RDMA Network Performance Anomalies Diagnosis with Hawkeye

Xiao Li[1], Shicheng Wang[1], Menghao Zhang[2], Zhiliang Wang[1], Mingwei Xu[1], Jiahai Yang[1]
[1]Tsinghua University [2]Beihang University

CCS CONCEPTS

• **Networks** → **Network monitoring**; *Programmable networks.*

KEYWORDS

Remote Direct Memory Access; Network Performance Anomalies Diagnosis

ACM Reference Format:
Xiao Li, Shicheng Wang, Menghao Zhang, Zhiliang Wang, Mingwei Xu, and Jiahai Yang. 2024. RDMA Network Performance Anomalies Diagnosis with Hawkeye. In *ACM SIGCOMM 2024 Conference (ACM SIGCOMM Posters and Demos '24), August 4–8, 2024, Sydney, NSW, Australia.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3672202.3673711

1 INTRODUCTION

RDMA (Remote Direct Memory Access), recognized for its high throughput and low latency, is increasingly deployed across data centers. In RDMA network, the hop-by-hop flow control, i.e., PFC (Priority-based Flow Control), is usually employed to eliminate packet loss and guarantee high performance. However, even with congestion control [9], PFC still frequently occurs, causing head-of-line blocking and congestion spreading [5], which poses new challenges for effective and efficient diagnosis of network performance anomalies (NPAs). For example, in Figure 1, short bursts from A1-An congest SW4.P1 and cause PFC to spread back along the path of flow F2. Finally, SW1.P1 gets PFC and forms into queue congestion. And F1 gets performance anomaly as the victim flow, although it has no queue contention with other flows.

Conventional wisdom on NPA diagnosis falls short in precisely diagnosing the root causes in RDMA networks efficiently. Firstly, existing diagnosis mechanisms (e.g., SpiderMon [6]) usually attribute NPAs to flow contention in switch queues. They analyze the queue information of victim flows, and identify the major contributor to the flow contention (e.g., bursts). Nevertheless, beyond flow contention, queue congestion in RDMA network can also result from congestion that propagates through PFC from downstream nodes several hops away, where the root cause flow may not share a path with the victim flow. For example, in Figure 1, analyzing the flow contention in SW1.P1 can not identify the true root causes, since the culprit bursts lie in SW4.P1, which is out of F1's path. Simply analyzing the queue contention on the victim flow path cannot locate the root causes accurately.

Secondly, because of high bandwidth and line-rate-start flows in RDMA, the shallow buffer can be rapidly filled up at a smaller time scale. Consequently, the performance anomalies are much more transient and frequent, making the detection and diagnosis challenging. Moreover, since switch queues are more prone to congestion, diverse factors, such as burst, ECMP imbalance and transient routing loops, can trigger NPAs. They may further cause PFC spreading congestion or even PFC storm or deadlock. However, current diagnosis methods fall short to catch diverse anomalies accurately and efficiently. For example, some in-network monitor systems (e.g., NetSeer [8]) require the prior configuration of the anomaly type and location, while the anomalies can be various and occur sporadically. Complete telemetry collection also introduces high overhead in both communication and analysis computation. Some host-based solutions, such as Trumpet [4] and Dapper [2] can store more detailed telemetry data. However, it is difficult to reconstruct the transient queue evolution at end hosts, resulting in lower accuracy.

To address the problems above, we present Hawkeye, an in-network RDMA network performance anomaly diagnosis system, via collecting the network-wide telemetry data and analyzing the anomaly causality dependency. First, to efficiently collect the relevant information for causality analysis, Hawkeye maintains fine-grained in-network telemetry data with programmable switches. Once performance degradation is detected, instead of polling the complete data from all switches, Hawkeye only collects the data from the switches relevant for the anomaly including the switches on victim flow and PFC path. Second, to accurately diagnose anomalies caused by PFC, Hawkeye proposes a novel provenance mode to analyse the heterogeneous anomaly causality including flow-level queue contention and port-level PFC spreading. We implement a prototype of the system on NS3 and conduct preliminary evaluations to demonstrate its effectiveness.

2 HAWKEYE DESIGN

Figure 2 shows the overall framework of Hawkeye. The switch preserves the recent anomaly causality information and fine-grained telemetry data. A detection agent is set on the host NIC to monitor the flow performance, such as the RTT. Once flow performance degradation (e.g., high latency) is detected, the agent sends a polling packet for the victim flow. Upon receiving the polling packet, Hawkeye polls the telemetry data on the relevant switches to the analyzer, where a provenance graph is construct to analyze the causality and locate the root cause.

