Measuring Queues in Campus Network via Link Tapping

Xiaoqi Chen
Princeton University
xiaoqic@cs.princeton.edu

Hyojoon Kim
Princeton University
joonk@princeton.edu

ABSTRACT

We monitored our local campus network to diagnose a packet drop phenomenon we observed on a lightly utilized link, using a queue monitoring program running on a P4-based programmable switch. The setup uses the programmable switch to tap both ingress and egress ports of a legacy router, which does not have the capability to report granular queuing metrics itself. Our investigation found that the packet drops are potentially caused by an active performance monitoring tool simultaneously scheduling too many tests.

CCS CONCEPTS

• **Networks** → **Network measurement**; Packet scheduling.

KEYWORDS

Queue monitoring, queuing delay, congestion control

1 INTRODUCTION

We observed one campus network router occasionally reporting a large amount of packet drops in a particular egress port even though it was operating at low average link utilization. There are several different causes for such behavior, including microbursts, hardware failure, bug in queue scheduling implementation, and so on. In particular, assuming there is no bug or failure, we suspect the router may indeed run out of queuing buffer briefly, due to a high rate of ingress traffic. However, existing network monitoring tools only provide very coarse-grained information, at seconds or minutes time scale, therefore yielding little insight for network operators to debug transient queue buildups.

The emergence of programmable switch allows us to write algorithms to monitor and analyze queuing directly in the data plane [1, 2], or let hosts collect fine-grained queuing metrics via in-band network telemetry (INT) and optimize congestion control [3]. However, replacing existing routers with programmable switches is operationally infeasible and uneconomical. Instead, we show that it is still possible to harness the power of programmable switches when analyzing the queues in legacy devices. In this paper, we present a novel setup to analyze queues in legacy devices with 10Gbps/100Gbps links, in a non-invasive fashion within an operational campus network, using tapping links and an off-path observation device. We report our experience of using a P4-based programmable switch to analyze queuing anomaly at a router in Princeton University's campus network.

2 BACKGROUND

Princeton University's campus network peers with the Internet as well as two research networks, ESNet and Internet2. The Office of Information Technology (OIT) administers and operates the campus network infrastructure as well as the university's data centers and High Performance Computing (HPC) clusters.

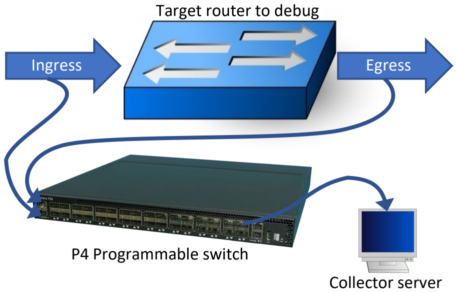


Figure 1: We use tapping and an off-path programmable switch to analyze queue utilization in the target router.

The routers in our campus network use conventional network monitoring and analysis tools, such as SNMP, NetFlow, Syslog, etc. Princeton OIT maintains a centralized monitoring system that gathers counters from all routers, such as bytes count and packet drop count. Princeton uses the Big Monitoring Fabric by BigSwitch Networks as its packet broker system.

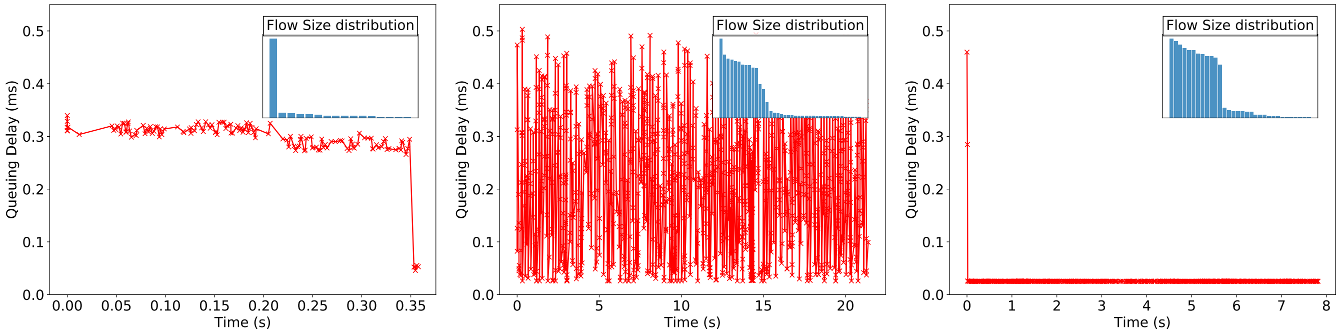
We occasionally observe a large number of packet drops in a border router (the "target router"). The target router has one 100Gbps upstream connection to Internet2 and multiple 10Gbps downstream connection to servers, and the drops are on one 10Gbps downstream link. Monitoring data showed that during the packet drop period, this downstream link has a low average utilization. We suspect that the queue may be experiencing bursty traffic and was overflown, or there could be bugs in the router implementation. However, since the target router can only provide monitoring data at 1-minute granularity, we cannot investigate the phenomenon further.

We have several P4-based programmable switches at hand, yet OIT have not deployed programmable switches to process production network traffic. As these switches can run fine-grained network measurement algorithms at per-packet granularity, we decided to use a programmable switch as an off-path observation device, and use it to analyze queuing in the target router.

