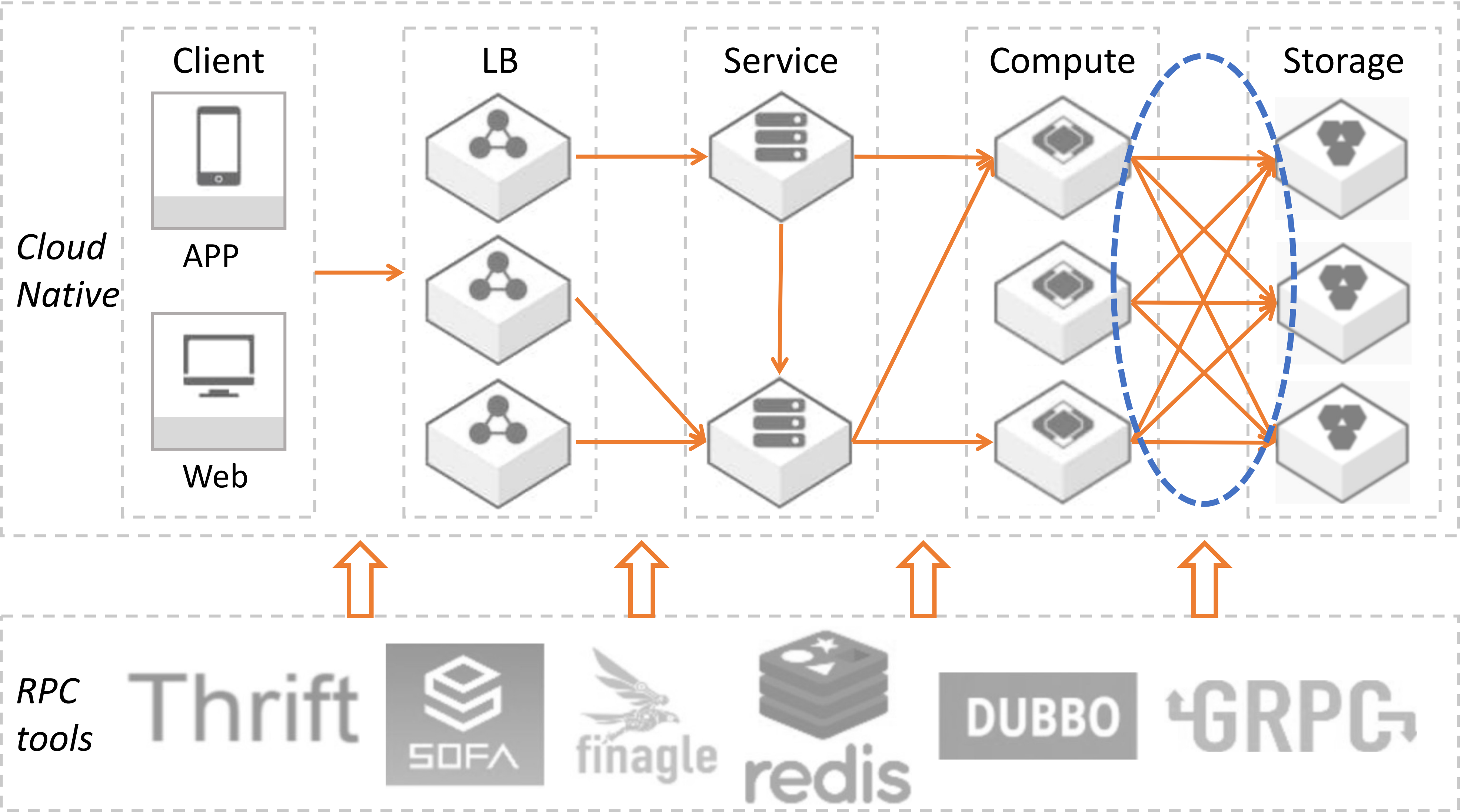
Buffer-based End-to-end Request Event Monitoring in the Cloud

Kaihui Gao

Chen Sun, Shuai Wang, Dan Li, Yu Zhou, Hongqiang Harry Liu, Lingjun Zhu, Ming Zhang

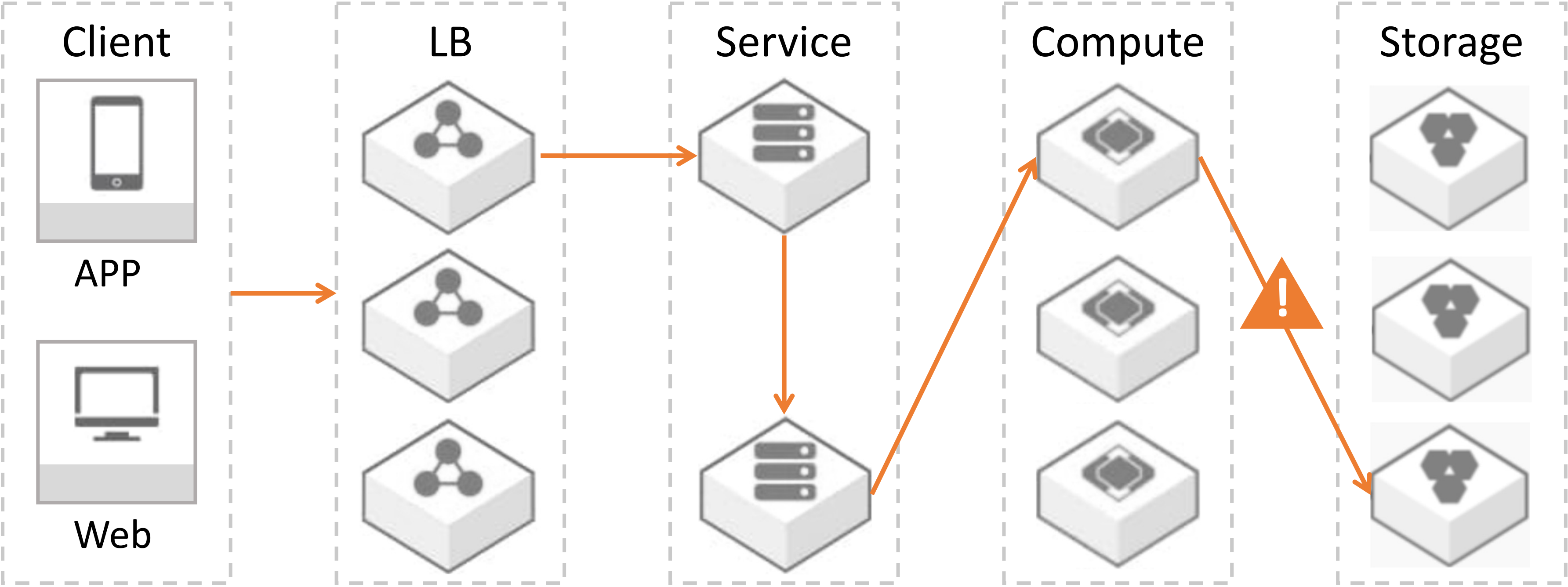
RPC: Fundamental Component in Clouds



440M RPC
per hour
per cluster

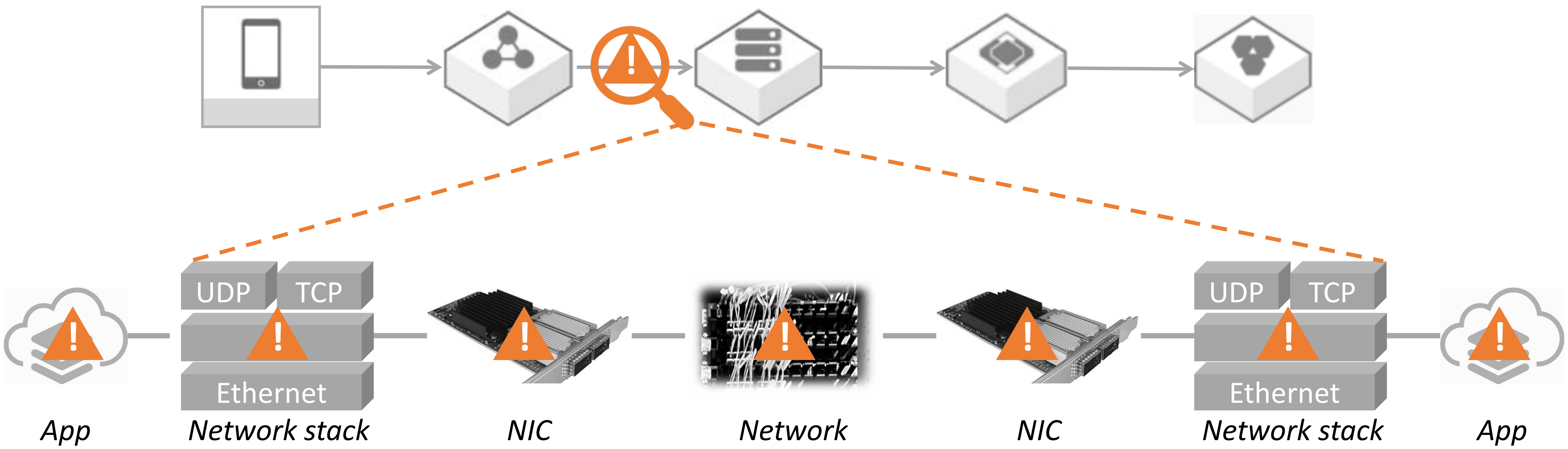
RPC Latency is Crucial for App Performance

