Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 2198–2216
Computer Networks
www.elsevier.com/locate/comnet

A Fokker–Planck equation method predicting Buffer occupancy in a single queue

Dawn S.L. Dolcy, Costas C. Constantinou *, Steven F. Quigley

Department of Electronic, Electrical and Computer Engineering, University of Birmingham, Edgbaston Birmingham, West Midlands B15 2TT, United Kingdom

Received 4 May 2005; received in revised form 20 October 2006; accepted 25 October 2006
Available online 29 November 2006

Responsible Editor: Udo R. Krieger

Abstract

The VIrtual Predictor BuffER (VIPER) algorithm is a novel algorithm for performing online prediction of the buffer space requirement of each competing traffic stream. It accomplishes its task through the employment of a theoretical, infinite capacity, virtual buffer. Information acquired from the virtual buffer is used to construct a probability distribution function that is based on the Fokker–Planck equation. This distribution function is central to the VIPER algorithm and is used to compute the queue length predictions. The predictions are shown to be promising.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Queuing theory; Dynamic buffer allocation; Queue management

1. Introduction

Until recently, the Internet infrastructure offered mainly a Best Effort (BE) service that was based on the availability of resources rather than on guarantees. Guarantees were implicitly provided through over-provisioning of resources such as link bandwidth, router buffer space and switching capacity. More recently a differentiated service (Diffserv) model has been adopted, in which resources are allocated differentially to various classes of traffic, according to a strict prioritisation scheme. A consequence of this approach is that a high priority class queue may be idle at the same time as a lower priority class queue overflows, unless gross over-provisioning of queue memory is made across all quality of service (QoS) classes. A much more highly desirable architectural arrangement is to reallocate both queue memory and queue service rates across the various router ports, as well as QoS classes, dynamically. The subject of this paper is a measurement-based, dynamic queue memory reallocation algorithm.

The paper is structured as follows: first we present a review of existing dynamic resource management mechanisms in networks. We then present a model of the traffic arrival and service processes for a single queue buffer, capable of predicting the

* Corresponding author. Tel.: +44 1214144303; fax: +44 1214144291.
E-mail address: c.constantinou@bham.ac.uk (C.C. Constantinou).

1389-1286/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2006.10.014

buffer occupancy on a pertinent time-scale from its recent history, which is discussed in outline. Subsequently, a queue length adaptation algorithm is developed based on this model and tested using real traffic data traces, as well as more taxing traffic traces which deliberately violate some of the model's assumptions in order to demonstrate the capabilities of the advocated approach. Finally, the paper concludes by presenting an outline architecture for a more complete adaptive resource allocation system.

2. QoS Mechanisms

Current network resource allocation methods can be classified as being either static or dynamic. A static method will allocate a predetermined amount of resources to a given data stream before transmission begins. Dynamic allocation methods periodically renegotiate the apportioned resources based on predicted requirements.

Several mechanisms for the provision of QoS guarantees are currently available. These mechanisms can be categorised as either data path or control path operations [1]. Data path mechanisms generally implement actions that need to be performed by the router on a per-packet basis in order to afford individual QoS commitments. The basic data path operations include packet classification, marking, metering, policing and shaping. Control path mechanisms are concerned with issues of network node configuration and utilise rule sets for resource usage. For example, a packet being serviced in IntServ's guaranteed service class will only be handled in the expected manner if the router has the correct configuration data installed. Admission control, policy control and bandwidth brokers are all control path mechanisms [1]. The data path mechanism is discussed in more detail in the following section.

2.1. Data path mechanisms

Data path mechanisms play an important role in the realisation of efficient and effective resource management. Fig. 1 [1] illustrates the control flow and relationship between the various data path elements. Together their functions contribute toward the realisation of a QoS-aware network architecture. This is achieved through the careful management of resources on a packet-by-packet basis. For the purposes of this paper the word "resources" will primarily be used to refer to bandwidth and buffer space in the routers or switches. The QoS performance metrics of interest are the observed loss rate and the packet switching delay.

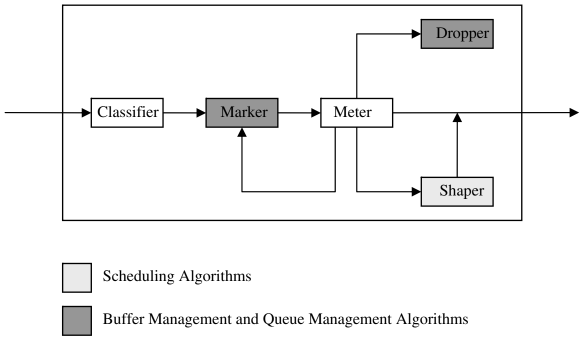

Fig. 1. Basic data path operations [1].

Both the Marker and Dropper units in Fig. 1 have a profound impact on the experienced loss rate. The set of procedures normally linked to these units are known as buffer management and queue management algorithms, respectively.

Buffer management algorithms allocate buffer space to the packets that traverse the router. A good buffer management policy will not only maintain high throughput through the router, but will also guarantee fair distribution of the buffer resource among packets originating from multiple sources. To achieve good overall throughput, the overall packet loss probability must be minimal. To achieve fairness, packets from conformant sources should be shielded from those originating from non-conformant ones. Buffer management can be either static (i.e. each flow has a static bound on the amount of space it can occupy in the buffer) [2–6] or dynamic (i.e. the memory allocated to each flow can be changed depending on traffic conditions) [7–15].

Queue management algorithms regulate the queue length, as opposed to the regulation of the total buffer space allocated to a particular flow or aggregate of flows. The queue manager could also be used to influence the flows that make up the queue occupancy distribution. It performs its tasks by determining which packet needs to be dropped or marked and when is the appropriate time to do this. Queue management algorithms can generally be classified into two broad categories depending on the way they respond to heavy traffic load, namely management for congestion recovery [16,17] and management for congestion avoidance [18–23].

The Shaper module affects delay and is mainly implemented by scheduling algorithms. Queue management and scheduling are closely related although

Download English Version:

https://daneshyari.com/en/article/452728

Download Persian Version:

https://daneshyari.com/article/452728

Daneshyari.com