Load-Balancing Multipath Switching System with Flow Slice

P.Poojitha[#1], G.Suhasini[*2]

[#1]M.Tech, CSE, Ganapathy Engineering College, Warangal, Andhra Pradesh, India
[2]Asst. Professor, Ganapathy Engineering College, Warangal, Andhra Pradesh, India

Abstract-- Load balancing plays a pivotal role in core routers as they need to handle multiple requests at a time. To achieve load balancing Multipath Switching Systems (MPS) are widely used. One of the challenging issues in building MPS is to ensure load balancing across multiple paths besides keeping the order of intraflow packets intact. The existing solutions are packet – based and they have drawbacks as they cannot scale and cause delay. Hashing algorithms that are based on flow-size also could not perform well due to improper distribution of flows. Recently Shi et al. presented a scheme "Flow Slice" to overcome this problem. According to this scheme each flow is cut into multiple slices based on a given threshold. In this paper we built a prototype application which is a custom simulator that demonstrates the usefulness of FS scheme in terms of load balancing efficiency. The experimental results revealed that the FS scheme is effective in load balancing.

Index Terms – Load balancing, flow slice, multipath switching systems

I. INTRODUCTION

In real world applications, core routers play an important role in processing requests. In order to process huge number of requests, core routers need a mechanism for load balancing. For this they depend on multipath switching systems as they play indispensable role to achieve state-of-the-art load balancing performance of core routers. The MPS is achieved by using various products from different companies. They include LBvN switches [1], [2], Parallel Packet Switch (PPS), Vitesse switch chip family [3], and Benes Multistage Switches in Cisco CRS-1 [4]. One of the challenging problems with MPS is to balance load across different switching paths in order to meet objectives such as uniform load sharing, Intraflow packet ordering, and Low timing and hardware complexity. When packets are dispatched one by one, packet based solutions are adequate in order to balance the load. The problem with this scheme is that the packets of a flow may be separated from that flow. Restoring packet orders in respective flows is the main problem here. To overcome the problem timestamp-based resequencers came into existence [5], [6], and [7]. They use delay concept in order to ensure the packets are in the correct flow at the end. This results in more average delay and thus proved to be an infeasible solution. Other kind of resequencers used sequence numbering of packets but it resulted in huge number of resequencers [8], [9], and [10]. Packet out of order is the main problem of many solutions. The resequencers also caused other problems like computational overhead. To overcome this problem, flow-based solutions came into existence [11], [12], [13], and [14]. These techniques try to send packets in the same flow while balancing the load using hashing concept. Load imbalance is the problem faced by hashing solutions. Figure 1 shows the problem with various techniques.

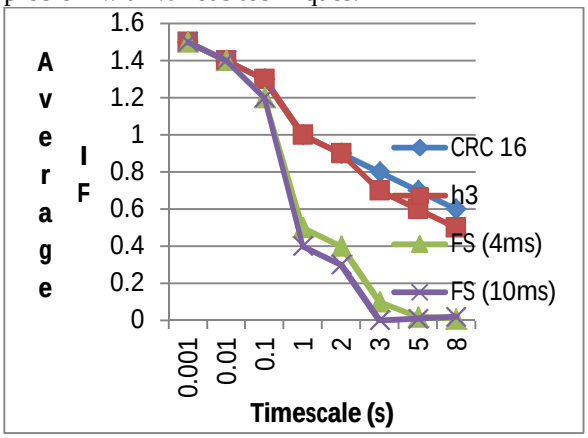

Fig. 1 – Average IFs for different techniques and timescales

As can be seen in figure 1, imbalance factor is presented for various techniques under different timescales. As the results show there is consistence imbalance stable for all techniques at 0.2 to 0.3. To

overcome this problem, later on, adaptive hashing [12], and [13] is used. However, this solution cannot work for MPS for many reasons such as load balancer is geographically separated, dual conflicting points exist. To handle the problem gracefully Shi et al. [15] proposed a new scheme known as Flow Slice. In this paper we practically implement Flow Slice. We built a custom simulator which demonstrates the proof of concept. The empirical results are encouraging. The remainder of the paper is organized as follows. Section II reviews literature. Section III presents proposed scheme. Section IV provides experimental results while section V concludes the paper.