$5 \times 30 = 150ms$ *vs.* $5 \times 200 = 1s$



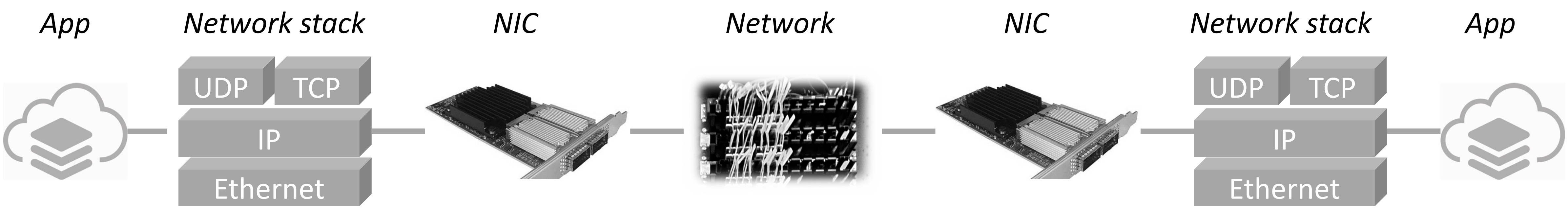
44k (0.01%) RPC latency anomalies
per hour
per cluster

RPC Latency Anomaly is *Difficult to Diagnose*



Possible problem at all layers managed by different teams

RPC Latency Anomaly is *Difficult to Diagnose*



Any ideas what happened to RPC#2?

Not sure… Got IP and time?

Yeah, check 192.168.1.22 @15:00?

