Study of Three Stage Load Balancing Switches

G. ANUSHA\(^1\), K. SRINIVASAN\(^2\)

\(^1\)Asst Prof, Dept of CSE, Annamacharya Institute of Technology and Science, Tirupati, AP, India, E-mail:anua13@gmail.com.
\(^2\)PG Scholar, Dept of CS, Annamacharya Institute of Technology and Science, Tirupati, AP, India, E-mail:ksreenivasan09@gmail.com.

Abstract: There has stayed an excessive arrangement of attentiveness newly in load-balancing adjustments owed to their humble construction and great furthering high frequency. However, the mis-arrangement difficult of the unique load-balancing alteration delays the performance of fundamental TCP presentations. Some load-balancing control strategies have stayed projected to discourse this mis-arrangement problem. It explains this mis-arrangement difficult at the price of also algorithmic difficulty or exceptional hardware necessities. Here the discourse the mis-arrangement difficult by presenting three-stage load-balancing adjustment construction improved with a production load-balancing appliance. This three-stage load-balancing adjustment attains a high accelerating capability while protective the instruction of packages deprived of the essential of expensive accessible arrangement procedures. Theoretic examines and repetition outcomes display that this three-stage load-balancing alteration delivers a conduction deferment that is greater- restricted through that of an output-queue up adjustment desirable a persistent that be subject to individual on the amount of input/output havens, representing the similar advancing capability as an output-queued alteration.

Keywords: Virtual Output Queue (VOQ), Full Frame First (FFF) Algorithm, Full Ordered Frames First (FOFF) Algorithm, 3SLB Switch, etc.

I. INTRODUCTION

There has been a great deal of interest recently in load balancing switches due to their simple architecture and high forwarding capacity. A typical load-balancing switch consists of a two-stage switching fabric and a one-stage buffer sandwiched between the two switch fabrics where each fabric executes a periodic connection pattern. The idea behind this kind of design is that the input traffic with an arbitrary distribution becomes uniformly distributed through the first-stage switch fabric and then is fully forwarded by the second-stage switch fabric. Thus, high throughput is achieved while the architecture remains scalable. However, such a load-balancing switch suffers a mis-sequencing problem that is, the order of packets may not be preserved during the transmission process Mis-sequencing is undesirable because it hinders the efficiency of TCP connections. To address the mis-sequencing problem, several new load-balancing switch designs have been proposed in the literature. The mis-sequencing problem is resolved by either resequencing out-of-order packets at the outputs or preventing it from occurring. Here, two schemes with the first-in--first-out (FIFO) and the earliest deadline first (EDF) scheduling policies are proposed to resequence mis-sequenced packets at the output ports, respectively.

In the FIFO-based load-balancing switch, a jitter controller is placed in front of the second-stage virtual output queue (VOQ) to impose proper delays for packets so that they arrive to the resequencing buffer in sequence. In the EDF-based load-balancing switch, packets are scheduled out of the resequencing buffer according to their deadlines. In a Full Frame First (FFF) algorithm is used to forward packets over the coordination buffer implementation, known as the three dimensional VOQ, in the second stage, which guarantees that packets are delivered to the outputs in sequence. In Keslassy et al. propose a Full Ordered Frames First (FOFF) algorithm and introduce the usage of optical mesh in load-balancing switches. A comprehensive overview regarding load balancing switches is given in Keslassy’s Ph.D. dissertation. To improve the scalability of high-performance switches, Chang et al. propose in a Mailbox load-balancing switch architecture that makes use of a set of symmetric connection patterns to transmit packets and feedback signals between the first and second stages within the same time-slot. The Mailbox switch significantly reduces computation and communication overheads as compared to input-queued (IQ) switches and achieves a forwarding capacity as high as 75%. With heuristic refinements, the throughput of the Mailbox switch can be improved to higher than 95%.

II. PROPOSED SCHEMA

We introduce a new load-balancing switch, called the three-stage load-balancing (3SLB) switch, which effectively solves the mis-sequencing problem without the need of costly online scheduling algorithms or hardware speedup. The switch architecture studied in this paper was originally proposed by Wang et al. (for details, see their patented work). The work presented in this paper significantly extends these
initial efforts, providing a thorough theoretical analysis and evaluating its performance through extensive simulations. The most significant difference between the 3SLB switch and prior load-balancing switches is the third stage of the 3SLB switch in which packets are buffered in an output load-balancing fashion and forwarded in order of ascending arrival time. With the third stage, the order of packets is preserved without using any complex real-time scheduling algorithm; the overall online scheduling complexity of the 3SLB switch is. Thus, the 3SLB switch retains the scalability of prior load-balancing switches. The 3SLB switch has the following advantages.

Scalability: The complexity of the online scheduling algorithm inside a 3LSB switch is, and no hardware speedup is required.

Bounded delay: The transmission delay of the 3SLB switch is bounded by that of an OQ switch plus a constant that depends only on the number of input/output ports.

100% throughput: The 3SLB switch achieves the same throughput as an OQ switch for arbitrary input traffic patterns.

Low average delay: By simulation, we show that the 3SLB switch has a much lower average delay than existing IQ switches under heavy inputs as well as burst inputs.

III. SWITCH ARCHITECTURE
The architecture of the 3SLB switch and illustrates how packets are transmitted through the switch.