II. RELATED WORKS

Many solutions came into existence for load balancing with MPS. However, there is a problem in which packets go out of flow. Turner [6] proposed timestamp-based resequencer for solving the problem. In this approach a contention resolution mechanism is implemented which ensures that the packets are not disturbed from corresponding flows. Time-wheel-like hardware [7] was proposed by Henrion in order to improve the performance of timestamp based resequencer. This solution also causes delay. To cope with the problem Turner [5] proposed an adaptive resequencer which can adjust the threshold dynamically. It improves performance when compared with fixed threshold solution. However, the average delay is still high with this solution. Other solutions started attaching a sequence number with each cell which also caused delay problem. To overcome this problem predefined patch scheduling protocol is used [16], [17], and [18]. SCIMA [9] is another solution proposed by Chiussi et al. [10] in order to deal with packet out of flow problem. However, this solution is not scalable and can't be used effectively with MPS.
Centralized scheduling algorithm was explored by Iyer et al. which ensure cells are ordered but suffers from complexity in communication. In [17] distributed scheduling was introduced which also need the use of resequencing process. Jitter control mechanisms were explored in [19] and [1] in the second stage of LBvN in order to overcome the delay problem. This work is also classified into timestamp based solution. Three dimensional queues were introduced in [16] between the stages of LBvN. It makes use of buffers for input port and the total N3 FIFOs. Static hashing was explored in [20] for load balancing. However, it is proved that CRC with 16 bit achieved excellent performance in load balancing. In [11] Weight approach was proposed which uses names of objects and weights are generated by servers thorough static hashing. It handles server failures gracefully. It also supports load balancing across heterogeneous servers. TCP's burstness was explored in [21] in order to improve load balancing. Based on this concept adaptive burst shifting [22] and flowset switching [21] were introduced. However, these solutions are not suitable for MPS. To overcome this problem, in [15] a novel scheme is introduced by name "Flow Slice". This scheme is useful to achieve MPS performance for load balancing while the packets are not disturbed from corresponding flows.

III. PROPOSED SCHEME

The proposed scheme for load balancing MPS is known as Flow Slice. Flow slice is a sequence of packets that belong to a flow. In that flow the packets have intraflow interval is less than a given threshold. By cutting flows it is possible to create miniflows with cumulative distribution functions (CDF) as shown in figure 2 where as the real traces used in the experiments are presented in table 1.

Trace Description (Collected Point)	Trace Name	Collect Time	Link Speed / Load Rate	Trace Duration	Packet Count	TCP Flow Count	Packet Length Avg. / SCV[1]
One hour continuous packet headers collected at North China CERNET backbone, 11/10/2005. (One of largest commercial backbones in China.)	Cernet-10	Nov. 10, 2005	OC-192c / 35.0%	63.1s	4.32E7	1.77E6	639.6Byte / 1.02
	Cernet-30	Nov. 10, 2005	OC-192c / 34.5%	63.1s	4.32E7	1.77E6	629.1Byte / 1.05
	Cernet-40	Nov. 10, 2005	OC-192c / 35.4%	62.4s	4.32E7	1.79E6	640.3Byte / 1.02
Continuous packet headers collected at Tsinghua University egress link to CERNET at June to July, 2006. Trace segments are as long as 10 minutes.	Ts-1	June 23, 2006	GE / 27.1%	347.4s	2.65E7	1.58E6	443.9Byte / 1.62
	Ts-3	July 12, 2006	GE / 25.5%	469.4s	2.73E7	1.46E6	547.3Byte / 1.26
	Ts-5	July 12, 2006	GE / 24.7%	511.0s	2.73E7	1.25E6	577.6Byte / 1.20
Continuous packet headers at network of American univ. and research labs by NLANR [2] during 2004–2006.	Pittsburgh SC Center	PSC	Mar. 8, 2006	OC-48c / 10.5%	89.2s	3.50E6	6.38E4	835.6Byte / 0.68
	Front Range GigaPOP	FRG	Dec. 5, 2004	OC-12c / 56.9%	89.6s	5.55E6	1.93E5	717.4Byte / 0.84
	Merit Abilene	MRA	Mar. 9, 2004	OC-12c / 56.2%	90.1s	6.12E6	4.48E5	645.9Byte / 1.03
	Univ. Florida at Gain	UFL	Dec. 5, 2004	OC-12c / 53.7%	89.8s	5.68E6	2.72E5	663.4Byte / 1.03
	Columbia Univ.	BWY	Mar. 2, 2004	2xOC-3c / 53.5%	90.1s	2.60E6	5.50E4	723.8Byte / 0.79
	Old Dominion Univ.	ODU	Dec. 4, 2004	OC-3c / 52.3%	89.4s	1.33E6	6.80E4	684.5Byte / 0.89
	Colorado State Univ.	COS	Dec. 5, 2004	OC-3c / 50.3%	90.2s	2.04E6	1.79E5	435.0Byte / 1.62

