Computer Communications 31 (2008) 3427–3437

Contents lists available at ScienceDirect

Computer Communications

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

Fair link striping with FIFO delivery on heterogeneous channels

Jingnan Yao *, Jiani Guo, Laxmi Bhuyan

Department of Computer Science, University of California, 900 University Avenue, Riverside, CA 92521, USA

ARTICLE INFO

Article history:
Received 14 June 2007
Received in revised form 15 May 2008
Accepted 19 May 2008
Available online 14 June 2008

Keywords:
Link striping
Out of order
Packet
Scheduling

ABSTRACT

Link aggregation techniques are often used to achieve higher communication bandwidth by striping network traffic across multiple transmission channels. Due to the variations in bandwidth, latency and loss rate on different channels, link striping suffers from packet reordering thereby adversely affecting the performance of any QoS concerned applications. Hardware-based solutions often prolong transmission latency which is undesirable for delay sensitive applications and are restricted with the available buffer space on the device. Thus, an effective striping protocol that ensures both load balancing and minimal packet reordering is important when striping traffic onto multiple channels.

In this paper, we first propose an sequence preserving scheduling (SPS) scheme to schedule packets among multiple heterogeneous communication channels assuming that the workload is perfectly divisible. Packets assigned onto different links for transmission are ordered perfectly by applying divisible load theory (DLT). We analyze the throughput and derive expressions for the batch size, scheduling time and the maximum number of channels that can be supported by the sender and receiver. Further, to effectively schedule variable length packets for link striping, we propose a packetized sequence preserving scheduling (P-SPS) scheme by applying a combined packetized technique of deficit round robin (DRR) and surplus round robin (SRR). Extensive sensitivity results are provided through analysis and simulation to show that the proposed algorithms satisfy both the load balancing and in-order requirements for efficient packet transmission.

1. Introduction

Scalable architectures over distributed environments have gained popularity in many research areas. With the prevalence of the web, immense amount of distributed resources can be harvested over the Internet in a cooperative manner. To cope with the ever increasing speed of the servers and workstations, it is desirable to boost the data transfer rate to/from the Internet. Scalable transmission mechanisms [1,2] are preferred to overcome the transmission bottleneck in computer networks.

Given that the network bandwidth on a single channel is limited, link aggregation techniques are employed to provide multiple physical links from a source to the same destination. By sending a fraction of data over each of the several independent communication links, link aggregation provides increased bandwidth and reliability between the two devices (switch-to-switch or switch-to-station) as more channels are added. In many LAN/WAN systems, link striping technique is also used to avoid the usage of expensive higher data rate single link connection. Implementations include the Cisco Etherchannel in the CISCO ONS 1500 Series based on the proprietary inter-switch trunking (ISL), Adaptec's Duralink port aggregation, IEEE 802.3ad for Ethernet link aggregation, etc. Inverse multiplexing [3] and Stripe protocol [4] are example techniques that were proposed in the context of ISDN, ATM and analog dialups that aggregate bandwidth over multiple links.

However, transmission of packets belonging to the same flow in different communication channels gives rise to out-of-order arrival of the packets at the receiver and incurs high delay jitter for the outgoing traffic. For TCP, it has been proved that out-of-order transmission of packets is detrimental to the end-to-end performance [5]. Although TCP is designed to handle reordering, this involves additional processing at the TCP end points. TCP for persistent reordering (TCP-PR) [6] is a variant of TCP that attempts to improve TCP performance in the presence of persistent reordering phenomena. TCP-PR is a software-based solution and thus is restricted only at the end hosts. For many applications like multimedia transcoding and VoIP, it is imperative to minimize this out-of-order effect because the receiver may not be able to reorder them easily to tolerate high delay jitter. Therefore, efficient packet scheduling is necessary in order to guarantee both high throughput and minimal out-of-order delivery of packets. However, these two goals are contradictory to each other because scheduling on more transmission channels improves throughput but also increases the out-of-order delivery of packets.

Existing channel striping algorithms [4,7–10] vary from simple static policies, such as round robin or random selection policy, to

* Corresponding author. Tel.: +1 408 306 0988.
E-mail address: jyao@cs.ucr.edu (J. Yao).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.05.042

fair scheduling policies such as surplus round robin and elastic round robin, to sophisticated adaptive load-sharing policies such as shortest queue first scheme and address-based hashing scheme. All these schemes either do not provide first-in-first-out (FIFO) delivery or do not provide load sharing for packets addressed to different destinations. Simple policies such as round robin are employed in practice because adaptive schemes are difficult to implement and involve considerable overhead. It is shown that round robin is simple and fast, but provides no guarantee to the performance because it causes large out-of-order arrival of the packets. Adaptive load sharing schemes on the other hand, achieve better unit order in output streams, but involve higher overhead to map the packets to an appropriate channel. Scheduling for multi-link transmission systems is analogous to load balancing on multi-processor computing systems. In fact, if the time for packet transmission on a link is replaced with an estimated processing time of a packet, the two problems are equivalent. Recently, we proposed two scheduling algorithms called dynamic batch co-scheduling [12] and quantum adaptive scheduling [11] for packet scheduling across multiple processing engines of a network processor.

