Flow-Slice: A Novel Load-Balancing Scheme for Multi-Path Switching Systems

Lei Shi[1], Bin Liu[1], Changhua Sun[1], Zhengyu Yin[1], Laxmi Bhuyan[2], H. Jonathan Chao[3]

[1]Department of Computer Science and Technology
Tsinghua University
Beijing, China
{shijim,sch04,yzy04}@mails.thu.edu.cn, liub@tsinghua.edu.cn

[2]Department of Computer Science and Engineering
University of California, Riverside
CA 92521, U.S.A
bhuyan@cs.ucr.edu

[3]Department of Electrical and Computer Engineering
Polytechnic University
Brooklyn, NY 11201, U.S.A
chao@poly.edu

ABSTRACT
Multi-Path Switching systems (MPS) are intensively used in the state-of-the-art core routers. One of the most intractable issues is how to load-balance traffic across its multiple paths while not disturbing the intra-flow packet orders. In this paper, based on the studies of tens of real Internet traces, we develop a novel scheme, namely Flow-Slice (FS), which cuts off each flow into flow-slices at every intra-flow interval larger than a slicing threshold set to 1ms~4ms and balances the load on the finer granularity. Through theoretical analyses and comprehensive trace-driven simulations, we show that FS achieves impressive load-balancing performance with little hardware cost while limiting the packet out-of-order chances to a negligible level (below 10^{-6}).

Categories and Subject Descriptors
C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *Packet-switching networks*

General Terms
Algorithms, Measurement, Performance

Keywords
Flow-Slice, Load-Balancing, Multi-Path Switching

1. INTRODUCTION
One of the major issues in designing MPS is the load-balancing problem defined as how to distribute incoming traffic $A(t)$ across its k switching paths to meet the three objectives simultaneously:

- *Uniform load-sharing*: The traffic destined for each output should be dispatched to all the switching paths uniformly;
- *Intra-flow packet ordering*: The packets in the same flow should depart MPS as their arrival orders;
- *Low complexity*: The load-balancing and the additional re-sequencing mechanisms should work fast enough to catch up with the switch fabric's line rate.

ANCS'07, December 3–4, 2007, Orlando, Florida, USA.
ACM 978-1-59593-945-6/07/0012.

The rule-of-thumb on this problem advocates packet-based solutions where the traffic is optimally balanced. However, in this way, packets in the same flow may be forwarded in the separate paths and experience various delays, thus violating the intra-flow packet ordering requirement. Although timestamp or sequence based re-sequencers can be added to restore packet orders, they are often shown to be costly and not scalable. By timestamp based re-sequencer [1], each packet is stalled statically (or adaptively) by the system delay upper bound, which will impose a huge delay penalty. On the other hand, the sequence based re-sequencer [2] will need to maintain at least N re-sequencers at each output, leading to $O(N^2)$ complexity. (N is the number of ports in a square MPS.) In a 1024-port/16-plane/8-priority-class 3-stage-Clos based MPS, it should maintain 4M re-sequencing FIFOs at each output.

To avoid the packet out-of-order, another choice is to use flow-based load-balancing algorithms [3]. They dispatch packets in the same flow to a fixed switching path by hashing its 5-tuple to path ID. However, hashing solution will lead to severe load-imbalance. It is further shown by our evaluation results.

In this paper, we present a new scheme, namely *Flow-Slice* (FS), that perfectly achieves the three objectives defined above. Our idea is inspired from the observations on tens of broadly located Internet traces that the intra-flow packet intervals are often, say in 40%~50% percentages, larger than the delay upper bound at MPS which is calculated statistically. As such, if we cut off each flow at every packet interval larger than a slicing threshold equaling to this bound and balance the load on the generated flow-slices, the three objectives are met triply.

- The load-balancing uniformity of FS is only moderately degraded from the optimal load-balancing;
- The intra-flow packet order is kept intact as their arrivals. Exceptions only happen in a negligible level. (below 10^{-6});
- The flow-slice table size to implement FS is limited below 1.8MB under 40Gbps line rate, which can be placed on-chip to provide an ultra-fast access speed.

2. FLOW-SLICE
Definition: A flow-slice is a sequence of packets in a flow, where every intra-flow interval between two consecutive packets is smaller than or equal to a slicing threshold *ST*.

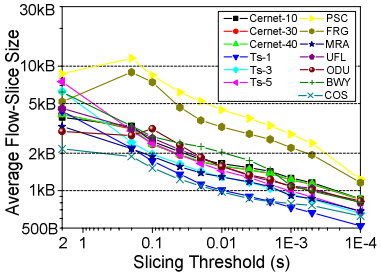

Figure 1. Flow-slice size.

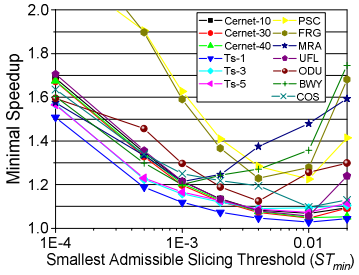

Figure 2. Speedup requirements in PPS.