Table 1 – Real traces used in the experiments (excerpt from [15])

As can be seen in table 1, the trace decription, trace name, collectin time, link speed or load rate, trace duration, packet count, TCP flow count and packet length are provided. These details are used in the experimts with Flow Slice scheme.
Load Balancing Scheme for MPS

Figure 2 shows the proposed scheme and also compares it with packet based approach and flow based approach.

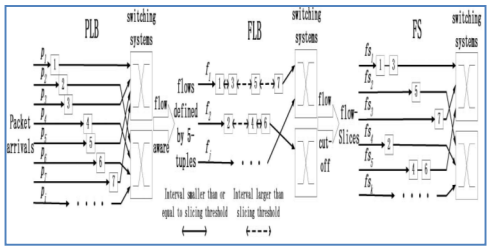

Fig. 2 – Flow Slice Scheme compared with other schemes

As can be seen in figure 2, the proposed load balancing scheme FS is compared with FLB and PLB. Out of all the schemes, the flow slice scheme is more efficient as it can handle all flows with respect to load balancing without disturbing the packet orders within any flow. The proposed solution met the three objectives reuiqred for optimal load balancing. They include flow based approach should improve switching performace; intra packet order is preserved; and the time complexity in dispatching packets is reduced greatly. More technical details about the FS scheme canbe found in [15].

IV. EXPERIMENTAL RESULTS

Experiments are made using the prototype application, a custom simulator, built using Microsoft .NET 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.

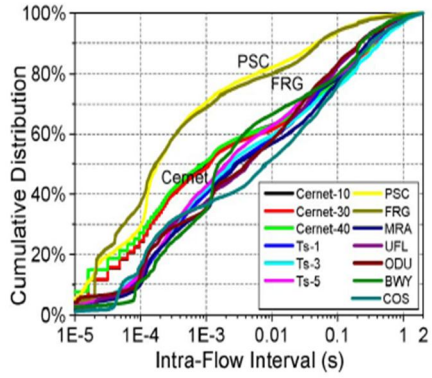

Fig. 2 – Intraflow interval CDF

As can be seen in figure 2, the infra-flow interval vs. CDF is shown for various traces. The intra flow intervals for various traces are presented. The graph is generated based on the real time flows. The properties used in the experiments include small size, light – tailed size distribution, and fewer active flow slices. The experimental results of these three properties are as shown in figure 3, 4 and 5.

