Modeling Components for Cut-Through Performance Analysis of Network Switches

Marat Zhanikeev*

* School of Economics Informatics, Kanazawa Gakuin University
Sue-machi 10, Kanazawa-shi, Ishikawa 920-1302
Email: maratishe@gmail.com

Abstract— There are several existing models which describe a given aspect of network performance. Leaky bucket is a good model for analysis and visualization of dynamics within a packet queue. Fluid models exist for analysis of larger aggregates of packet traffic. This paper argues that modeling of contention in network equipment has become important in recent years. The original contribution of this paper is in the form of several components with major focus on cut-through which can be used for modeling and analysis of contention in network switches.

Index Terms— traffic contention, performance modeling, packets vs circuits, cut-through, network virtualization

I. Introduction

Today, Big Data transfer – also referred to as bulk transfer or bigdata networking – has to compete with all other traffic because outgoing ports on network switches are mostly shared by a high number of flows. This creates unfair conditions for Big Data because these transfers take a long time and suffer from interference with many other flows. Recent literature [1] shows that is possible to emulate circuits over traditional packets simply by making sure that paths (including end-to-end) are used exclusively and therefore can benefit from the cut-through mode of packet switching.

Intra- and Inter-DC networking also suffers from network congestion. More and more services spread their resources across multiple DCs in order to run a distributed network of virtual servers. In these conditions, long-distance bulk traffic is increasing but is also limited in practically achievable end-to-end (e2e) throughput [2].

In clouds, migration of virtual machines as well as resources in general can be triggered by the cloud provider while pursuing a global optimizational objective. However, due to limitations in achievable throughput there is literature which tries to minimize the number of network transfers [3]. DC networking in general is conducted at least at three distinct layers:

- inside racks or small hardware clusters;
- intrAnets inside DCs;
- intErnets across DCs, also commonly referred to as interconnect.

The cut-through mode of switching is available to networking at all these layers [10]. There are recent attempts to enforce the cut-through mode end-to-end, literally from a unit of in-rack equipment to another such unit in another cloud location [4]. In terms of modes, the cut-through mode is followed by store-and-forward and further by various queuing disciplines, in the order of decreasing achievable throughput.

There are several predecessors to this paper. The idea of emulated circuits was first introduced in [1], and the same general idea was placed in the context of bigdata transfer in [5]. The Tall Gate model for distributed coordination of circuit management was proposed in [6]. A holistic technology with all the above elements put together plus using hotspots for realistic traffic modeling can be found in [7]. Network designs for circuits, specifically the e2e cut-through management technology is in [4].

Fig.1 shows the overall setting of analysis in this paper. The new generation of Network Operation Centers (NOCs) [8] will manage both packet- and circuit-dedicated lines. Circuits are not physical but are emulated – this is referred to as circuits-over-packets emulation in [1]. The modeling methods proposed in this paper are intended as a ready-made toolkit for performance analysis whose results should help the NOC optimize the use of switching equipment, given that some of its ports are dedicated to cut-through traffic.

II. The Basic Model of Contention

As shown in Fig.2, for each packet processed by a switch (left to right in the figure), the following states of the per-packet processing overhead (cost) can be experienced.

The C state is the cut through mode [10] – the core focus in this paper. In case of C, only the 6-8 headmost bytes

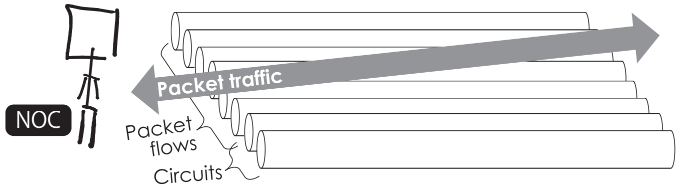

Figure 1: A future NOC that manages a mixed pool of packet and circuit lines.

978-3-903176-31-7 © 2020 IFIP

of the packet are received before making the forwarding decision and sending the packet.

If, while a packet is being processed, another packet arrives at the same outgoing port, then at least SF state is inevitable. In which case the packet is fully received before making further decisions. If the outgoing port becomes open for transmission in the meantime, the packet is sent to its next hop – this case is referred to as store-and-forward (SF), or simply S further in this paper. During heavy contention, even the SF state is rare, as the packet has to be placed in the queue and wait for its turn to be transmitted. While in queue, the packet is subjected to the policies dictated by QoS classes, if those are enforced on the switch [9].

