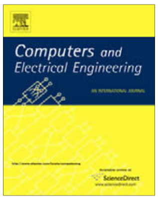
Computers and Electrical Engineering 36 (2010) 718–734

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng



A novel resource scheduling algorithm for QoS-aware services on the Internet

Fariza Sabrina [*]

ICT Centre, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Cnr Vimiera and Pembroke Roads, Marsfield, NSW 2022, Australia

ARTICLE INFO

Article history:
Available online 21 December 2008

Keywords:
QoS
Resource scheduling

ABSTRACT

The popularity and availability of Internet connection has opened up the opportunity for network-centric collaborative work that was impossible a few years ago. Contending traffic flows in this collaborative scenario share different kinds of resources such as network links, buffers, and router CPU. The goal should hence be overall fairness in the allocation of multiple resources rather than a specific resource. In this paper, *firstly*, we present a novel QoS-aware resource scheduling algorithm called *Weighted Composite Bandwidth and CPU Scheduler (WCBCS)*, which jointly allocates the fair share of the link bandwidth as well as processing resource to all competing flows. WCBCS also uses a simple and adaptive online prediction scheme for reliably estimating the processing times of the incoming data packets. *Secondly*, we present some analytical results, extensive NS-2 simulation work, and experimental results from our implementation on Intel IXP2400 network processor. The simulation and implementation results show that our low complexity scheduling algorithm can efficiently maximise the CPU and bandwidth utilisation while maintaining guaranteed Quality of Service (QoS) for each individual flow.

1. Introduction

Provisioning smart, efficient, dynamic collaborative service has drawn huge interest from industries and as a result has become a challenging research issue. Many of these applications in collaborative environment need processing the packet upon arrival, before transmitting it to the clients [1,2]. Many applications have strict delay bound where as others are intolerable of packet loss [2]. The best effort Internet, with no guarantee of network capacity or packet delivery, is a challenge for the real time interaction required for most of these collaborative services. Efficient resource allocation in such a system is an important and fundamentally complicated problem. In order to satisfy QoS requirements of various applications the node must control the use of network and processing resources by properly scheduling them. The system must ensure that all the flows receive their reserved resources while QoS is also maintained. To ensure this, there must be mechanisms to give guaranteed bandwidth and computational resources to incoming flows. However, allocation of bandwidth and CPU resources are interdependent and maintaining fairness in one resource allocation does not necessarily entail fairness in other resource allocation. Therefore, for better maintenance of QoS guarantees and overall fairness in resource allocations for the contending flows, the processor and bandwidth scheduling schemes should be integrated.

A significant amount of work has been done in bandwidth resource scheduling for traditional network. Packet Fair Queueing (PFQ) disciplines such as WFQ and WF^{2}Q [3] provide perfect fairness among contending network flows. However, WFQ and WF^{2}Q cannot readily be used for processor scheduling because they require precise knowledge of the processing times for the incoming packets at time of their arrival in the node. Moreover, the work complexity of these algorithms are $O(N)$,

---

[*] Tel.: +61 2 9372 4148; fax: +61 2 9372 4585.
E-mail address: Fariza.Sabrina@csiro.au

0045-7906/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2008.11.006

where N is the number of flows sharing the link. Another PFQ algorithm for bandwidth scheduling is Start-Time Fair Queueing (SFQ) [4], which does not use packet lengths for updating virtual time, and therefore seems suitable for scheduling computational resources (since it would not need prior knowledge of the processing times of packets) [4]. However, the worst-case delay under SFQ increases with the number of flows and it tends to favor flows that have a higher average ratio of processing time per packet to reserved processing rate [6]. One algorithm called deficit round-robin (DRR) achieves fair scheduling with $O(1)$ complexity. But it is used only for link bandwidth scheduling.

A large amount of work has also been done on CPU scheduling [7,8], but most of them are on CPU scheduling for end systems and work on task level (not on packet level). Moreover, the processing times of various applications on packets are not known in advance, thus constraining efficient and fair processor scheduling algorithms, which in turn limits the applicability of well-known bandwidth scheduling algorithms and also makes explicit or implicit admission control at the flow level more difficult.