Emm, I see a burst, RPC#2 maybe affected

SNMP counters

Active probing

Sampling

Telemetry

Statistical analysis

TCP profiling

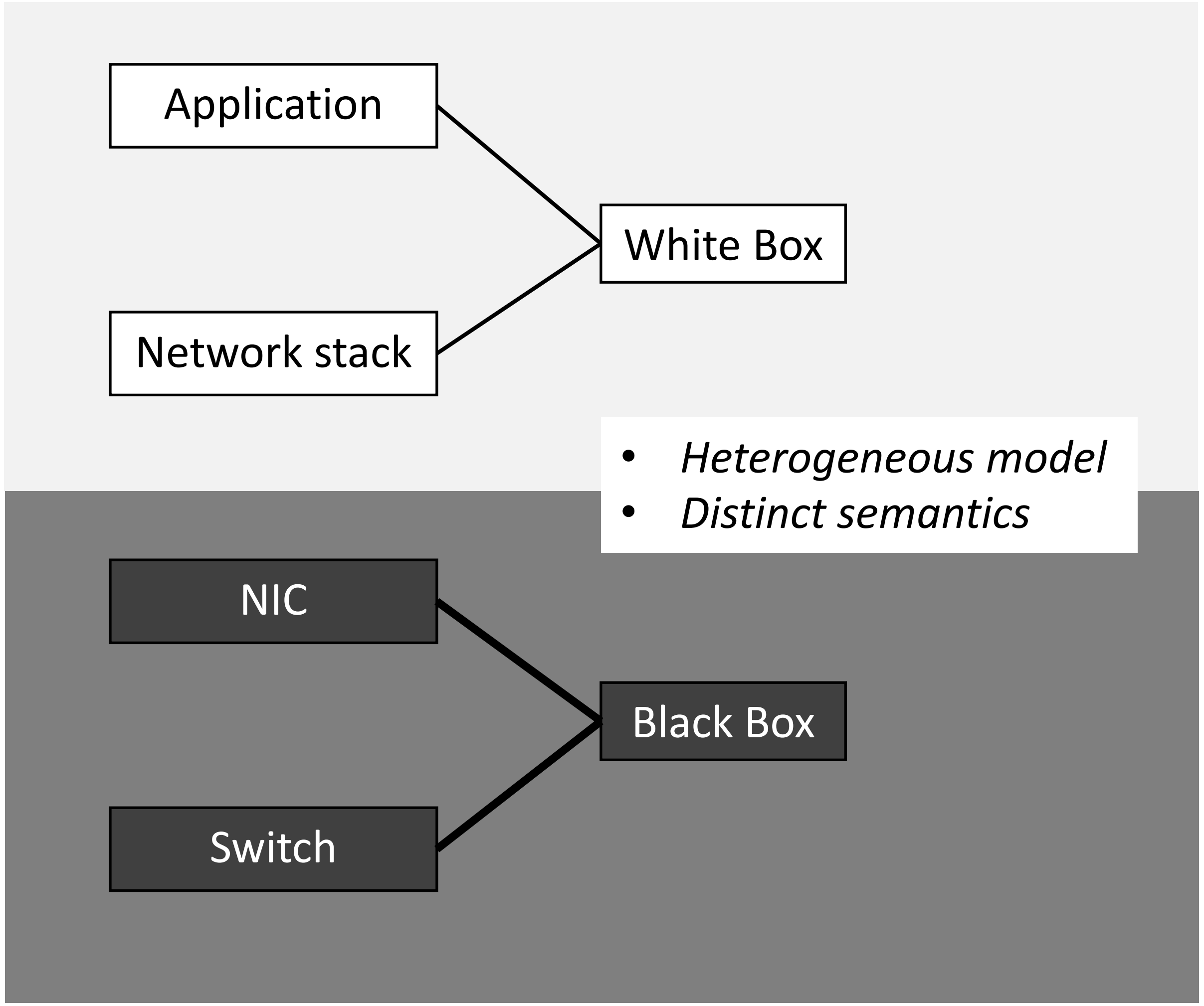
TCP replay

Kernel statistics

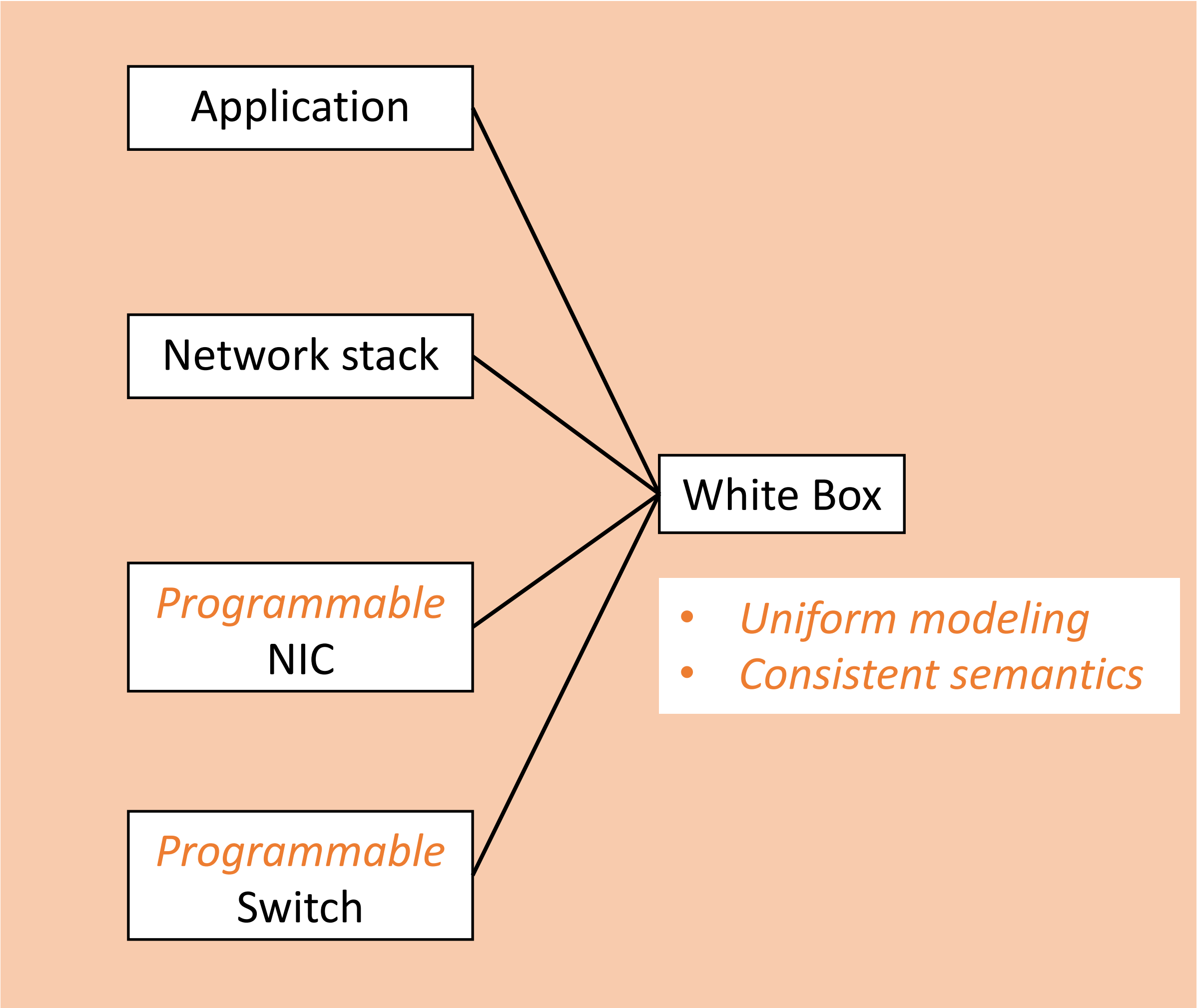
Incomplete Visibility

Inconsistent Semantics

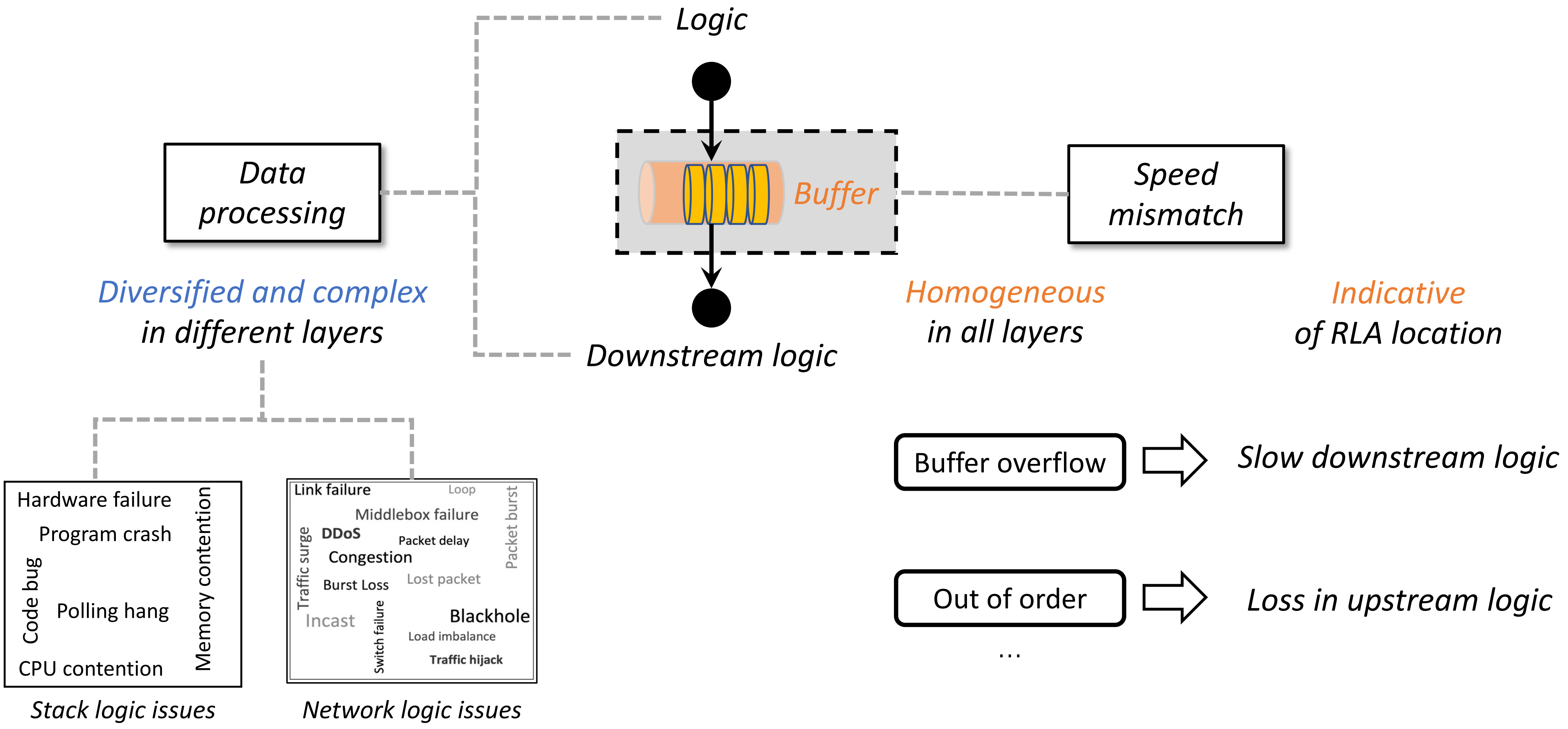
Why Difficult?



Any Opportunity?



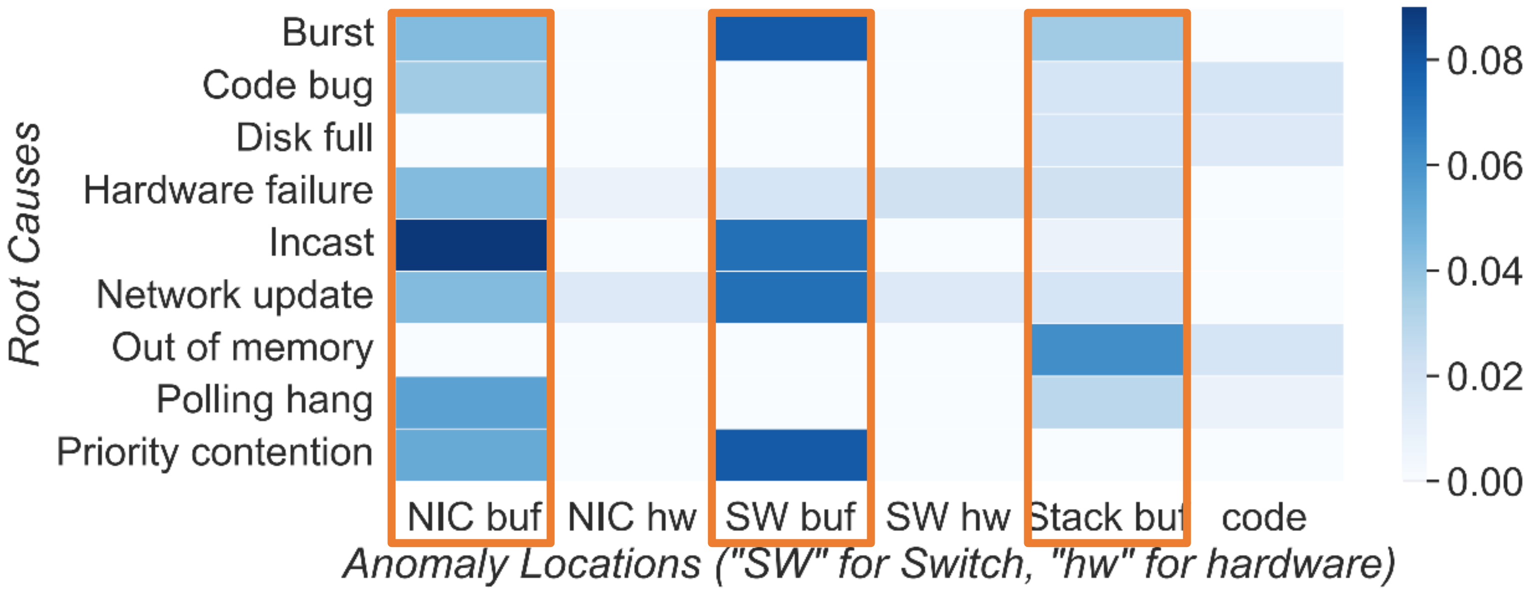
What's the Right Unifom Model for RLA Locating?