In context of bulk transfer (bigdata networking, etc.), the obvious objective for the NOC is to maximize end-to-end (e2e) throughput, that is, create and maintain e2e paths in which each hop delegates the packet in the C mode. This can be referred to as e2e circuits because the entire e2e paths is effectively contention free and used by only one flow. The cut-through mode is actively discussed in clouds – data center networking and interworking – with the same goal of maximizing throughput for large bulk transfers [10]. Cut-through technology is also considered in optical networks [11].

Note that Software-Defined Network (SDN), Network Function Virtualization (NFV) and other kinds of network virtualization technology are represented by at least the SF or, in many cases, the SQF processing path in Fig.2. More details on the subject is provided in the next section when reviewing the related literature.

In practice, the subject of contention (as a larger framework of the cut-through goal) can be treated in two distinct ways: traditional versus circuit-oriented. Note that, from the viewpoint of the old circuits vs packets argument, the packet model is more efficient on average. However, if only bulk transfer flows are isolated, then the circuit model is better. An attempt to resolve this problem is made in [8] by using a hybrid NOC that has separate pools for packet and circuit switching and optimizes the global networking resource between the two. Repeating an earlier statement, the modeling components proposed in this paper are supposed to help the decision-making process in such a NOC.

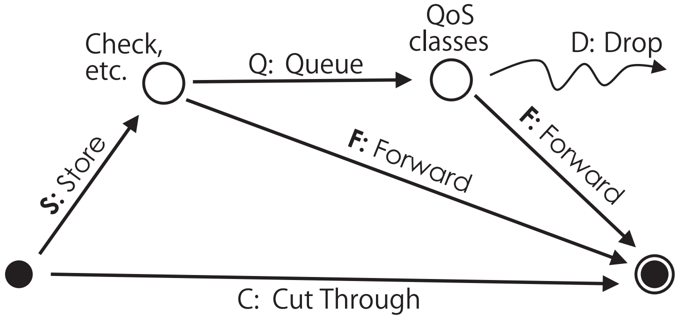

Figure 2: A modeling the states/kinds of overhead spent on every packet in any existing packet switch.

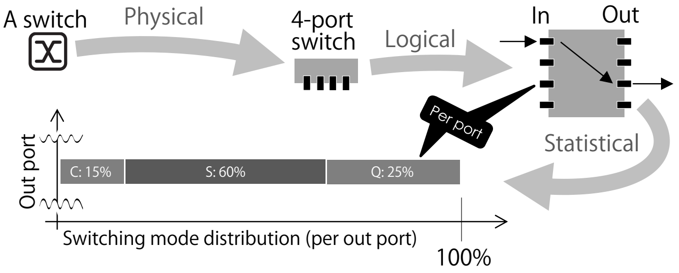

Figure 3: The proposed modeling method and all its main components. The diagram at the bottom is referred to as the CSQ diagram in this paper.

III. The Proposed Modeling Method

C, S, Q, from this point on, are simplifications of the cut-through, store-and-forward, and queuing switching modes. The queuing itself is not considered but Q is considered to be everything that exceeds the S mode. Literally, if a packet is classified as Q when, at its time its first byte is received by the switch, the previously received packet is at least in the store-and-forward mode prior to start of transmission (of its first byte). For example, if the transmission of the previous received packet has already started, the newly arriving packet is itself classified as S. In other words, when transmission of one packet overlaps with one other, the situation is classified as S. When two or more transmissions attempt to grab the same outgoing port, then the situation is classified as Q. Note that this is a useful simplification as modeling and performance analysis the various queueing disciplines (and QoS classes) is a complicated task [9] which would unnecessarily burden the proposal.

The proposed method also focuses on a single switch. In fact, the unit of analysis is an individual outgoing port (out port is also used for short) which has its own CSQ statistic. CSQ statistic is defined as relative ratios of the counts of packets in the respective switching modes.

Modeling transformations are shown as the left-to-right circle in Fig.3. The following stages of modeling abstraction are proposed.

