios in both short and long time-scales on average when the delay ratio is small. However, it yields large ratio variations in statistics in short time-scales. For large desired delay ratios, large relative errors (on average) are observed as well. Details of the PDD algorithms are reviewed in Section 2.

In this paper, we present a new PDD algorithm, called Little's average delay (LAD), based on a proof of Little's Law. Little's Law regarding a queueing system states the *stationary* relationship between queue length, arrival rate, and queueing delay on average in the long run [15]. Its proof reveals a *transient* property regarding the queueing length [22]. That is, the queueing length of a class at any time is equal to the product of the traffic arrival rate and the waiting time of backlogged packets, plus the experienced delay of departed packets. Accordingly, LAD monitors the average arrival rate of every traffic class and the queueing delay of arrived packets, including both the waiting packets in the queue and departed packets for the purpose of controlling the delay ratio in both long and short time-scales.

Simulation results show that LAD overcomes the limitations of its main competitors: AWTP, HDP, and MDP. Specifically, whenever the PDD model of a desired class delay ratio is feasible, LAD is capable of providing more accurate and robust control over the delay ratio than its competitors in short time-scales. The improvement is significant when the desired delay ratio is large. In long time-scales, LAD performs no worse than its competitors under any load conditions. Moreover, the performance of LAD is independent of the distributions of packet arrivals and packet sizes because of the generality of Little's Law.

The remainder of the paper is organized as follows. Section 2 gives an overview of the PDD model and a brief review of the existing PDD algorithms. Section 3 presents the LAD algorithm and discusses its design and implementation issues. Section 4 evaluates the algorithm via extensive simulation and compares it with other PDD algorithms. We conclude this paper in Section 5.

2. Background and related work

2.1. Proportional Delay Differentiation Model

We consider packet scheduling of a lossless, work-conserving, and non-preemptive link that services M ($M \geqslant 2$) first-come-first-served (FCFS) queues, one for each traffic class (Hereinafter the terms "queue" and "class" will be used interchangeably). The lossless property requires that the average arrival rate of the aggregate traffic must be less than the link capacity and that there is enough queueing space to buffer backlogged packets. The work-conserving property is that the link is never left idle as long as there are backlogged packets waiting for service in the queues. The non-preemptive property requires the transmission of a packet cannot be interrupted. It is assumed that the traffic in different classes has independent arrival and packet size processes. Therefore, the aggregate traffic of the queueing system is determined by the superposition of the M traffic streams. Denote λ_i the arrival rate of class i, $1 \leqslant i \leqslant M$. It follows that the arrival rate of aggregate traffic of the system, $\lambda = \sum_{i=1}^{M} \lambda_i$. Let C represents the link capacity. The system utilization rate $\rho = (\sum_{i=1}^{M} \lambda_i x_i)/C$, where x_i represents the average packet size of of class i.

The objective of the PDD model is to control the quality spacing between different classes so that their average delay ratios be proportional to certain class differentiation parameters pre-defined by network operators. Let W_i denote the average delay of class i, and δ_i the pre-defined delay differentiation parameter. The PDD model requires to ensure that for any two classes i and j, $1 \leqslant i, j \leqslant M$,

$$\frac{W_i}{W_j} = \frac{\delta_i}{\delta_j}. \tag{1}$$

Notice that the PDD model is not always feasible. Because of the additional constraint of the Conservation Law in priority queueing systems, there may not necessarily exist a work-conserving scheduler that can meet the constraint of (1). It is known that the average delay of a class has a minimum value due to its inherent class load and the minimum value can be achieved by the use of a strict priority