Motivation for Using Buffer to Locate RLAs

~500 Production RLA tickets

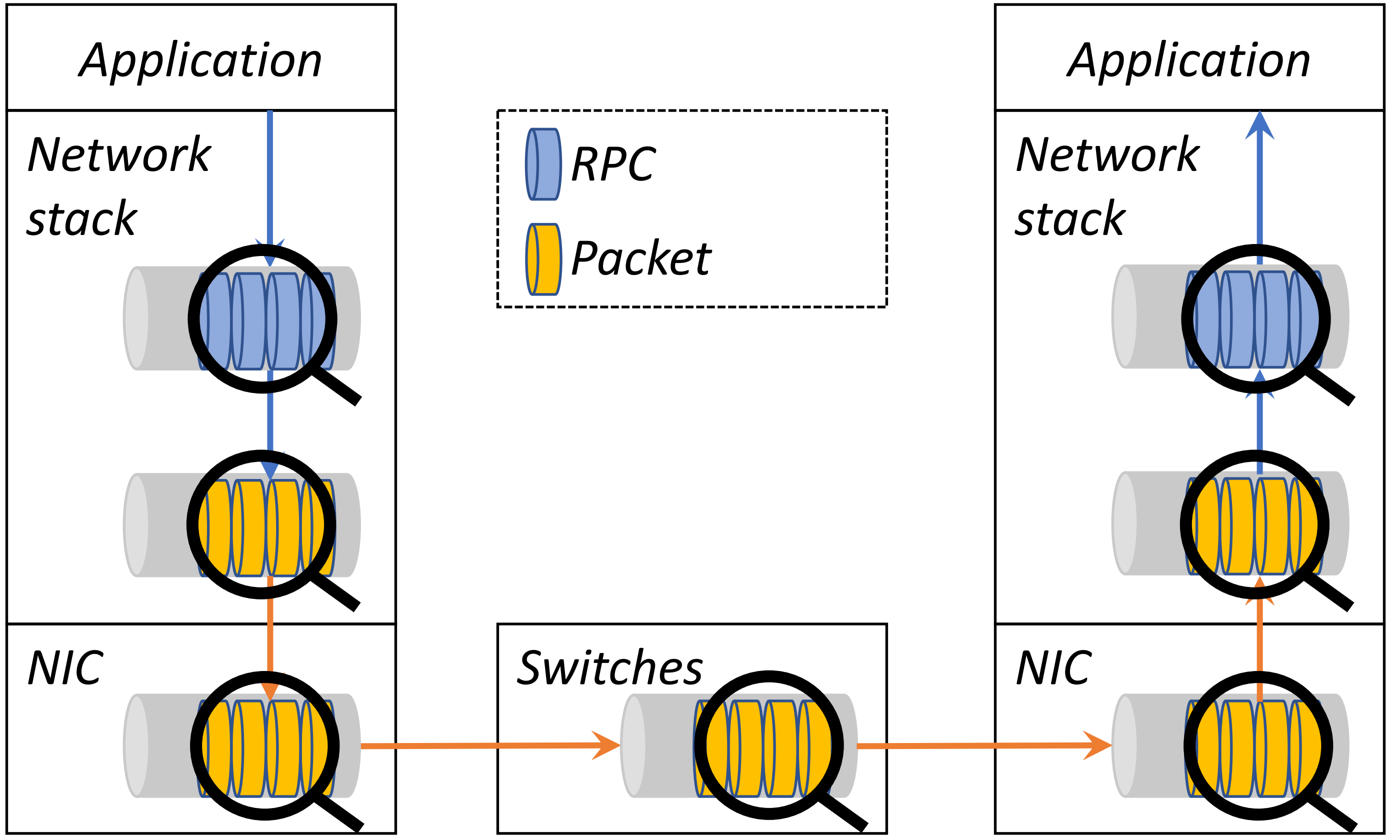
in Alibaba Cloud Storage Service



*90% expose **buffer anomalies***

among all RLA tickets

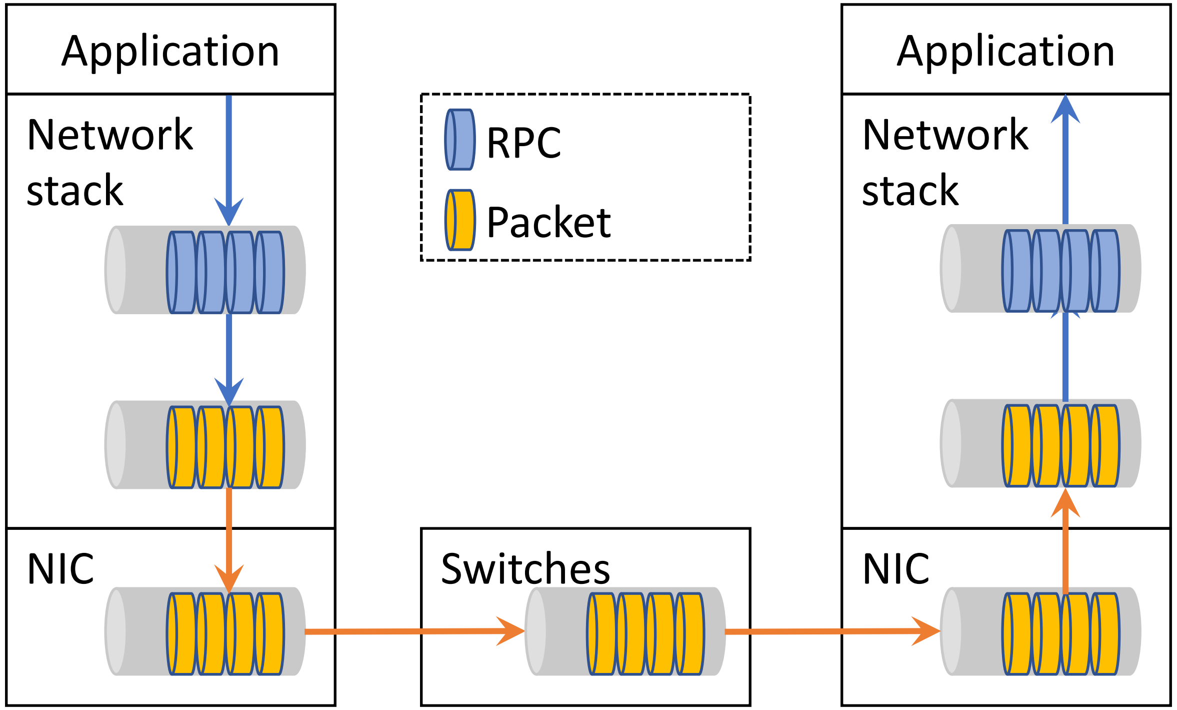
Buffer Chain Abstraction



BufScope: Diagnosing RLAs based on **Buffers**

- Complete visibility at all layers *- Consistent RPC-level semantics* *- Low overhead*

Design Challenges



*How to **define events** for various buffers*

- *Drop* for lossy buffer
- *Pause* for lossless buffer
- *Out-of-order* for TCP buf
- …