Physical stage is when a switch is substantiated, particularly by specifying the number of ports. Logical stage is when the ports are separated into in and out ports. Under the duplex mode (which is common in network switches today), any port can be in and out, concurrently. This is why the logical representation of the switch has double number of ports. On the surface, this is confusing. However, when considering that switching configurations allow any combination of in and out ports (granted, loops within the same port are impractical), such a

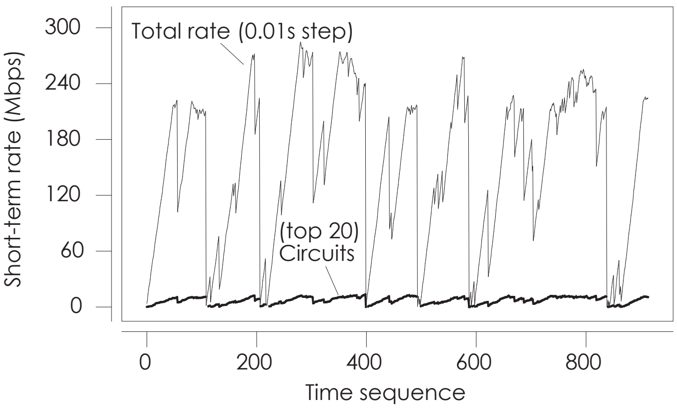

Figure 4: A snippet of a packet trace replayed with and without separating circuit flows from the rest of the traffic.

representation becomes convenient as it allows to connect any in port to any out port, as is shown in Fig.3.

Statistical stage is where the results of the analysis of the logical representation is visualized using the CSQ statistic (represented as packet counts for each respective switching mode). As the example diagram in Fig.3 shows, the statistic can be plotted as bars representing relative percentage for each switching mode. This paper proposes modeling/measurement/simulation components for the various stages in Fig.3. Specifically, the CSQ diagram is used to visualize results from a wide range of conditions applied to a switch under real traffic (packet trace replay).

IV. Trace-Based Analysis : Traces

The problem with statistically modeled traffic is that it has flatter dimensionality than real traffic. Most models, in fact, would focus on only one metric, normally flow size (in bytes or packets) or aggregate packet count. Packet trace replay is a good alternative to statistical modeling. Several recent traces from [12] were randomly selected for replay. Several random 10s cuts of the larger trace were replayed in accordance with the timestamp and size for individual packets. In total, 100 intervals 10s each were used. Separately, the replay process was controlled using two parameters – rate and topn, which will be introduced further in this paper.

Fig.4 shows one entire trace (900s) randomly selected from the above dataset. The short-term rate is calculated using 0.01s window and the same size step. Note that CSQ analysis further in this paper will focus on even shorter term by, literally, analyzing a moving window of 3 packets (as was explained above).

In Fig.4, top 20 flows (by byte count) are separated into hypothetical circuits and their aggregate throughput is shown as a thicker line at the bottom. Circuits amount to about 20-30 Mbps (about 10% of total traffic). The figure shows that circuit flows are synchronized with peaks in total throughput which hints at the bursty nature of this traffic trace (common for general traffic as well). The subject of the share in traffic occupied by bulk transfer flows is revisited further in this paper when discussing results.

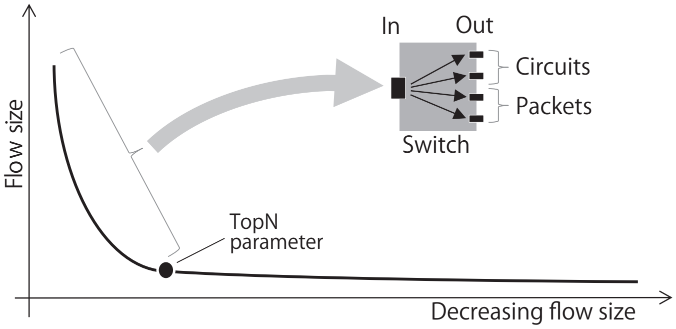

Figure 5: The design of the TopN method.

V. Trace-Based Analysis : The TopN Method

TopN stands for top N largest flows, which is also the topn parameter in trace-based simulation further in this section. The other parameter is the rate, the detailed for which are provided below.

Fig.5 shows the design and the modeling approach of the TopN method. Flows always form a distribution with a very small head (when ordered by decreasing flow size, etc.). The topn parameter therefore corresponds to the size of the head of the distribution (or tail in the long tail viewpoint).

The replay method for packet traces is original and focuses on the contention conditions. We do not care about in ports, which is why Fig.5 shows that all the traffic (several traces, etc.) are mixed and arrive through a single large incoming port. Inside the switch, traffic is separated in a round-robin manner, that is, the first empty out port is picked for each newly arrived packet. Circuit- and packet-able out ports are handled separately. The special logic is as follows. Selection of out ports for packet traffic is on the per-packet basic. For circuits ports, once a circuit port is captured by a flow, it cannot be recaptured by another flow until the first flow has completed its transmission.