A. Preliminaries
Throughout the paper, we adopt the following conventions. Variable-length packets are segmented into cells of the same size after arriving at input ports. These cells are transmitted inside the switch and then assembled back into packets at output ports. In what follows, we use the terms “packet” and “cell” interchangeably. Moreover, time is slotted and synchronized so that at most one cell arrives at an input port during one time-slot and only one cell can be transmitted over a connected input–output pair of a switch fabric within one time-slot. Denote by the number of input/output ports of the switch.

B. Architecture Overview
We now introduce the switch architecture. Fig. 1 shows the architecture of an 3SLB switch consisting of three stages of buffering and switching. The input and output ports of the whole switch are called external inputs (EIs) and external outputs (EOs), respectively. In contrast, the input and output ports of the middle stage (i.e., the second stage) are called internal inputs (IIs) and internal outputs (IOs), respectively. Note that the IIs coincide with the output ports of the first stage, and so do the IOs with the input ports of the third stage. The EI, II, IO, EO ports are indexed with, respectively. In the 3SLB switch, each of the three switch fabrics is a crossbar running with a periodic sequence of connection patterns such that every input is connected to every output once during consecutive time-slots, and vice versa. Concretely, we choose for each crossbar a periodic sequence such that every output is connected to every input in a round-robin (RR) fashion in order of increasing input port index. The details of the three stages are as follows.

Stage1: There are VOQs at each EI port. Each VOQ corresponds to one of the II ports. A combination of a flow splitter and a load balancer is placed at each EI port to evenly distribute packet flows into the VOQs, where a flow is defined as an EI–II pair. This kind of flow differs from a traditional flow that is usually defined as an EI–EO pair. Denote by S1-VOQ the VOQ at EI port. All packets buffered in S1-VOQ are designed to be transmitted to II Ports are stored in S1-VOQ in an FIFO manner.

Stage2: There are VOQs at each II port. Each VOQ corresponds to one of the IO ports. Denote by S2-VOQ the th VOQ at II port. All packets buffered in S2-VOQ are designed to be transmitted to IO port. Packets are stored in S2-VOQ in an FIFO manner.

Stage3: Between the second and third crossbars are memory banks (MBs), each lying in front of one of the inputs of the third crossbar. An MB is nothing but an array of contiguous memory blocks, where each memory block can hold one packet/cell. Each MB is composed of memory blocks and is evenly divided into segments. Circular queues (CQs) are logically built across the MBs in a way that every CQ comprises segments aligned vertically, each coming from one MB. Each CQ exclusively serves one of the EO ports: all packets with the same destination EO port are buffered in the corresponding CQ, being organized in a top-down right-left manner, according to their arrival times.

For queue management purposes, a global agent (GA) is associated with each CQ that keeps track of the beginning and end positions of a circular queue inside the CQ, where a position is defined in terms of the coordinates (row No., column No.) of the memory block. Notice that the row No. of a block is also the index of the IO port where it resides. For simplicity, the leftmost to the rightmost columns in a CQ are numbered from 1 to M.
IV. EXPERIMENTAL RESULTS

Experiments are made using the prototype application, a custom simulator, built using JAVA platform. The details of real traces used in experiments are presented in table 1. Statistics of the traces and also experimental results are presented in this section. Figs. 2–4 show the average packet delay versus the probability under the four traffic models. From them, we make the following observations.

V. CONCLUSION

In this paper, we introduced a three-stage load-balancing switch and studied its performance via theoretical analysis and experimental simulation. In particular, we demonstrated a method to resolve the mis-sequencing problem by using a nested load-balancing scheme (i.e., a combination of the input and output load-balancing mechanisms). Moreover, we rigorously proved that the transmission delay of the 3SLB switch is bounded by that of an OQ switch plus a constant that depends only on the number of the input/output ports, and, as a result, it achieves the same forwarding capacity as the OQ switch. By simulations, we showed that the constant difference between the average packet delays of the 3SLB and OQ switches holds for several typical input traffic models. Finally, we extended our analysis onto the overheads involved in the 3SLB switch as well as their impacts. The output load-balancing mechanism is critical to the performance of the 3SLB switch in that in combination with input load-balancing, it produces finite queue lengths at the first two stages, regardless of the input traffic pattern. This property implies that packets are buffered only in the third stage, i.e., in circular queues, in the event of congestion. In this sense, the circular queues in the 3SLB switch correspond to the output queues in the OQ switch: A circular queue is always busy whenever its length exceeds a certain threshold. Such a property leads to the relationship between the 3SLB and OQ switches, that is, the length of a circular queue is upper-bounded by that of an output queue, plus a constant, which in turn implies the identical forwarding capacities of them, as shown in Theorem 4. Like prior load-balancing switches, the 3SLB switch exemplifies the possibility of achieving high performance using common hardware resources. We believe that the deterministic operations of the 3SLB switch help facilitate more hardware-based packet-forwarding functionalities.

VI. REFERENCES


Author’s Profile:
G. Anusha working as an Assistant Professor, Department of CSE, Annamachary Institute of Technology and Science, Tirupati, A.P., INDIA.

K. Sreenivasan has received his B.Tech from Vemu Institute of Technology, P. Kota, Chittoor, A.P., INDIA and Pursuing his M.Tech, Dept of CS, in Annamacharya Institute of Technology and Science, Tirupati, A.P., INDIA.