2.1 Telemetry Data Structure on Switches

Hawkeye records per-epoch telemetry data at both the flow-level and port-level to monitor changes among epochs. Hawkeye keeps a fixed number of epochs, which is maintained as a ring buffer. For each flow, Hawkeye records the 5-tuple, sequence number range, packet number, total queuing depth and the number of packets

This work is supported by the National Natural Science Foundation of China under Grant 62221003 and the Beijing Science and Technology Plan Project under Grant Z231100010323013.

ACM SIGCOMM Posters and Demos '24, August 4–8, 2024, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0717-9/24/08.
https://doi.org/10.1145/3672202.3673711

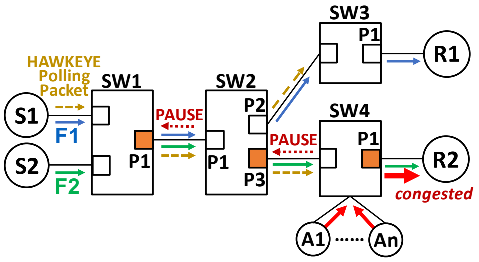

Figure 1: PFC Spreads Congestion.

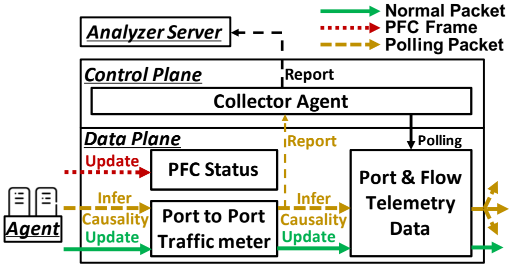

Figure 2: Hawkeye Framework.

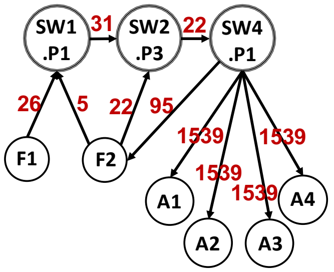

Figure 3: Provenance Graph.

paused by PFC. The slot of a flow is indexed based on the hash value of the 5-tuple. The outdated flow information is evicted upon hash collisions. For each port, Hawkeye maintains the egress queue length and the number of packets paused by PFC, indexed by the port number. As shown in Figure 2, Hawkeye updates the current port PFC status upon receiving PFC frames. Data packets queued during PFC OFF are counted as PFC-paused, updating the telemetry data accordingly. Additionally, Hawkeye introduces a port-level traffic meter that monitors port-to-port traffic within the switch to infer traffic causality, as elaborated next.

2.2 Telemetry Collection Workflow

As shown in Figure 2, when the performance of a flow degrades (e.g., RTT exceeds a certain threshold), the Hawkeye host agent sends out a polling packet to trigger the collection process. The flow information such as the 5-tuple is encoded in the polling packet header, and Hawkeye switch forwards it along the path of the victim flow. As mentioned in § 1, the performance degradation may be caused by experiencing queue contention, being paused or both. Therefore, to find the true culprit, Hawkeye should collect not only the telemetry information about the queue contention at each hop in the flow path, but also the causality information of the PFC experienced by the victim flow. It should track the spreading path of PFC and collect the relevant telemetry data.

Specifically, when a switch receives the polling packet, it 1) reports the stored telemetry data to the analyzer, and 2) identifies the relevant switches to poll telemetry data further. First, to poll the data atomically and serially, we set a "lock" register in data plane inspired by Mantis [7]. Upon receiving polling packets, the switch sets the lock bit to freeze the current data, diverts further updates into a separate set of registers, and notifies the control-plane analysis program. The controller then starts to read the telemetry data and resets the lock bit after the read is finished. Second, Hawkeye determines the egress ports (i.e., the downstream switches) which are causally relevant to this anomaly and sends out polling packets accordingly. Taking SW2 in Figure 1 for example, Hawkeye sends out a polling packet along the victim flow path (e.g., SW2.P2 for F1), so as to collect further flow contention information. Moreover, Hawkeye should also identify the relevant ports which contribute to PFC back-spreading. Hawkeye selects congested ports with significant incoming traffic from the victim flow ingress port, which facilitates the PFC backward propagation into its path. For example, since SW2.P3 is PFC paused and receives traffic from SW2.P1 (F2), a polling packet is sent through it to collect telemetry information about the origin of PFC at the downstream switch (SW4). Upon gathering all the relevant telemetry data, the root-cause bursts are identified through our provenance analysis algorithm.