The TopN method has two distinct uses:

- analysis of contention performance of a switch under a real trace with circuit flows removed from the trace;
- optimizing switch utilization – specifically the share of dedicated circuit ports and/or time slots by NOC during operation.

The first purpose is perfect for offline analysis of performance using previously captured packet traces. This case also allows for a certain degree of simplification. For example, the removed TopN flows do not have to be simulated (replayed). One can just assume that they have captured and held a circuit out port for the duration

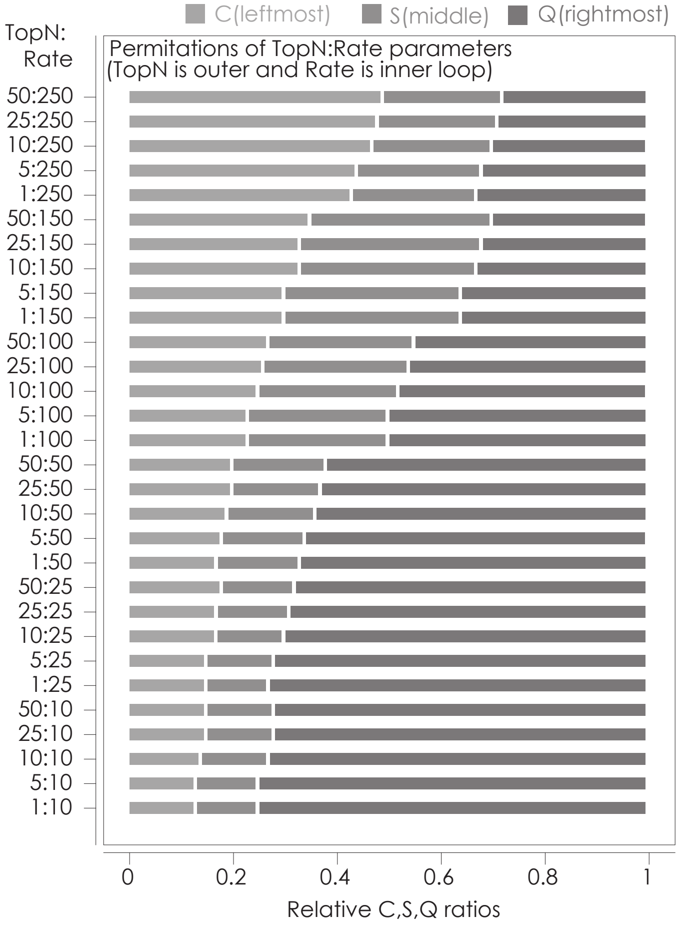

Figure 6: An example CSQ diagram generated for all permutations of rate and topn parameters in the above trace-based analysis.

of its lifespan. The rest of the traffic is replayed on other packet-specific out ports.

For analysis, the replay process needs the rate parameter. This is a means for speeding up (or slowing down) a trace and seeing how it affects the CSQ statistic. Note that the effect is not proportional to the parameter, since the gap between packets in non-uniform. This section will use several rates from 10Mbps up to 250Mbps. Having packet size from the trace, it is possible to calculate the new packet gap and replay the packet in accordance to its scaled timing.

Fig.6 is a CSQ diagram created from replaying the actual traces from the WIDE [12] repository. All the 100 cuts were used, as was explained earlier in this paper. Each value (C, S, Q on each bar) is an average from all the traces, thus, representing the average performance for the entire trace.

The vertical axis of Fig.6 shows permutations of the two main parameters, first TopN (outer loop of permutations) and then rate. The horizontal scale shows the relative ratios of CSQ statistic for each respective setting.

The following reading of Fig.6 can be offered. With highly conjected traffic (rates of 10-50Mbps), C ratio is low. Both C and S are roughly the same size and together form about 20-30% of total volume. At the rate of 100Mbps, one can see a drastic increase in S and less in C ratios. Physically, this means that at this rate there is often sufficient gap between packets for the cut-through or store-and-forward (no queueing) mode, larger shared given to the S mode. The TopN effect is there but it is relatively small compared to the effect of the rate parameter. The same trend is found for the rate of 150Mbps.

For for rate of 250Mbps, C takes the largest share of traffic. Physically, this means that there is often sufficient gaps between packets to avoid contention altogether.

