www.ijifr.com | Volume 5 | Issue 11 | July 2018

International Journal of Informative & Futuristic Research | ISSN: 2347-1697

Load-Balancing Multipath Switching System With Flow Slice

Paper ID	IJIFR/V5/ E11/ 010	Page No.	9356-9361	Subject Area	Networking
Key Words	Load Balancing, Traffic Measurement, Switching Theory, Hash Algorithm				

Alok Rajendra Khandare	Student, Department of Computer Engineering, MCOE & RC, Maharashtra

Abstract

Multipath Switching systems (MPS) are intensely used in state-of-the-art core routers to provide terabit or even petabit switching capacity. One of the most intractable issues in designing MPS is how to load balance traffic across its multiple paths while not disturbing the intraflow packet orders. Previous packet-based solutions either suffer from delay penalties or lead to hardware complexity, hence do not scale. Flow-based hashing algorithms also perform badly due to the heavy-tailed flow-size distribution. In this paper, we develop a novel scheme, namely, Flow Slice (FS) that cuts off each flow into flow slices at every intraflow interval larger than a slicing threshold and balances the load on a finer granularity. Based on the studies of tens of real Internet traces, we show that setting a slicing threshold of 1-4 ms, the FS scheme achieves comparative load-balancing performance to the optimal one. It also limits the probability of out-of-order packets to a negligible level (10−6) on three popular MPSes at the cost of little hardware complexity and an internal speedup up to two. These results are proven by theoretical analyses and also validated through trace-driven prototype simulations.

I. INTRODUCTION

Multipath Switching systems (MPS) play a pivotal role in fabricating stateof-the-art high performance core routers. A well-known paradigm is the deployment of Benes multistage switches in Cisco. As a rule of thumb, packet-based solutions are advocated where traffic is dispatched packet by packet to optimally balance the load. However, packets in the same flow may be forwarded in separate paths and experience different delays, thus violating the intraflow packet ordering requirement. A straightforward solution is to use an explicit re-sequencer at each output to restore packet orders. They delay each packet at output until the system delay upper bound is reached. Each packet is time shifted by the

ISSN: 2347-1697
International Journal of Informative & Futuristic Research (IJIFR)
Continuous 59th Edition, Volume - 5, Issue -11, July 2018
Page No. : 9356-9361

same offset before departing, thus preserving the arrival order. Another kind of re-sequencers records each packets sequence number in the flow (defined by input, output port, and priority class), instead of absolute timestamp. By allowing only in-order packets with expected sequence number to depart, they preserve packet orders without penalizing packet delays, but at the cost of maintaining at least N re-sequencers (N is the number of input/output port in a square MPS) at each output, leading to hardware complexity. To avoid the packet out-of-order, another choice is to use flow-based loadbalancing algorithms They dispatch packets in the same flow to a fixed switching path by hashing its 5-tuple to path ID. However, hashing solutions intrinsically suffer from load-imbalance under every timescale. In summary, previous solutions cannot gracefully deal with the load-balancing problem in MPS to meet the three objectives outlined above. they develop a new scheme called Flow Slice (FS) that achieves our load balancing goals perfectly. Based on the observations on tens of broadly located Internet traces, they find that the intraflow packet intervals are often, say in 40-50 percent, larger than the delay upper bound at MPS which can be calculated statistically.