Pappu et al. [6] presented a processor scheduling algorithm for programmable routers called Estimation-based Fair Queueing (EFQ) that estimated the processing times of various applications on packets of given lengths off-line and then scheduled the processing resources based on the estimations. Fixed values of the estimation parameters measured off-line may not always produce good estimations due to variation in server load and operating system scheduling. Galtier et al. [9] proposed a scheme to predict the CPU requirements of executing a specific code on a variety of platforms. Doulamis et. al. [11] used least square algorithm to predict task work load and used this information for resource scheduling in grid computing. However, all these schemes seem too complicated to be implemented in routers.

All the scheduling schemes discussed above are designed to schedule only a single resource, i.e., either bandwidth or processing resource. Unlike all the previous works, our work takes an integrated approach and provides a composite scheduler for both bandwidth and CPU scheduling in order to provide better QoS guarantees to the contending data flows. In our previous work [12,13], we presented a composite bandwidth and processor scheduler called *Composite Bandwidth and CPU Scheduler (CBCS)*, which can schedule multiple resources adaptively, fairly and efficiently among all the competing flows. Detailed simulation, analytical and experimental work presented in [12] proves that our novel idea of integrating the CPU and bandwidth scheduling functionalities within a single scheduling scheme can provide significantly better delay guarantees than those achievable through separate resource schedulers.

Although, CBCS has high efficiency but it was developed only for best effort flows and does not ensure flow differentiation. In this paper, we present our new composite scheduling algorithm called Weighted Composite Bandwidth and CPU Scheduler (WCBCS), which is the extended version of CBCS to make it suitable for QoS flows. The novelty of this algorithm is, (1) it schedules multiple resources in a single algorithm, (2) it employs a simple and adaptive online prediction scheme called modified Single Exponential Smoothing for determining the packet processing times. (3) it is suitable for QoS flows where flows with different reserved rate are weighted differently, but unlike other scheduling algorithm of similar capability WCBCS has very low work complexity ($O(1)$), making it attractive for implementing in high-speed routers.

The paper is organised as follows: Section 2 presents the proposed algorithm. Section 3 presents some analytical results and Section 4 describes details of the simulation setup and analyses the performance of the scheduler through simulation. Section 5 presents the hardware implementation details and Section 6 presents some experimental results. Conclusions are drawn in Section 7.

2. WCBCS – Weighted Composite Bandwidth and CPU Scheduler

This section describes the WCBCS scheduler and prediction technique used to estimate the packet processing duration.

2.1. Online prediction process

Since the processing requirement of each packet is not known apriori, the WCBCS scheduler needs to estimate the processing duration for each arriving packet. We have investigated several smoothing methods and their suitability for predicting the processing requirements of the packets. A detailed analysis, investigations and a comparative performance analysis of the alternatives are discussed in [13]. Our investigations show that the Single Exponential Smoothing (SES) technique is well-suited to estimate the processing times of the packets. SES is computationally simple and an attractive method of forecasting. SES uses the following equation to calculate a new predicted value

$$F_{t+1} = \alpha X_t + (1-\alpha)F_t \quad \textit{where} \quad 0 \leqslant \alpha \leqslant 1 \tag{1}$$

where F_t and F_{t+1} are the predicted value at tth and $(t+1)$th period respectively. X_t is the actual duration required to process the packet that arrived at time t, and α is the SES coefficient which determines the relative weight allocated to the history and the current estimated sample. In our work, the SES coefficient, α, was set to 0.4, based on earlier experimentation, which indicates that a value of 0.4 provides the most accurate results.

Most of the packets that are processed by today's routers can be broadly classified into two categories based on their processing needs: (a) header processing and (b) payload processing. Header processing application (i.e., IP forwarding) only requires read and write operations in the header of the packet and so the processing complexity is independent of the size of the packets. In contrast, payload processing application (such as IPSec Encryption, packet compression and packet content transcoding, etc.) involves read and write operations on all the data in the packet, and therefore the processing complexity