The following conclusion can be drawn from this analysis. First, the obvious (and intuitive) conclusion is that the network that created that trace can upgrade its capacity to 250Mbps to drastically improve the share of the cut-through traffic. The non-obvious conclusion is that the TopN parameter has much smaller effect than could be expected. However, this is strictly due to the nature of the trace, which was confirmed to have relatively few flows with traffic volume that can be considered as bulk transfer. In other words, this trace does not contain DC-like traffic which should have a larger share of flows with considerable bulk.

VI. Conclusion

This paper proposed a new methodology for analyzing performance of network switches strictly from the viewpoint of contention. The concept of contention in this paper is viewed as distinct from congestion. While the latter includes queueing delay, contention is a low-level representation of switching modes. Specifically, this paper clearly distinguishes the cut-through and store-and-forward modes – these are well-defined and can be calculated precisely having access to packet size and nominal transmission speed. The unique viewpoint of this paper is that all the processing delays above store-and-forward are viewed as excessively bad performance.

Analysis using the TopN method offerred a very simple and easy-to-control method for visualizing the ratio of the three separate traffic types – naturally focusing on the cut-through – depending on network conditions.

For a Network Operating Center dealing with contention issues, the modeling methods proposed in this paper will both help estimate the degree of problems with content but also can help optimize utilization of network switches in realtime. Note that optimal here includes the effect from enforcing the cut-through mode for selected bulk transfer flows. When considering that cut-through sessions can cut transfer time by several times, the effect from such optimization can offer financial benefits for providers of cloud infrastructure.

References

[1] M.Zhanikeev, “Can We Emulate Local Circuit Switching in Cloud Storage?”, IEICE Technical Report on Network Systems (NS), vol.114, no.107, pp.1–4, June 2014.

[2] M.Zhanikeev, “Performance Management of Cloud Populations via Cloud Probing”, IPSJ Journal of Information Processing, vol.24, no.1, pp.99–108, January 2016.

[3] M.Zhanikeev, “Optimizing Virtual Machine Migration for Energy-Efficient Clouds”, IEICE Transactions on Communications, vol.E97-B, no.2, pp.450–458, February 2014.

[4] M.Zhanikeev, “Cut-Through Network Designs for High-Throughput E2E Networking”, IPSJ Technical Meeting on High Performance Computing (HPC), vol.148(36), pp.1–4, March 2015.

[5] M.Zhanikeev, “Circuit Emulation for Bulk Transfers in Distributed Storage and Clouds”, 6th RICC Workshop, September 2014.

[6] M.Zhanikeev, “A City Traffic Model for Optical Circuit Switching in Data Centers”, IEICE Technical Report on Optical Circuit Switching (OCS), vol.114, no.281, pp.113–116, October 2014.

[7] M.Zhanikeev, “The All-In-One Package for Massively Multicore, Heterogeneous Jobs with Hotspots, and Data Streaming”, Summer United Workshops on Parallel, Distributed, and Cooperative Processing (SWoPP), vol.2015-ARC-216, no.22, pp.1–6, August 2015.

[8] M.Zhanikeev, “The Next Generation of Networks is all about Hotspot Distributions and Cut-Through Circuits”, IEICE Technical Report on Communication Quality (CQ), vol.115, no.11, pp.1–4, April 2015.

[9] M.Zhanikeev and Y.Tanaka, “Analytical Models for L2 versus L3 QoS Provisioning”, IEICE Technical Report on Photonic Networks, Vol.112, No.276, pp.13–18, November 2012.

[10] “Cut-Through and Store-and-Forward Ethernet Switching for Low-Latency Environments”, Cisco White Paper, 2014.

[11] G.Wang, D.Andersen, M.Kaminsky, K.Papagiannaki, T.Ng, M.Kozuch, M.Ryan, “c-Through: Part-time Optics in Data Centers”, ACM SIGCOMM, pp.327–338, October 2010.

[12] MAWI working group traffic archive.. [Online]. Available: http://tracer.csl.sony.co.jp/mawi/ (Retrieved June 2017)

[13] M.Zhanikeev, “Towards a Practical Method for Interactive Traffic Visualizations in Data Centers”, IEICE Technical Report on Service Computing (SC), vol.114, no.50, pp.1–4, May 2014.

[14] M.Zhanikeev, “Experiences from Measuring Per-Packet Cost of Software Defined Networking”, IEICE Technical Report on Service Computing (SC), vol.113, no.86, pp.31–34, June 2013.