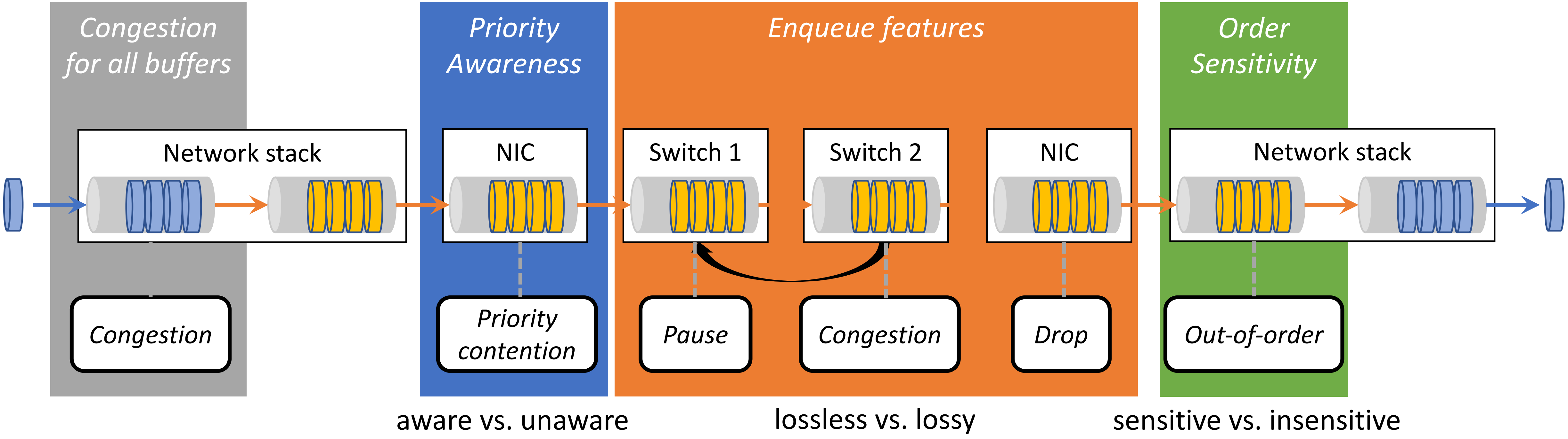
*How to **reduce CPU overhead***

- *For software monitoring and semantic operations*

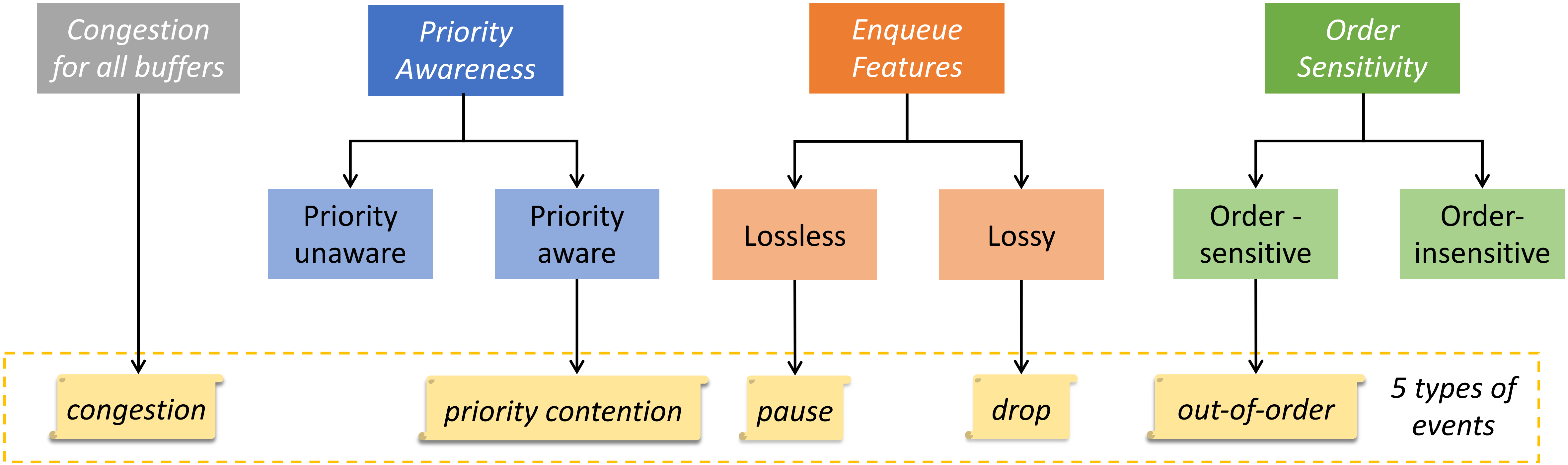
*How to **obtain RPC semantics** from packets **in NIC and switch***

- *RPC semantics are encapsulated in packet payload*

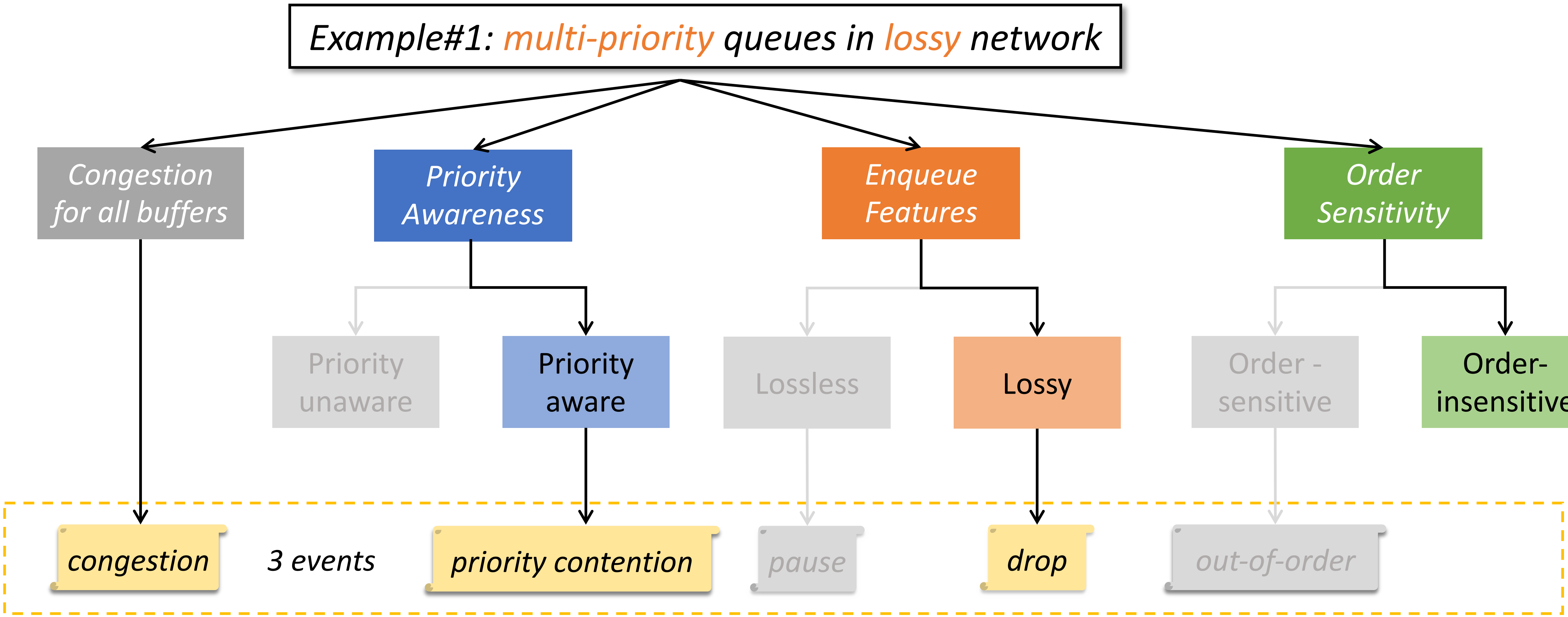
Event Definition based on Buffer *Classification*