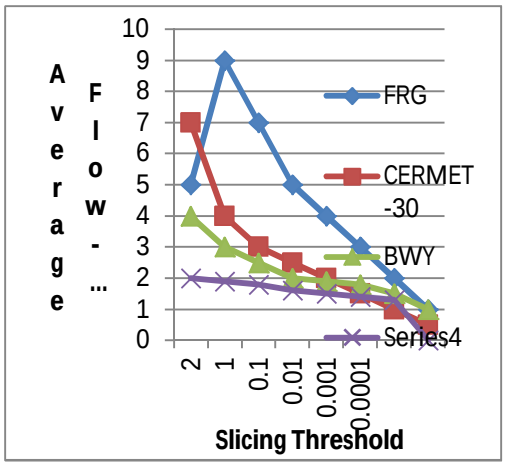

Fig. 3 – Illustrates Flow Slice Size

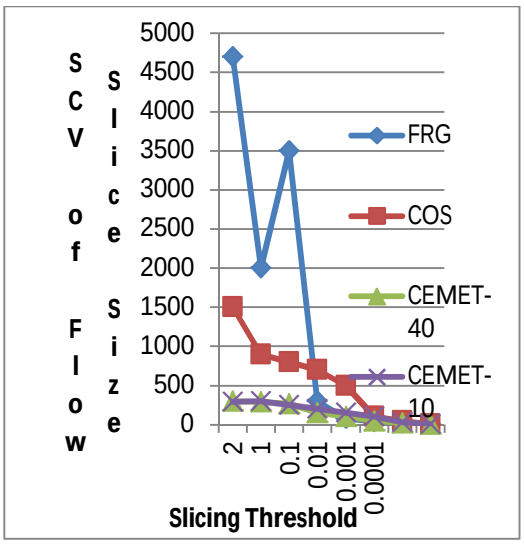

Fig. 4 – illustrates SCV of flow slice size

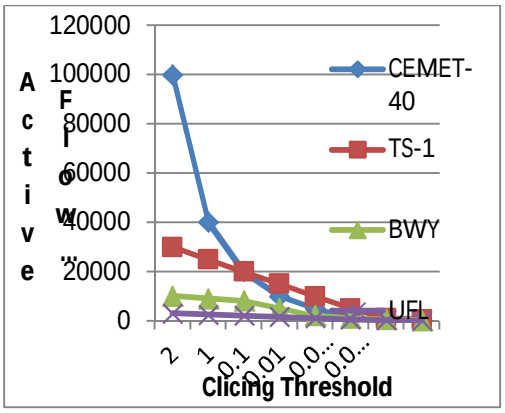

Fig. 5 – Illustrates active flow slice number

The extension made to multipath switching systems using parallel packet switches (PPS) is evaluated and the results are as presented in figure 6, 7 and 8.

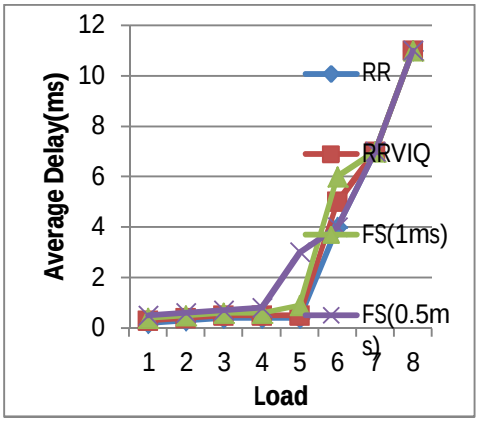

Fig. 6 – Average Delay vs. Load (PPS)

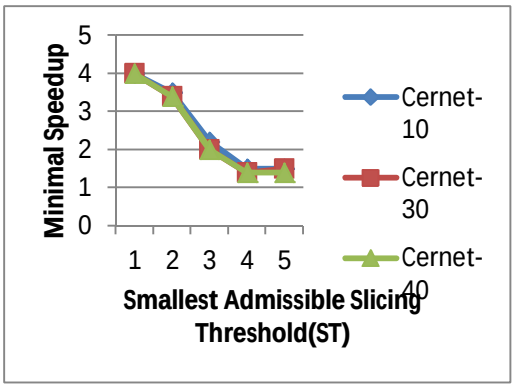

Fig. 7 – Speedup requirements for M^2 Clos

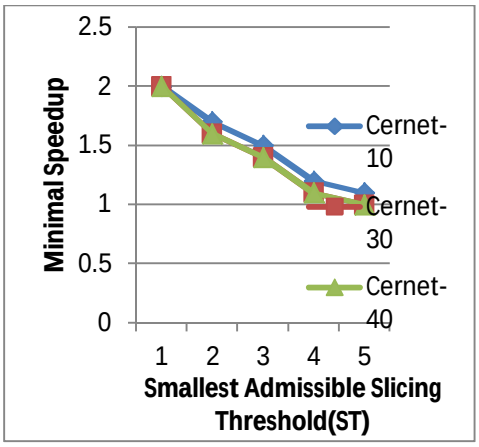

Fig. 8 – Speedup requirements for LBvN

V. CONCLUSIONS

In this paper we study the problem of load balancing multipath switching system. Though many techniques were available to solve this problem, they either cause more average delay or cause packets out of order. It does mean that the packets reach destination but not in correct flow. To let the packets in order, many techniques came into existence. They lack in scalability or they are expensive solutions. Some solutions are not suitable for MPS. To overcome this problem, this paper implements "Flow Slice", a novel scheme, for ensuring the packets are in the same flows while achieving load balancing. We built a prototype application which is a custom simulator that demonstrates the proof of concept. The empirical results are encouraging.