3 ANALYZE QUEUING IN LEGACY DEVICES

We now give a high level overview of our tapping-based queuing delay measurement experiment. As illustrated in Figure 1, we used an off-path programmable switch running a queue analysis algorithm [1] to monitor the target legacy network device, which does not provide fine-grained measurement of queuing delay. The programmable switch then sends reports to a collector server whenever high queuing delay is observed.

Tapping legacy switches: To analyze the queuing delay in the target switch, we need to tap both its ingress and egress link. We can implement this using physical layer split-fiber tapping, or use a router's SPAN port. In our setup, we tap both the 100Gbps ingress and the 10Gbps egress port, using physical layer tapping.



(a) Steady state congestion

(b) Bursty

(c) Outlier high-delay packets

Figure 2: Three types of high queuing delay events observed in our monitoring period.

Analyzing queuing delay: After tapping links, we can use an observation device (a two-port smart NIC or a programmable switch) to match two appearances of the same packet, once on the ingress link and once more on the egress link, and precisely compute the time difference between the two. In our setup, we use a Barefoot Tofino Wedge 100-32X programmable switch. We refer interested readers to [1] for the technical details.

Incident reporting: We connect the programmable switch to a collector server using a 10Gbps link, and configure the programmable switch to send postcard-style reports with timestamp, calculated delay, flow ID etc., whenever it observes the queuing delay on the target router exceeds 0.2ms, as typically the delay is much lower when the queuing buffer is empty. Furthermore, we configure the switch to send at least 1000 subsequent packets after observing any high-delay packet.

4 RESULTS

We collected data for approximately 4 weeks (from May 4th to May 30th) and gathered 18 high-delay incidents. Meanwhile, we also examined coarse-grained drop statistics from the existing router monitoring interface.

After analyzing the collected data, we categorized the high-delay incidents into three categories. In Figure 2, We present the queuing delay over time as well as the flow size distribution in three example incidents, with flow defined using five-tuples. We have successfully recovered queuing delay for a subset of egress packets observed; each data point in the figure shows one such packet's delay.

Steady state congestion (4 incidents): In these incidents we observe a consistent, high queuing delay in the queue for a period of time, as shown in Figure 2(a). In these cases, there's usually only one significant flow in the queue, with its congestion control algorithm probing for maximum throughput; the delay observed might corresponds to the Early Congestion Notification (ECN) threshold. Drop statistics show that there are minimal packet dropping in these cases.

Bursty (10 incidents): The queueing delay oscillates wildly during these incidents, as shown in Figure 2(b). Apparently, there are multiple large flows competing for bandwidth. Each flow's congestion control algorithm failed to probe the bottleneck bandwidth together, thus their total sending rate is either too large or too small throughout the period. We also verified that the egress link's peak throughput reaches nearly 10Gbps, although it varies wildly.

Drop statistics show that these incidents caused a large amount of packet drops, while the average throughput is low. These features match the incidents previously observed by the campus network operator. After investigating individual flow IDs, we suspect that these incidents are likely caused by multiple PerfSonar throughput and latency tests being initiated concurrently, from multiple external hosts on Internet2, although PerfSonar scheduler [4] would have scheduled multiple tests in series (not in parallel). As our data were aggregated and sampled, we need to collect more data and carefully analyze to pinpoint the root cause of these bursty, concurrent flows.

Outlier high-delay packets (4 incidents): In Figure 2(c), the link appears underutilized and most packets have very low queuing delay. However, there are a handful of packets (less than 5) suffered from high queuing delay. We suspect the delay experienced by these outlier packets are not caused by a full queuing buffer, and leave them for further investigation.

5 CONCLUSION

We inferred the queue utilization in a campus network router using link tapping and an off-path programmable switch, and categorized our observations into three types of incidents: normal, bursty, and outliers. We observed that a large number of packet drops will occur when there are many active large flows, i.e. those entering Congestion Control and competing for bandwidth at this bottleneck link, and the queue will exhibit bursty pattern.

REFERENCES

[1] Xiaoqi Chen, Shir Landau Feibish, Yaron Koral, Jennifer Rexford, Ori Rottenstreich, Steven A Monetti, and Tzuu-Yi Wang. 2019. Fine-Grained Queue Measurement in the Data Plane. In *Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. ACM.

[2] Raj Joshi, Ting Qu, Mun Choon Chan, Ben Leong, and Boon Thau Loo. 2018. BurstRadar: Practical real-time microburst monitoring for datacenter networks. In *Proceedings of the 9th Asia-Pacific Workshop on Systems (APSys)*. ACM, 8.

[3] Yuliang Li, Rui Miao, Hongqiang Harry Liu, Yan Zhuang, Fei Feng, Lingbo Tang, Zheng Cao, Ming Zhang, Frank Kelly, Mohammad Alizadeh, et al. 2019. HPCC: high precision congestion control. In *Proceedings of the ACM SIGCOMM 2019 Conference*. ACM, 44–58.

[4] perfSONAR Project. 2019. perfSONAR Toolkit 4.2.2 documentation: Test and Tool Reference. https://docs.perfsonar.net/pscheduler_ref_tests_tools.html#test-classifications.