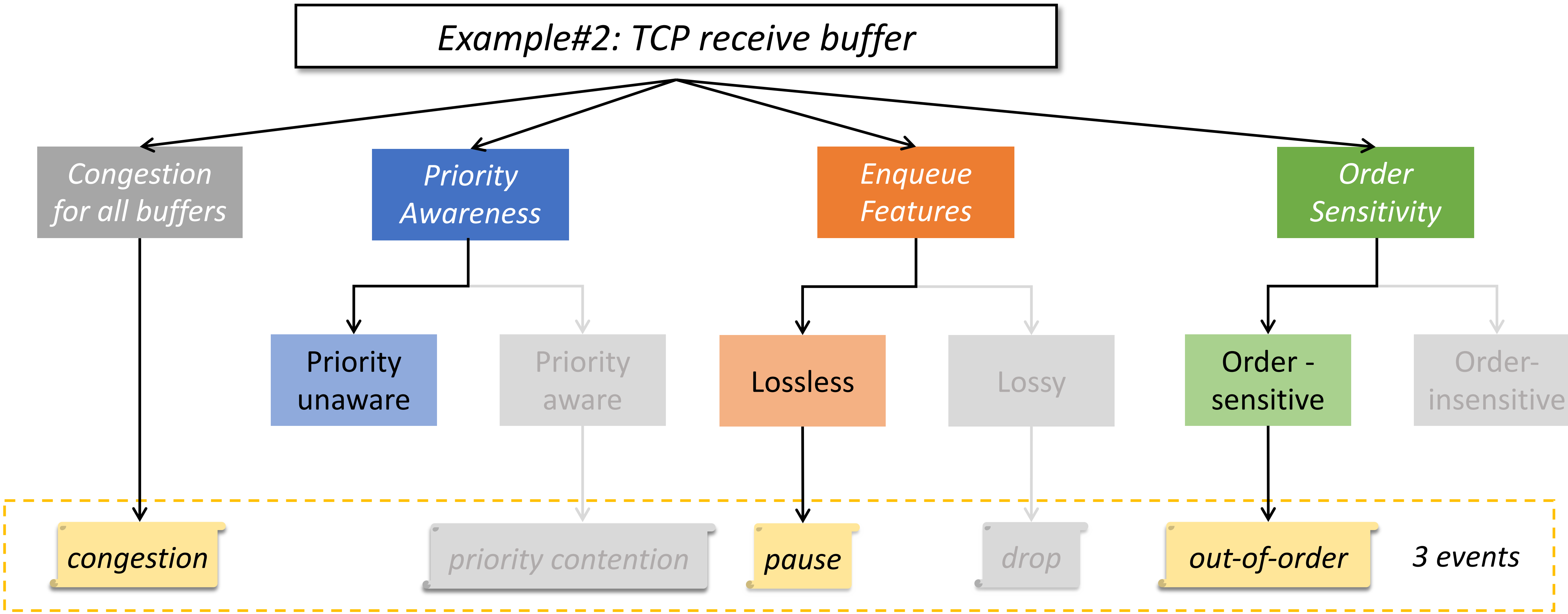
Event Definition based on Buffer *Classification*



Event Definition based on Buffer *Classification*

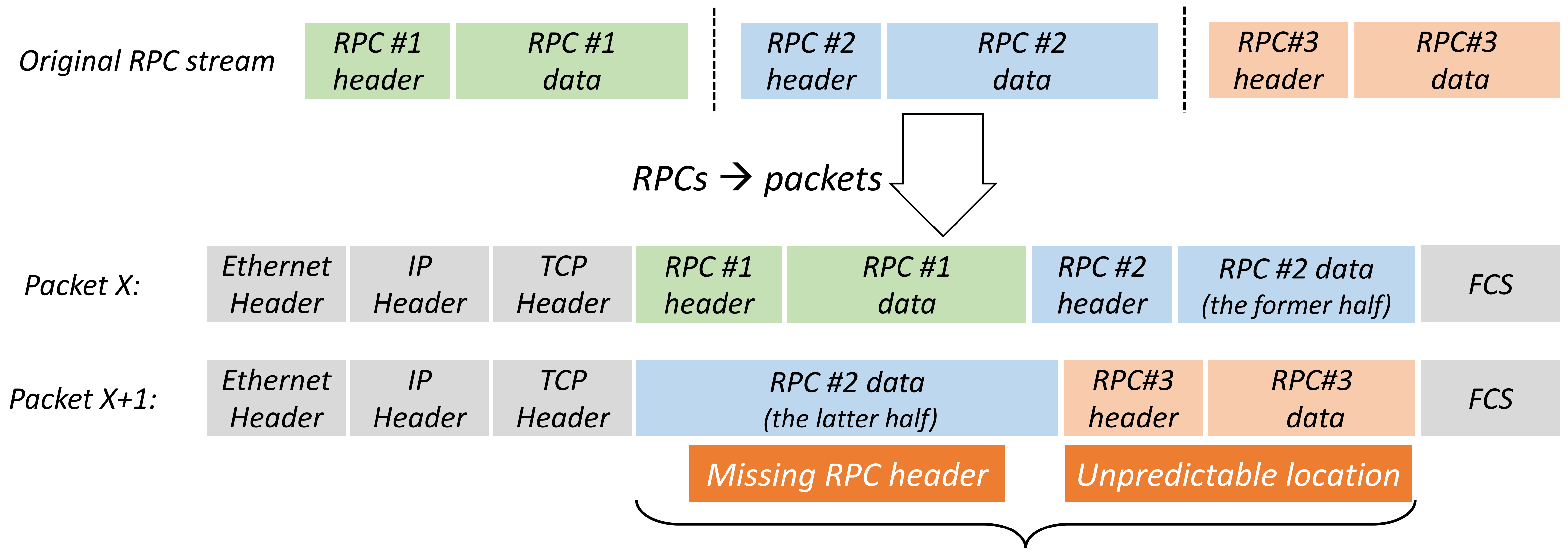


Event Definition based on Buffer *Classification*



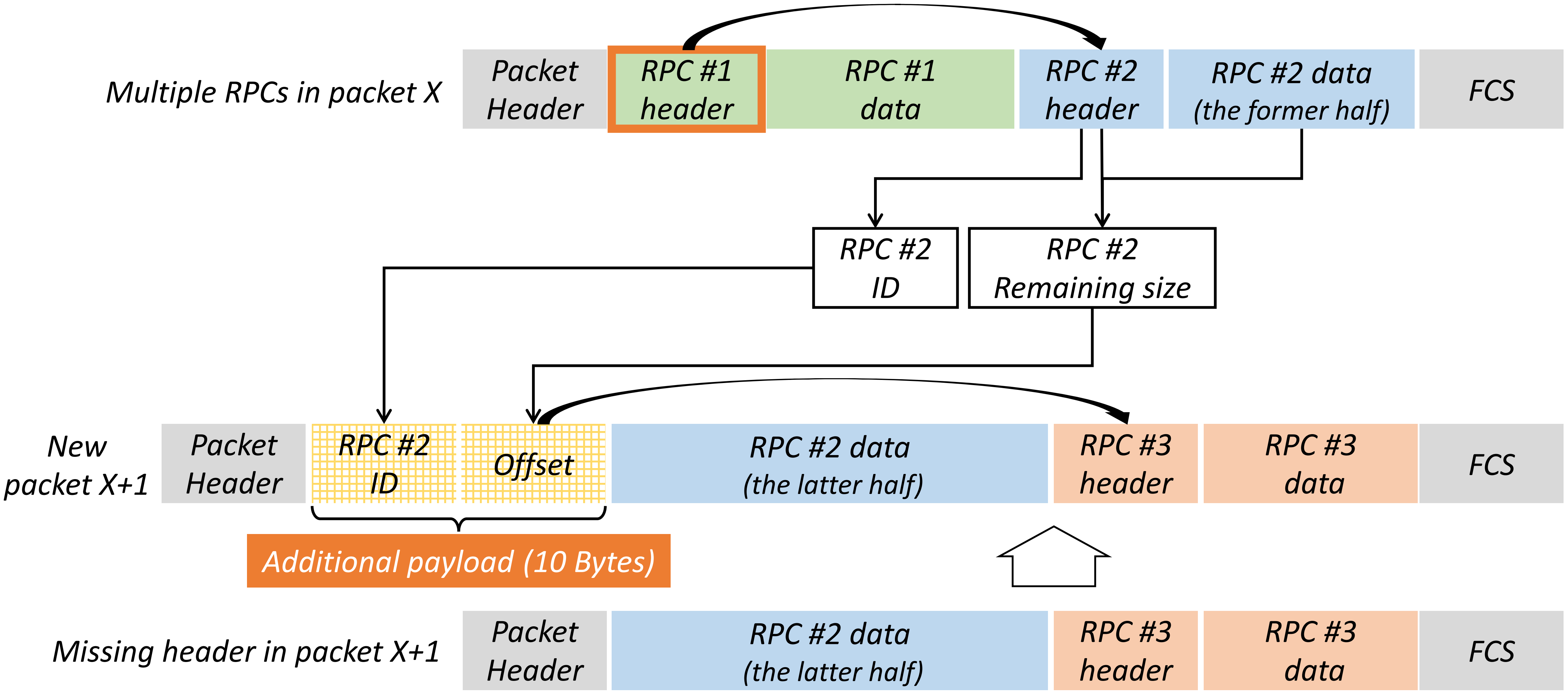
Obtaining RPC Semantics in NIC and Switch