REFERENCES

[1] C.S. Chang, D.S. Lee, and Y.S. Jou, "Load Balanced Birkhoff-von Neumann Switch, Part II: Multi-Stage Buffering," Computer Comm., vol. 25, pp. 623-634, 2002.

[2] C.S. Chang, D.S. Lee, and Y.S. Jou, "Load Balanced Birkhoff-von Neumann Switches, Part I: One-Stage Buffering," Computer Comm., vol. 25, pp. 611-622, 2002.

[3] Vitesse Intelligent Switch Fabrics, http://www.vitesse.com, 2011.

[4] J.S. Turner, "Resilient Cell Resequencing in Terabit Routers," Technical Report WUCS-03-48, June 2003.

[5] J.S. Turner, "Resequencing Cells in an ATM Switch," Technical Report WUCS-91-21, Feb. 1991.

[6] M. Henrion, "Resequencing System for a Switching Node," US Patent, 5,127,000, June 1992.

[7] D.A. Khotimsky and S. Krishnan, "Evaluation of Open-Loop Sequence Control Schemes for Multi-Path Switches," Proc. IEEE Int'l Conf. Comm. (ICC), pp. 2116-2120, 2002.

[8] D.A. Khotimsky, "A Packet Resequencing Protocol for Fault-Tolerant Multipath Transmission with Non-Uniform Traffic Splitting," Proc. IEEE Conf. Global Comm. (GLOBECOM), pp. 1283-1289, 1999.

[9] F.M. Chiussi, D.A. Khotimsky, and S. Krishnan, "Generalized Inverse Multiplexing of Switched ATM Connections," Proc. IEEE Conf. Global Comm. (GLOBECOM), pp. 3134-3140, 1998.

[10] D.G. Thaler and C.V. Ravishankar, "Using Name-Based Mappings to Increase Hit Rates," IEEE/ACM Trans. Networking, vol. 6, no. 1, pp. 1-14, Feb. 1998.

[11] W. Shi and M.H. MacGregor, "Load Balancing for Parallel Forwarding," IEEE/ACM Trans. Networking, vol. 13, no. 4, pp. 790-801, Aug. 2005.

[12] L. Kencl and J.-Y.L. Boudec, "Adaptive Load Sharing for Network Processors," Proc. IEEE INFOCOM, pp. 545-554, 2002.

[13] G. Dittmann and A. Herkersdorf, "Network Processor Load Balancing for High-Speed Links," Proc. Int'l Symp. Performance Evaluation of Computer and Telecomm. Systems (SPECTS), 2002.

[14] Lei Shi, Bin Liu, Changhua Sun, Zhengyu Yin, Laxmi N. Bhuyan, "Load-Balancing Multipath Switching System with Flow Slice", IEEE, March 2012, p1-16.

[15] I. Keslassy and N. Mckeown, "Maintaining Packet Order in Two- Stage Switches," Proc. IEEE INFOCOM, pp. 1032-1041, 2002.

[16] S. Iyer and N. McKeown, "Analysis of the Parallel Packet Switch Architecture," IEEE/ACM Trans. Networking, vol. 11, no. 2, pp. 314-324, Apr. 2003.

[17] A. Aslam and K. Christensen, "Parallel Packet Switching Using Multiplexors with Virtual Input Queues," Proc. Ann. IEEE Conf. Local Computer Networks (LCN), pp. 270-277, 2002.

[18] C.S. Chang, D.S. Lee, and C.Y. Yue, "Providing Guaranteed Rate Services in the Load Balanced Birkhoff-von Neumann Switches," IEEE/ACM Trans. Networking, vol. 14, no. 3, pp. 644-656, June 2006.

[19] Z. Cao, Z. Wang, and E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing," Proc. IEEE INFOCOM, pp. 332-341, 2000.

[20] S. Sinha, S. Kandula, and D. Katabi, "Harnessing TCP's Burstiness with Flowlet Switching," Proc. ACM SIGCOMM Workshop Hot Topics in Networks (HotNets), 2004.

[21] W. Shi and L. Kencl, "Sequence-Preserving Adaptive Load Balancers," Proc. ACM/IEEE Symp. Architecture for Networking and Comm. Systems (ANCS), 2006.