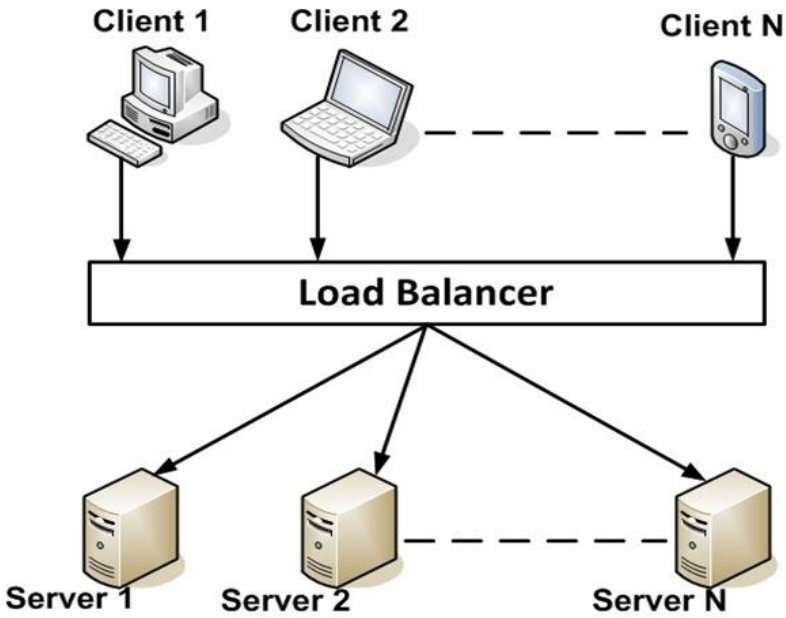

Figure 1: Network Load Balancing

Load balancers have a decisive role in every enterprise network as they serve often as an entry point and have major impact on the performance and the availability of the network. They distribute the incoming server workload to an array of replicated servers in order to serve more clients with a minimum of latency and a maximum of throughput. This operation requires a rewrite of the destination NAT in the IP header. Incoming requests on one public IP are spread to a private subnet of servers and vice versa. Often applied load balancing strategies include policies such as round robin, random, load-based or connection-based balancing.

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-

ISSN: 2347-1697
International Journal of Informative & Futuristic Research (IJIFR)
Continuous 59th Edition, Volume - 5, Issue -11, July 2018
Page No. : 9356-9361

based solutions came into existence. 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. General network load balancing shown in figure 1.

II. SURVEY REVIEW

1. Turner proposed a basic timestamp-based resequencer to deal with the cell out-of-order issue, when cells within a virtual circuit are allowed to take different paths in ATM switching systems. In each scheduling, the resequencer implements a smart contention resolution mechanism to select the oldest cell and then compares its age with a predefined threshold equal to the system delay upper bound. If the selected cell exceeds this age threshold, it will be sent without disturbing cell orders since all previously arrived cells have left the system.
2. Henrion improved the timestamp-based resequencer by introducing time-wheel-like hardware which does not need to compare cells timestamps at output. It maintains an array of D pointer lists, where D is larger than delay upper bound at system measured by timeslots. Each pointer at the list stores location of a cell waiting for resequencing. At timeslot t, the pointer list at slot t modulo D is linked to the tail of the output cell list and removed from the array. After that, pointers of arrival cells at timeslot t are linked together and stored in this empty slot (t modulo D) of the array. This approach delays every cell by fixed D timeslots with O(1) complexity and strictly guarantees cell orders.
3. Iyer and McKeown introduced distributed scheduling that does not exactly emulate OQ, hence also requires resequencing mechanisms at the output.
4. C.S. Chang, D.S. Lee, and Y.S. Jou proposed a jitter control mechanism is inserted before the second stage of LBvN to equalize delays experienced in its first stage and preserve cell order.
5. Keslassy and Mckeown use three-dimensional queues (3DQs) between LBvNs two stages. It arranges buffer by external input/output port and also by internal input port, with totally FIFOs.
6. S. Sinha, S. Kandula, and D. Katabi proposed that TCPs burstiness can be utilized to improve load balancing. Based on this observation, flowlet switching and adaptive burst shifting.

III. STATEWISE WORKING

It shows the state of the system with input to every state and relative output of that state. The figure below shows the state diagram of proposed system. At the very first step the administrator gets login into system successfully then after the network traffic measured according to the packets arrival amount, its size and intervals also. This is shown in first state the output of this module will be administrator username and password for system and packet parameters. Same input will be given to the next state as packet based module which examines the packet parameters. After providing correct input to this state the interval time is going to calculated and it will forwarded to the switching system module

ISSN: 2347-1697
International Journal of Informative & Futuristic Research (IJIFR)
Continuous 59th Edition, Volume - 5, Issue -11, July 2018
Page No. : 9356-9361

which is responsible for deciding the ow path for the packets. Then after the next session is to be for take decision on the packet slicing that is normal slicing or slicing according to the cut-o. If the packet is having small interval then it will be get sliced according to the packet interval for minimum transmission delay and simple forwarding path. Whereas if the packets interval time is larger than slicing threshold then those packets were get sliced according to the equal slicing threshold which cut-o each packet ow and allows minimum hardware complexity and low cost.

