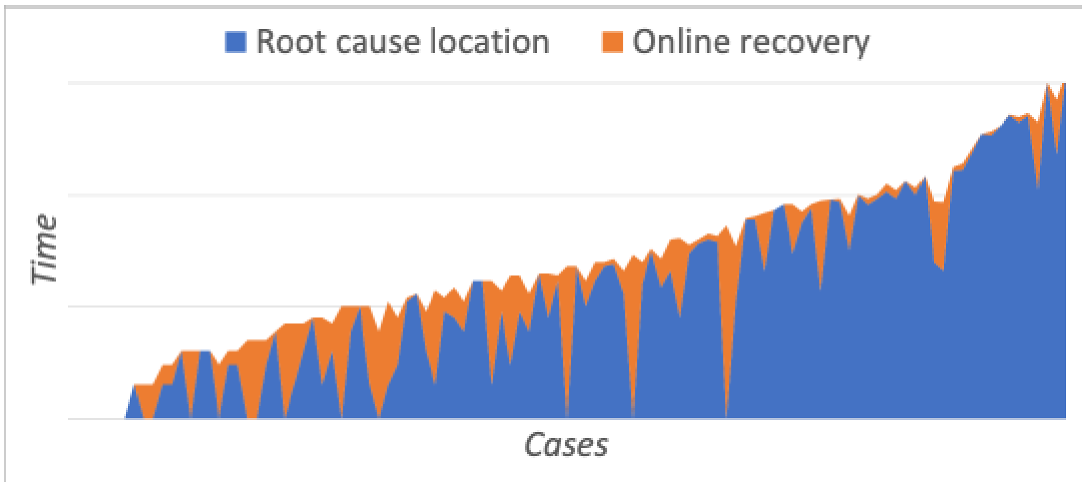
[SIGCOMM 2020] Flow Event Telemetry on Programmable Data Plane

1. Introduction

随着块存储系统从 kernel TCP 迁移到 RDMA，数据中心内端到端的网络延迟期望值从 2ms 下降到 20us。因此，偶然的网络异常波动都会影响用户体验和 IO 性能。但是快速消除 NPA(Network Performance Anomalies) 极其困难，它对网络监测的 **覆盖范围、速度、准确度** 要求很高。在大多数 NPA 案例中，消除 NPA 的 **瓶颈在于故障定位** 。



(a) Recovery time of NPAs

目前的网络监测方法并不能满足 NPA 定位的需求。比如 off-the-shelf switch 仅提供 per-interface / per-device / per-sampled-flow 粒度的聚合计数器； probe-based 监测系统只能探测 10s+ 粒度，且不能感知原始流量事件。而packet-level的监控系统虽然可以定位故障单overhead比较大。

本文提出了基于流事件的监控系统NetSeer来监控造成NFA的数据面事件

2. Architecture

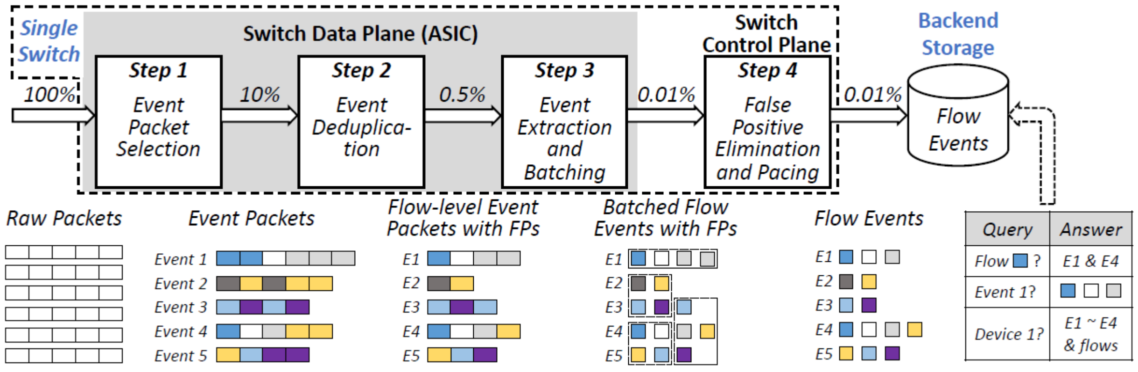


Figure 2: The architecture and workflow of NetSeer. (*FP* stands for false positive.)