2.3 Root Cause Analysis

The Hawkeye analyzer constructs the heterogeneous provenance graph including port and flow nodes to diagnose the performance anomaly. Hawkeye first constructs a port-level provenance graph to analyze the PFC causality dependency. In particular, for a paused egress port P_i, it actually "waits" for the downstream congested ports to drain out. Therefore, we define a wait-for directed edge from P_i (e.g., SW2.P3) to the downstream port P_j (e.g., SW4.P1), where the edge weight is the PFC-paused packet number in P_i. Subsequently, Hawkeye constructs the provenance between flows and ports. For a flow f_i passing through a PFC paused port P_j, f_i waits for P_j to start the transmission. We define the wait-for edge from f_i to P_j whose weight is the paused packet number of f_i at P_j. However, for a port experiencing flow contention instead of PFC, the port reversely waits for flow contention. We hence define an edge from P_j to f_i with the weight as the packet number of f_i. As Figure 3 shows, the complete anomaly causality including the root causes (e.g., flow contention), the PFC spreading path and the victim flows can be identified by constructing and analyzing the provenance graph.

3 EVALUATION AND FUTURE WORK

We implement an open-source Hawkeye prototype [3] based on the open-sourced HPCC NS3 project [1]. The network topology and traffic patterns are shown in Figure 1, and we diagnose the root cause of the performance anomalies for F1 and F2. As depicted in Figure 3, Hawkeye clearly demonstrates the anomaly experienced by F1 and F2 and its root cause. F1 and F2 are blocked by PFC at SW1.P1, which is originated from SW4.P1's flow contention. Besides, the root cause flows are also well identified via the excessive packet number of A1-A4 in SW4.P1. In future, we plan to propose a more formalized provenance model to cover more diverse RDMA NPAs, implement the complete Hawkeye system with programmable hardware (e.g., Tofino), and conduct extensive evaluations in real testbeds.

REFERENCES

[1] Alibaba. 2020. High Precision Congestion Control. https://github.com/alibaba-edu/High-Precision-Congestion-Control.
[2] Mojgan Ghasemi, Theophilus Benson, and Jennifer Rexford. 2017. Dapper: Data plane performance diagnosis of tcp. In *ACM SOSR 2017*.
[3] Hawkeye. 2024. Hawkeye-NS3. https://github.com/lixiao19/Hawkeye-NS3.
[4] Masoud Moshref, Minlan Yu, Ramesh Govindan, and Amin Vahdat. 2016. Trumpet: Timely and precise triggers in data centers. In *ACM SIGCOMM 2016*.

[5] Shicheng Wang, Menghao Zhang, Yuying Du, Ziteng Chen, Zhiliang Wang, Mingwei Xu, Renjie Xie, and Jiahai Yang. 2024. LoRDMA: A New Low-Rate DoS Attack in RDMA Networks. In *NDSS 2024*.

[6] Weitao Wang, Xinyu Crystal Wu, Praveen Tammana, Ang Chen, and TS Eugene Ng. 2022. Closed-loop network performance monitoring and diagnosis with SpiderMon. In *USENIX NSDI 2022*.

[7] Liangcheng Yu, John Sonchack, and Vincent Liu. 2020. Mantis: Reactive programmable switches. In *ACM SIGCOMM 2020*.

[8] Yu Zhou, Chen Sun, Hongqiang Harry Liu, Rui Miao, Shi Bai, Bo Li, Zhilong Zheng, Lingjun Zhu, Zhen Shen, Yongqing Xi, et al. 2020. Flow event telemetry on programmable data plane. In *ACM SIGCOMM 2020*.

[9] Yibo Zhu, Haggai Eran, Daniel Firestone, Chuanxiong Guo, Marina Lipshteyn, Yehonatan Liron, Jitendra Padhye, Shachar Raindel, Mohamad Haj Yahia, and Ming Zhang. 2015. Congestion control for large-scale RDMA deployments. In *ACM SIGCOMM 2015*.