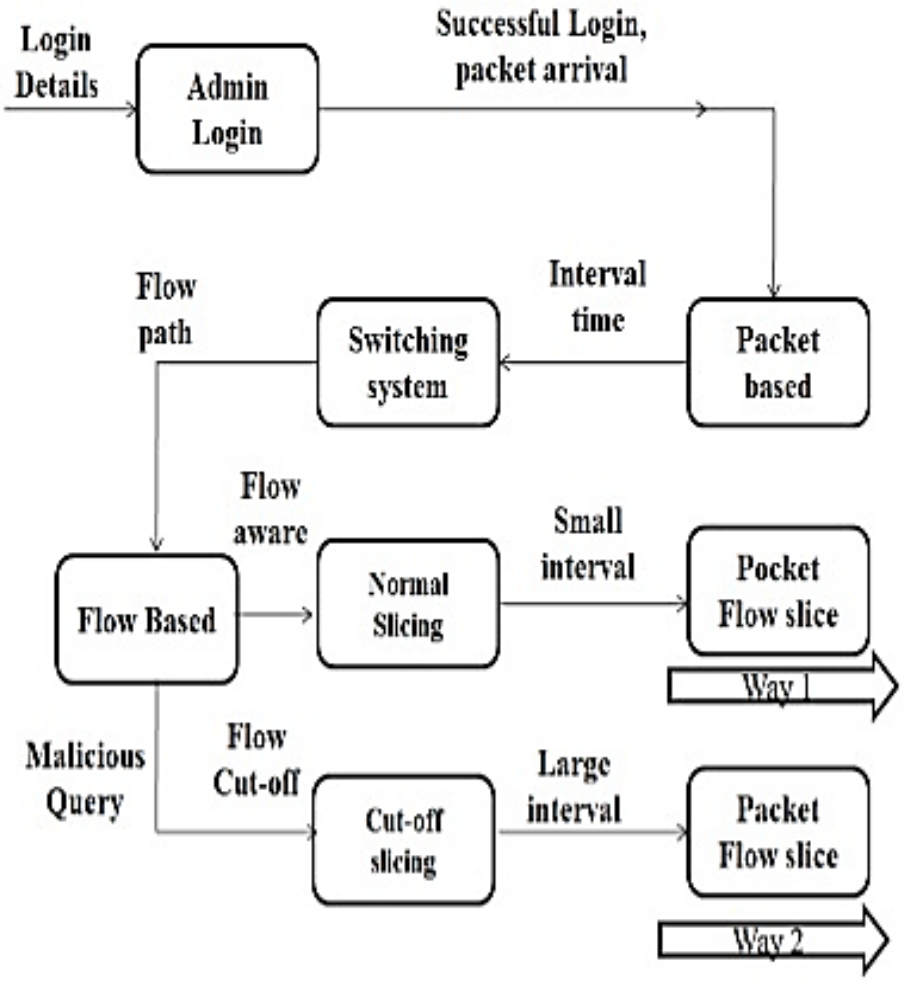

Figure 2: State Machine

IV. PERFORMANCE ANALYSIS

FS under Uniform Trac Pattern depicts the average packet delay in PPS under a uniform traffic pattern. As expected, Round Robin exhibits the smallest delay, as it optimally balances the load. Packet Flow Slice under Unbalanced Traffic Pattern Results under unbalanced traffic is similar to the case under uniform traffic. Packet Flow Slice under Overloaded Traffic are also test the case in which load rate is set to 2.0 under a uniform trac pattern.