3.Flow Event

- Packet drop：会导致timeout/retransmission/slowing down at senders
- Packet queuing:会延迟包的传输和到达,通常拥塞造成
- Packet out-of-order: 导致NAK,deep buffer(因为要等待无序包到达才能交付，缓冲区会保持很满状态一段时间) 通常因为path change引起

NAK：非确认帧，当在一定时间内没有收到某个数据帧的ACK时，回复一个NACK。
在发送过程中，如果一个数据帧计时器超时，就认为该帧丢失或者被破坏，接收端只把出错的的帧丢弃，其后面的数据帧保存在缓存中，并向发送端回复NAK。发送端接收到NAK时，只发送出错的帧。

- Packet pause: PFC造成

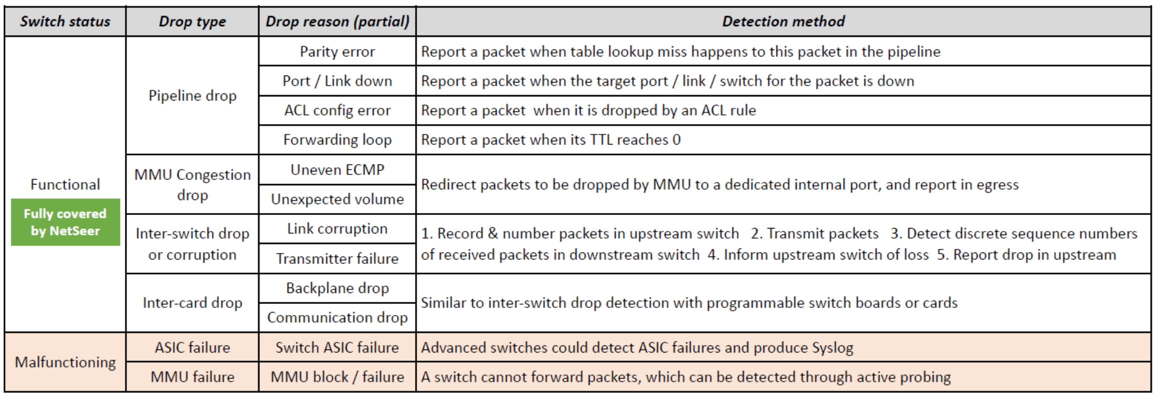
  基于优先级的流量控制（PFC: Priority-based Flow Control）在IEEE:802.1Qbb标准文档中定义，对传统流控的暂停机制一种增强。与传统的流控机制相比，当出现拥塞时传统流控但会阻止一条链路上的所有流量。而PFC允许在一条以太网链路上创建8个虚拟通道，并为每条虚拟通道指定一个IEEE 802.1P优先等级（cos），允许单独暂停和重启其中任意一条虚拟通道，同时允许其它虚拟通道的流量无中断通过。这一方法使网络能够为单个虚拟链路创建无丢包类别的服务，使其能够与同一接口上的其它流量类型共存。其实PFC就是普通流控功能的一种增强。

4.How to identify flow event

Congestion, path change and pause detection

- Congestion：通过switch内的进出时间戳得到排队时长，超过阈值就记录
- Path Change: 记录 flow 的进出 port，把新流的第一个包和老流的第一个路径变化的包作为 event packet。因为硬件资源有限，快速替换老流，保证新流被记录，虽然有时会导致老流被当作新流被上报多次，但是后续可做聚合
- Pause Detection：在ingress口去检测PFC Messages(PAUSE or RESUME)识别状态
- Packet Drop:

Intra-detect 比较容易把事件上报代码嵌入ASIC中，当发生Pipeline Drop和MMU Drop的时候就可以把Event Packet给上报



Switch status	*Drop type*	*Drop reason (partial)*	*Detection method*
Functional **Fully covered by NetSeer**	Pipeline drop	Parity error	Report a packet when table lookup miss happens to this packet in the pipeline
		Port / Link down	Report a packet when the target port / link / switch for the packet is down
		ACL config error	Report a packet when it is dropped by an ACL rule
		Forwarding loop	Report a packet when its TTL reaches 0
	MMU Congestion drop	Uneven ECMP	Redirect packets to be dropped by MMU to a dedicated internal port, and report in egress
		Unexpected volume	
	Inter-switch drop or corruption	Link corruption	1. Record & number packets in upstream switch 2. Transmit packets 3. Detect discrete sequence numbers of received packets in downstream switch 4. Inform upstream switch of loss 5. Report drop in upstream
		Transmitter failure	
	Inter-card drop	Backplane drop	Similar to inter-switch drop detection with programmable switch boards or cards
		Communication drop	
Malfunctioning	ASIC failure	Switch ASIC failure	Advanced switches could detect ASIC failures and produce Syslog
	MMU failure	MMU block / failure	A switch cannot forward packets, which can be detected through active probing

Figure 4: The types and reasons of packet drops, and the methods NetSeer uses to detect them.

Inter-detect

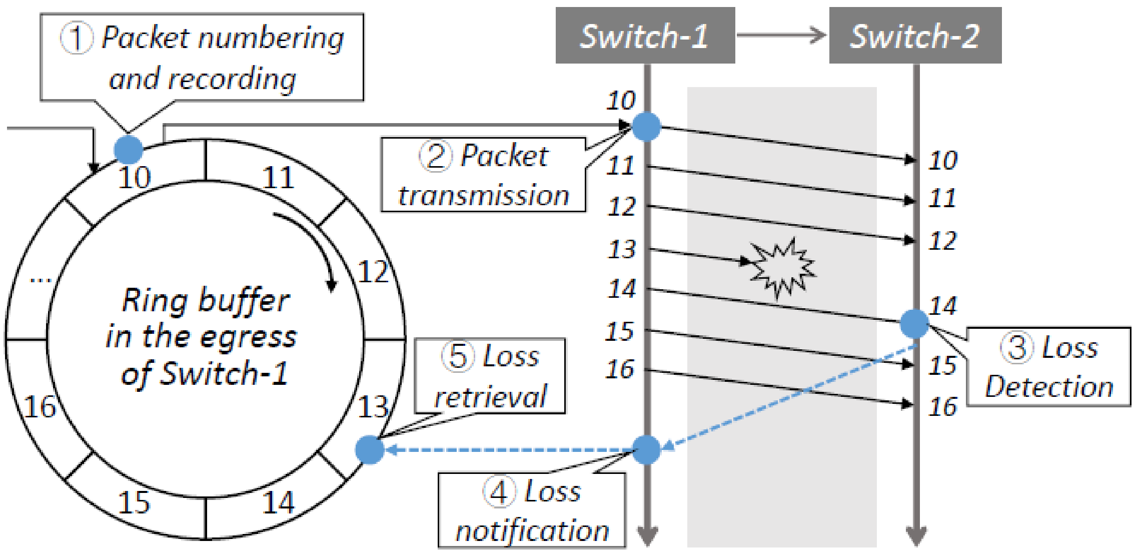


Figure 5: Inter-switch drop/corruption detection.

ring buffer会缓存一些包，然后当丢掉的包后面的包到达switch2的时候，会触发switch2向switch1汇报(start,end)

但是一旦发生连续丢包，因为buffer有限，被override掉的包丢掉的话就没法上报，但是因为packet id是唯一的，所以不会出现误报。

5.Flow Event Generation & Compression

we define redundant event packets as those belonging to the same flow and encountering the same event.This could reduce the monitoring traffic volume from O(#event packets) to O(#event flows)

- **Event packets to flow events**： 通过 hash 表基数，达到阈值/发生冲突产生替换时生成流事件。同样，当大流被产生冲突替换时，后续还会替换回来，导致一条流产生多个事件。

Algorithm 1: Deduplication based on group caching

```
Input: Event packet $\mathcal{P}$; Group caching table cache[]
Function event_packet_deduplication ($\mathcal{P}$, cache[])
    index ← calculate_hash($\mathcal{P}$.flow_info);
    if cache[index].flow_info is equal to $\mathcal{P}$.flow_info then
        cache[index].counter ++;
        if cache[index].counter ≥ cache[index].target then
            produce_event(cache[index]);
            cache[index].target ← cache[index].target + C;
    else
        cache[index].flow_info ← $\mathcal{P}$.flow_info;
        cache[index].counter ← 1;
        cache[index].target ← C; // C is a constant;
        produce_event(cache[index]);
```