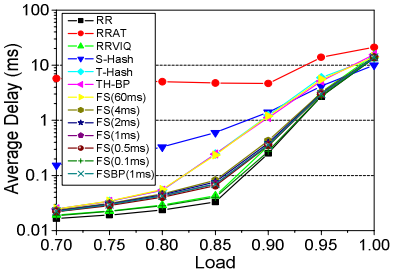

Figure 3. Average packet delay in PPS.

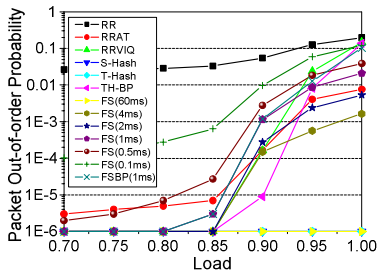

Figure 4. Packet out-of-order probability in PPS.

Flow-slices can be seen as mini-flows created by cutting off every intra-flow interval larger than *ST*. Compared with the original 5-tuple flows, three specific properties are observed for flow-slice in all the traces we study.

Property 1 (*Small Size*): Both the average per-flow-slice packet count and the average per-flow-slice size are much smaller than those of the 5-tuple flows.

Figure 1 shows the average per-flow-slice size, while the per-flow sizes shown by the intersections of the curves and the Y axis are much larger. Using the per-flow-slice (per-flow) size to indicate the load-balancing granularity, the flow-based algorithm is 3.5~12 times coarser than the packet-based one, while the flow-slice based algorithm is only 41%~97% coarser at *ST*=1ms.

Property 2 (*Light-Tailed Size Distribution*): The per-flow-slice packet count and size distributions are light-tailed while the per-flow distributions are heavy-tailed.

Property 3 (*Fewer Active Flow-Slices*): The active flow-slice number is 1~2 magnitudes fewer than the active 5-tuple flow.

3. ADMISSIBLE SLICING THRESHOLD

Theorem (*Packet Out-of-order Probability*): Setting a slicing threshold *ST* for MPS adopting FS, which leads to a statistical delay upper bound of $D_{1-\delta}$ in $1-\delta$ confidence interval, the packet out-of-order probability in MPS will be guaranteed of no more than δ, if only it suffices $ST \geq D_{1-\delta}$.

The slicing threshold *ST* is defined to be *admissible* if it guarantees a packet out-of-order probability of no more than 10^{-6}. We are most interested in the *smallest admissible slicing threshold* (ST_{min}), as it provides the best load-balancing performance while satisfying the packet out-of-order requirement. We calculate the ST_{min} for three popular MPS designs, including Parallel Packet Switch (PPS), Load-Balanced Birkhoff-von Neumann switch (LBvN) and Multi-plane Multi-stage Clos network based switch (M^2Clos).

Consider a PPS with port number (*N*) below 32 and *R*/*k*=5Gbps, where *R* denotes the external line rate and *k* denotes the switch plane number, the ST_{min} is shown in Figure 2, as a inverse function of the provided minimal speedup *S* of the PPS. We observe that a speedup of 1.409 is sufficient to ensure $ST_{min} \leq 2$ms for all traces. Given a slightly larger speedup of 1.627, $ST_{min} \leq 1$ms can be expected. For the typical LBvN and M^2Clos design, the speedup of 2 is required to ensure $ST_{min} \leq 4$ms.

4. EVALUATIONS

We establish prototypes for all the three MPS by software modeling. Specifically, the PPS prototype has 32 external ports working at 40Gbps and 8 parallel switch planes working at 5Gbps. No speedup is provided. We use homogeneous real trace data sets collected at CERNET backbone to generate the traffic at each input. Each segment has an average traffic speed around 3.5Gbps and is condensed to simulate the expected traffic rate. Each incoming packet's information, including the packet arrival time, packet length and 5-tuple, are extracted from the trace files. In each test slot, 1.2 billion packets are injected to the prototype.

Figure 3 depicts the average packet delay experienced in PPS when the traffic arrival is uniform. At the load rate above 0.85, FS with slicing threshold of 1ms receives the average delay only one times larger than the optimal Round-Robin (RR) algorithm; while the hashing algorithms and the re-sequencing methods are generally more than six times larger. Figure 4 depicts the packet out-of-order probability. RR without re-sequencer consistently disorders more than 2% packets, while FS limits the packet out-of-order at a negligible level (below 10^{-6}) if only slicing threshold is no less than 1ms and load rate is no larger than 0.8. This corresponds to a speedup requirement of 1.25.

ACKNOWLEDGMENTS

This work is partially supported by National Science Foundation of China (No. 60573121, 60625201), and National Basic Research Program of China (973 program, No. 2007CB310702).

5. REFERENCES

[1] J. S. Turner, "Resequencing Cells in an ATM Switch," *Tech. Rep.*, WUCS-91-21, Feb. 1991.

[2] D. A. Khotimsky and S. Krishnan, "Evaluation of Open-loop Sequence Control Schemes for Multi-path Switches," in *Proc. IEEE ICC*, pp. 2116-2120, 2002.

[3] L. Shi, W. Li, B. Liu, and X. Wang, "Flow Mapping in the Load Balancing Parallel Packet Switches," in *Proc. IEEE HPSR*, pp. 254-258, 2005.