ISSN: 2347-1697
International Journal of Informative & Futuristic Research (IJIFR)
Continuous 59th Edition, Volume - 5, Issue -11, July 2018
Page No. : 9356-9361

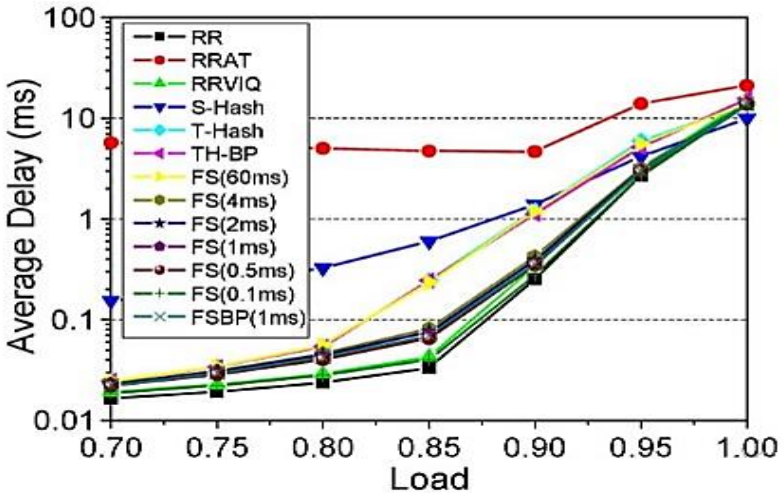

Figure 3: Average packet delay

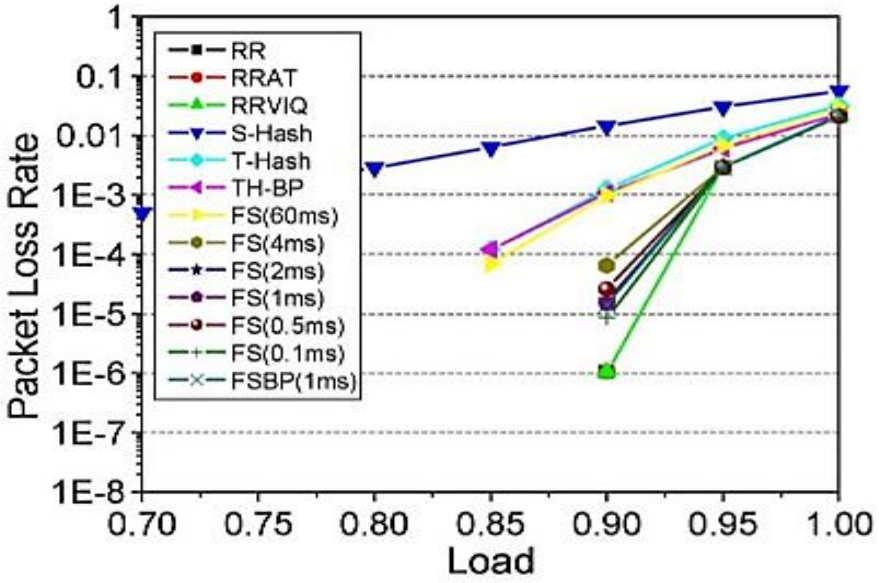

Figure 4: Average packet loss

Results plotted in Figure 3 and 4 show that both packet delay and loss of all the algorithms are similar as this time all the buers are always full, leading to the largest delay and a 50 percent loss rate.

V. CONCLUSION

The implemented system is a load-balancing scheme, i.e. Flow Slice, based on the fact that the intraow packet interval is often, say in 40-50 percent, larger than the delay upper bound at MPS. Due to three positive properties of ow slice, achieves good load-balancing uniformity with little hardware overhead and O(1) timing complexity. By calculating delay bounds at three popular MPSes, it show that when the slicing threshold is set to the smallest admissible value at 1-4 ms, the FS scheme can achieve optimal performance

while keeping the intraow packet out-of-order probability negligible, given an internal speedup upto two.

VI. REFERENCES

[1] Lei Shi, Bin Liu, Changhua Sun, Zhengyu Yin, "Load-Balancing Multipath Switching System with Flow Slice", IEEE on Computers, VOL. 61, NO. 3, MARCH 2012.

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

[3] G. Dittmann and A. Herkersdorf, "Network Processor Load Balancing for High-Speed Links," Proc. Intl Symp. Performance.

[4] 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.

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

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

[7] W. Shi, M.H. MacGregor, and P. Gburzynski, "A Scalable Load Balancer for Forwarding Internet Traffic: Exploiting Flow-Level Burstiness," Proc. Symp. Architecture for Networking and Comm. Systems (ANCS), 2005.

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

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

[10] 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.

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

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

[13] 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. 623634, 2002.

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

[15] 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.

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

[17] 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.