Goal: Using RPC ID to correlate events in all layers

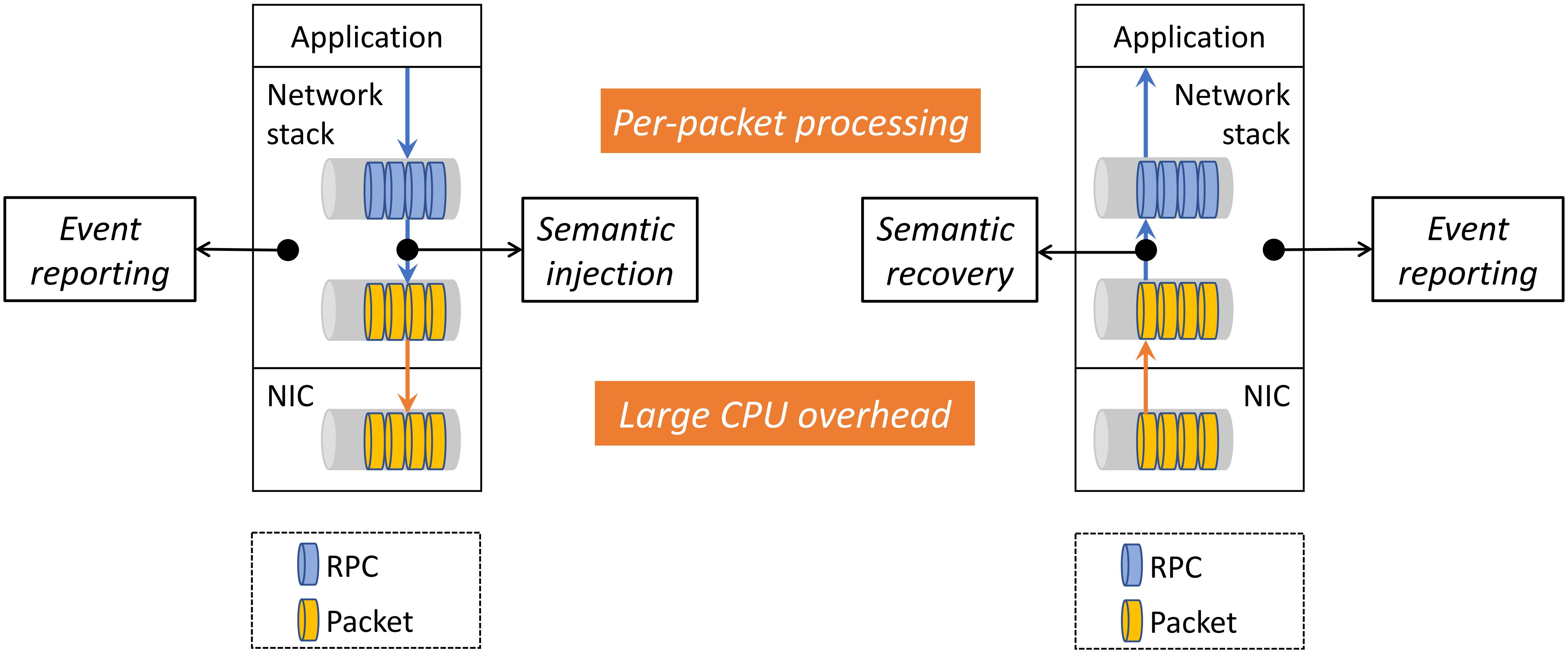


Difficult to obtain RPC semantics for NIC & Switch

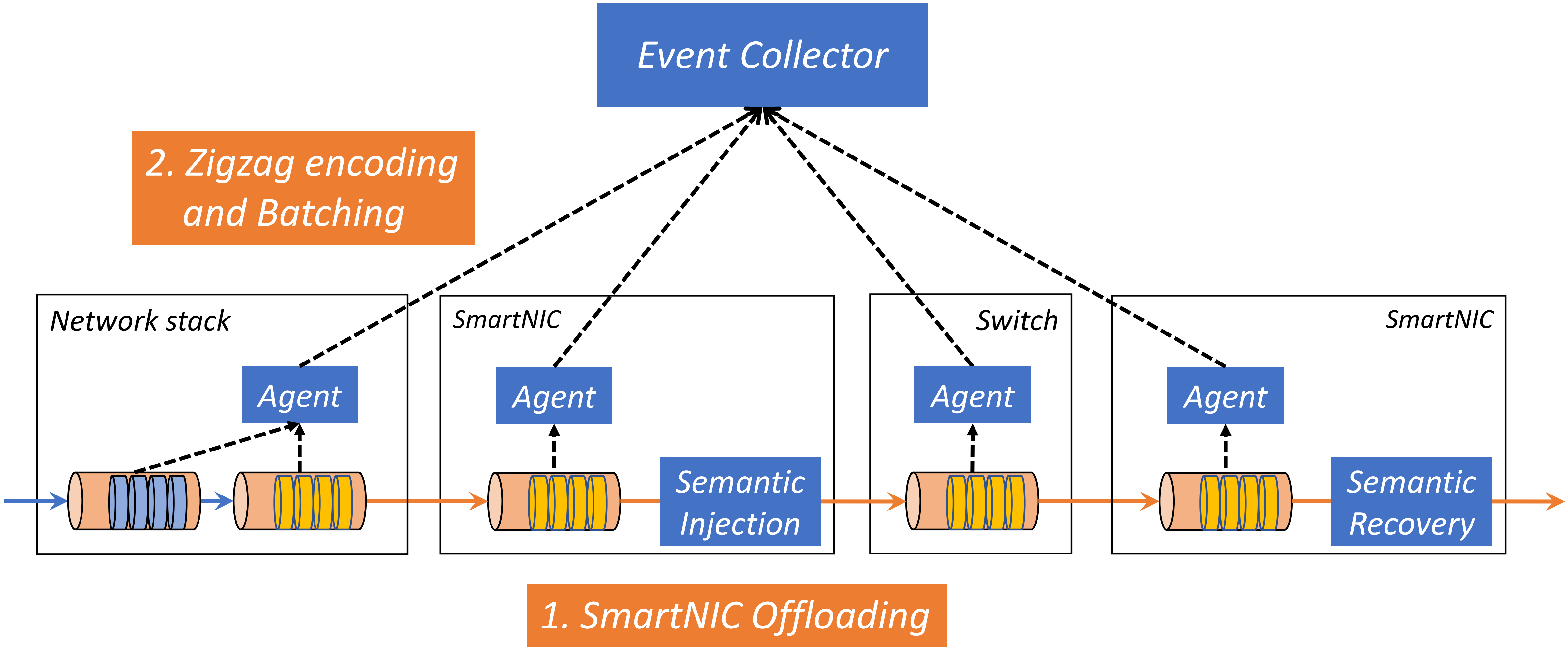
Per-packet RPC-level Semantic Injection



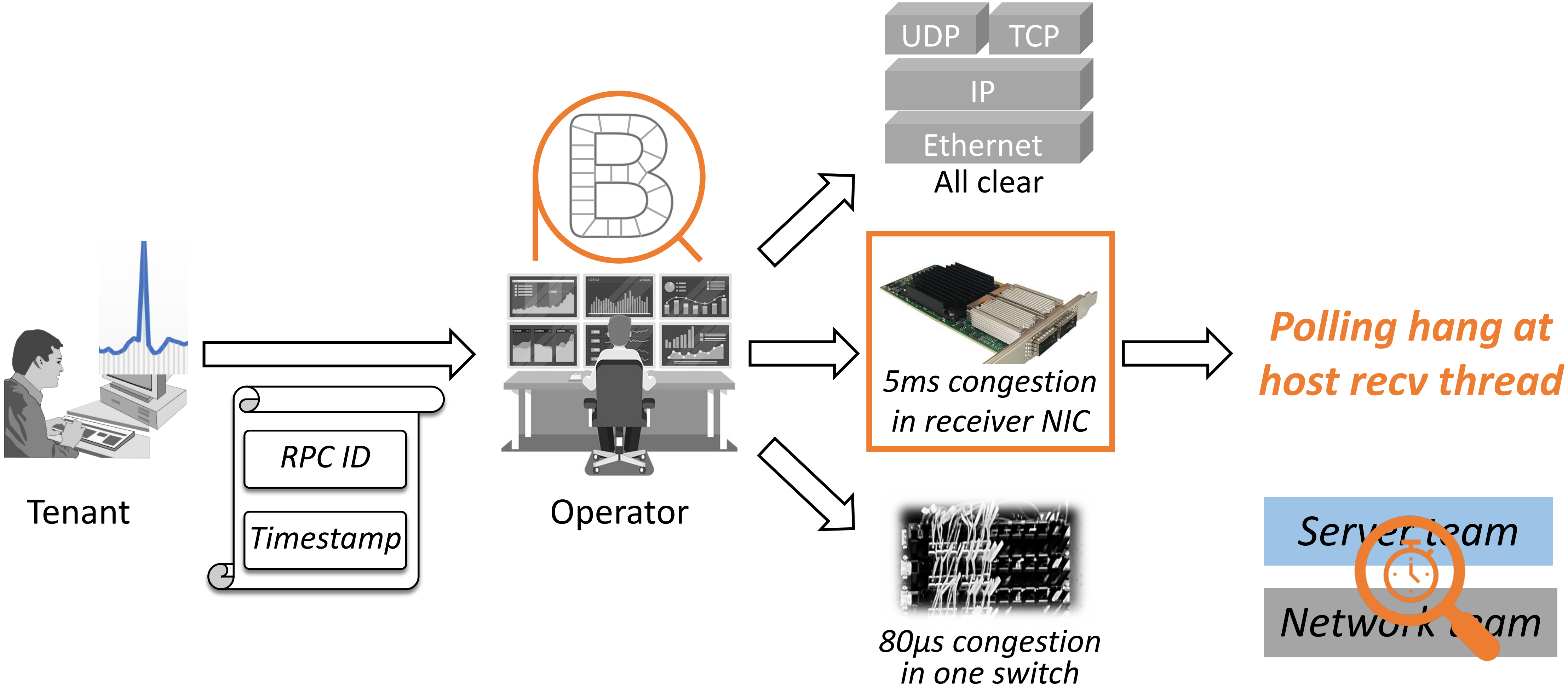
CPU Overhead



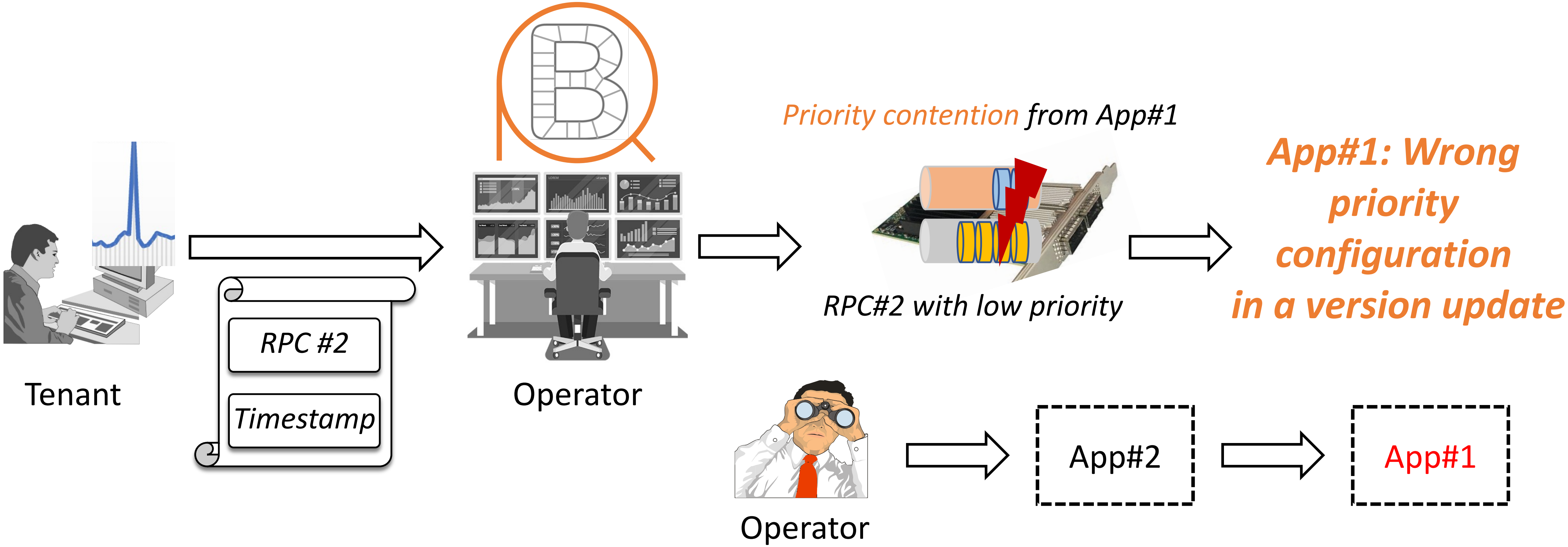