A practical link striping protocol, called surplus round robin (SRR), is proposed by Adiseshu [13] to schedule variable length packets over multiple links with different capacities. They also demonstrate that striping is equivalent to the classic load-balancing problem over multiple channels. They solve the variable packet size problem by transforming a class of fair queuing algorithms into load sharing algorithms at the sender. Although their solution is elegant and efficient, it does not consider the transmission order among the packets in different channels. Hence, the receiver needs to run a re-sequencing algorithm to restore the packet order in the original flow. A strict synchronization between the sender and the receiver is difficult to implement. Cobb and Lin considered several sorting techniques in their paper [14] to avoid packet reordering that require access to upper layer protocol headers and thus potentially incur significant overhead. Moreover, the time to move packets between the single input/output port of the sender/receiver and different channels is assumed to be negligible in most of the channel striping papers. There should at least be a time overhead in executing the scheduler at the IP level processing which is appropriately modeled in our scheme.

The aim of this paper is to derive an efficient packet-scheduling algorithm in a link striping model that comprises a number of channels for packet transmission. It should provide (1) load balancing for processing variable length packets using a group of heterogeneous channels, and (2) in-order delivery of packets without considering receiver's rearranging capability. We derive an sequence preserving scheduling (SPS) algorithm that considers a backlogged queue, and can be adapted to dynamic arrival of packets. Like other packet scheduling algorithms such as RR/SRR, SPS is essentially a scheduler based algorithm. Most of the sequence control schemes [15,20] in the literature are based on maintaining extra sequence number or pointers in the packets. Our goal, however, is to minimize packet out-of-order without incurring extra overhead and without throughput degradation. All the control is at the sender and there is no extra mechanism or information maintained to keep the sequence.

The divisible load theory (DLT) [16,17,19] develops scheduling assuming that the workload is perfectly divisible for such distribution among the processors, so that all processors finish at the same time. It considers heterogeneous processors and different communication times to send data to those processors. DLT is particularly suitable for data parallel operations, where the volume of data can be perfectly distributed without causing any error. DLT is highly applicable in parallel processing of many applications [17]. We have also demonstrated that DLT can be applied to packet processing in a network with good accuracy even though a packet is not divisible [12]. However, its possible use in link striping has not been explored earlier. As required for packet transmission over multiple channels, DLT has to be tuned to consider sequential ordering of the packet transmission times. Our algorithm schedules the packets in batches by computing the minimal batch size, scheduling time, and number of schedulable links given the maximum packet size and channel bandwidths. A batch is similar to the concept of time epochs when scheduling is done. Several interesting results are derived regarding scalability of our algorithm.

Expressions for load distribution in heterogeneous transmission channels are derived first by assuming that the schedulable workload is perfectly divisible in bytes. Since the arriving packets cannot be distributed to different links in bytes, we derive a packetized version of the SPS algorithm by applying a combined version of deficit round robin (DRR) and SRR algorithms [18,13]. The packetized SPS (P-SPS) algorithm produces better results in terms of throughput and out-of-order rate compared to round robin and pure SRR schemes. To completely eliminate packet out-of-order, we propose a complement method to be used along with the P-SPS main algorithm. Finally, we perform a number of simulations and sensitivity studies to verify the accuracy of our theory and obtain performance over wide-ranging input parameters.

The rest of the paper is organized as follows. In Section 2, we present the preliminaries and certain design issues for a link striping approach. In Section 3, we design the SPS algorithm by giving theoretical derivation and analysis. In Section 4, we propose and design a packetized version of the SPS algorithm named Packetized-SPS (P-SPS) to deal with variable length packets. Simulation results are presented in Section 5 in comparison with several other schemes. Finally, in Section 6, we conclude the paper with future possible extensions related to this paper.

2. FIFO link striping formulation

Fig. 1 illustrates the link striping model for a single traffic flow of variable length packets. In order to overcome transmission bottleneck of a single channel, M heterogeneous communication channels are deployed between the sender and the receiver. The sender and the receiver communicate with the I/O ports sequentially. The sender implements the striping algorithm to split the outgoing traffic stream and schedules the packets among multiple channels c_1 through c_M for parallel packet transmission. The receiver unloads packets from these channels on a first-come-first-serve (FCFS) basis and combines the traffic back into a single stream. Such a process is detailed as follows:

- A common scheduler resides at the sender node with a common buffer pool where the packets are stored.
- Each link is modeled as a transmitting entity with a fixed transmission rate preceded by a packet buffer.
- The scheduler arbitrates the entire packet dispatching process. Packets are selected and dispatched to the respective buffers of different links at every time epoch. Quantum for each link is predetermined by our scheduling algorithm with respect to different link bandwidth available and the desired FIFO delivery pattern.
- Packets are dispatched serially onto the channels from the sender but are transmitted in parallel.
- Similarly at the receiver, packets are removed from the transmission channels sequentially in the same order as they arrive. If multiple packets arrived at the receiver simultaneously, the channel ID is used to break the tie. The receiver will get the packet from the channel with smaller ID first.