- 对于 ACL 丢包事件，聚合是按 ACL 粒度，而不是流粒度，因为通常 ACL 丢包属于正常行为。而且ACL的头部就包含了packet的信息。对ACL的每一项都维护一个packet drop counter

- **Event information Extraction**： 只记录必要的流信息，比如 5-tuple、switch-port-queue、事件相关数据（拥塞-延迟，丢包-丢包原因）

6.Circulating Flow Event Batching

单独的事件信息只占 24 bytes，以太网最小帧 64 bytes，如果直接发送会导致额外开销。考虑到以太网包长，建议 50个event为一个batch 发送。但是 switch 资源限制不足以维持 50 个 event，因此通过采取构造一个circulating event batching packet，然后在 switch pipeline 内循环拼接，避免在处理阶段内维持内存，直到拼接完 50 个 event。event会先进栈, 然后在栈里面不断pop，拼接到circulating event batching packet的payload之中。

7.False Positive Elimination

false positive 即重复上报的事件。switch CPU 维持一个 hash 表来消除重复，为了节省 CPU，可以在 switch pipeline 中计算 hash 值，把值给拼到event上，然后可以直接访问index来获得hash值。

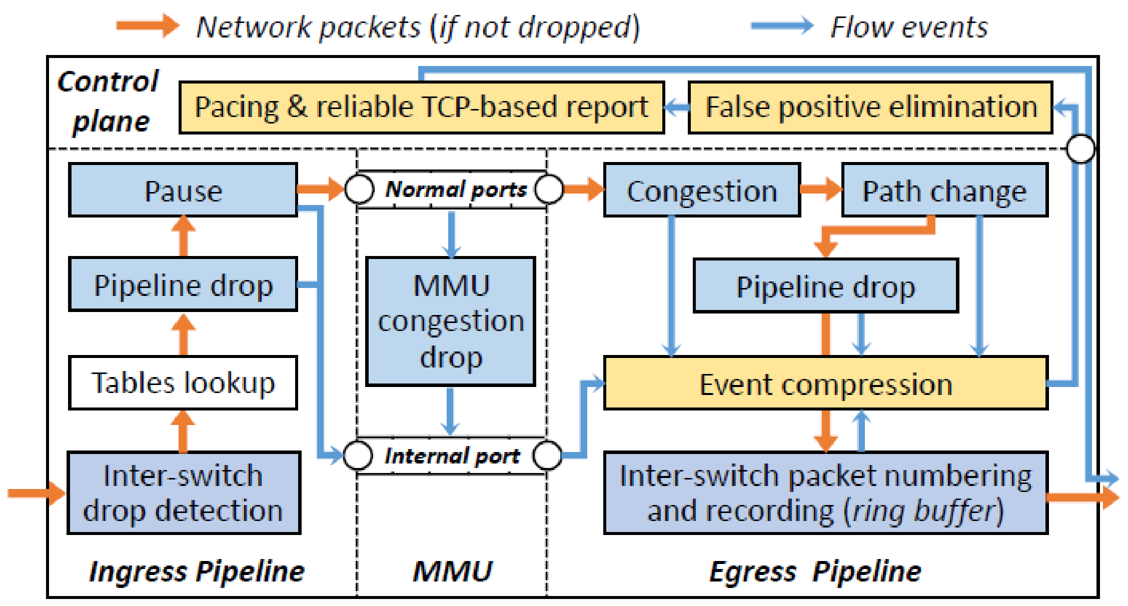


Figure 6: NetSeer switch implementation.

Event Formats：

• Flow (13B): <5-tuple> for TCP/UDP packets. Flow fields can be flexibly defined and extended according to packet formats.(记录flow信息)
• Congestion (5B): <egress port, egress queue, queue latency> (识别交换机的哪个出口和哪个队列上发生了Congestion,同时记录拥塞的排队时延)
• Path change (2B): <ingress port, egress port>.
• Pause (2B): <egress port, egress queue> (识别交换机的哪个出口和哪个队列上发生了Pause)
• Drop (3B): <ingress port, egress port, drop code>. Packet drop reasons are encoded into the drop code field.