BufScope Implementation with Low Overhead



Case Study 1: Polling Hang in Host Thread



Case Study 2: Priority Contention in Recv NIC



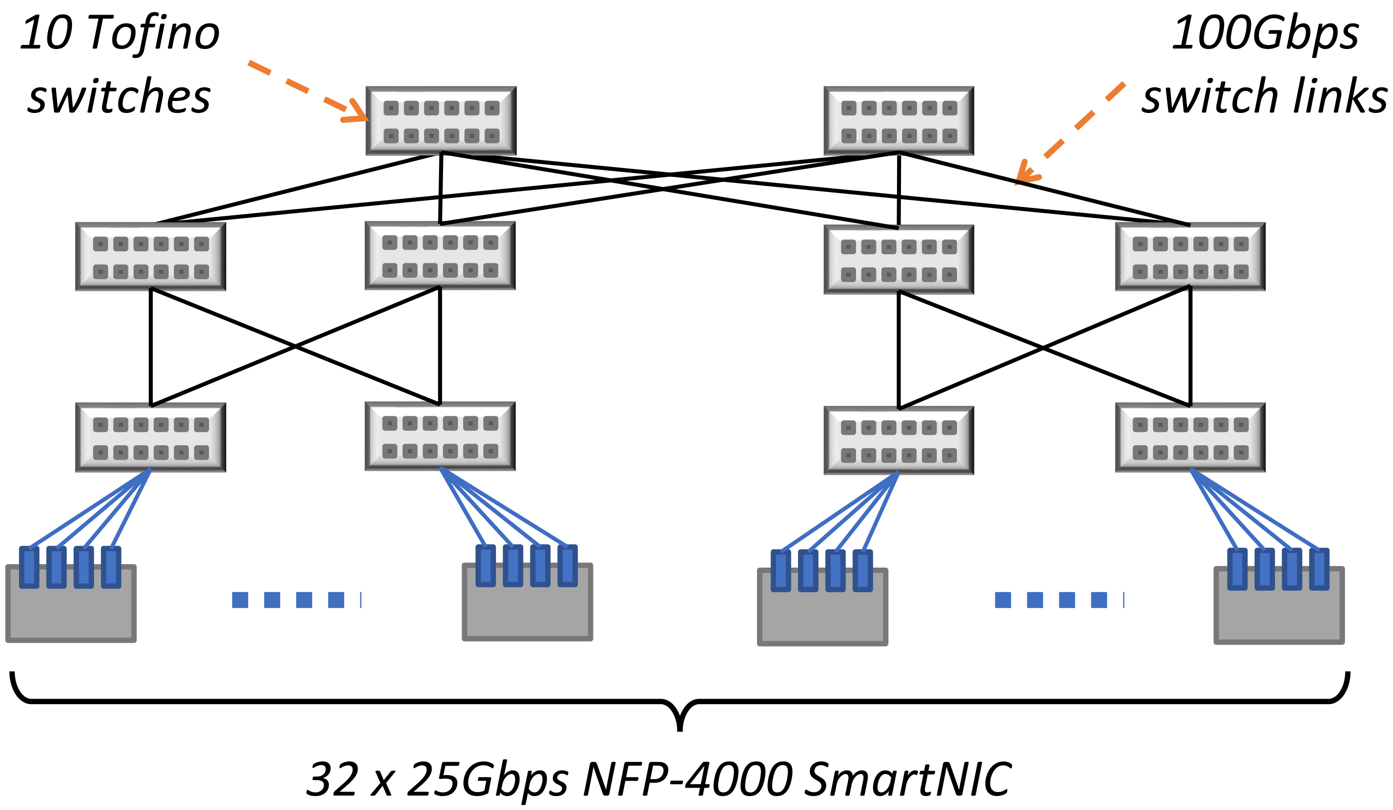
Evaluation Setup

Evaluation Goal
Coverage & Overhead

Baseline
Tracing+TCP monitor+ Network monitor

RPC Framework
Finagle, Alibaba storage

Traffic Patterns
DCTCP, VL2, Storage, Web



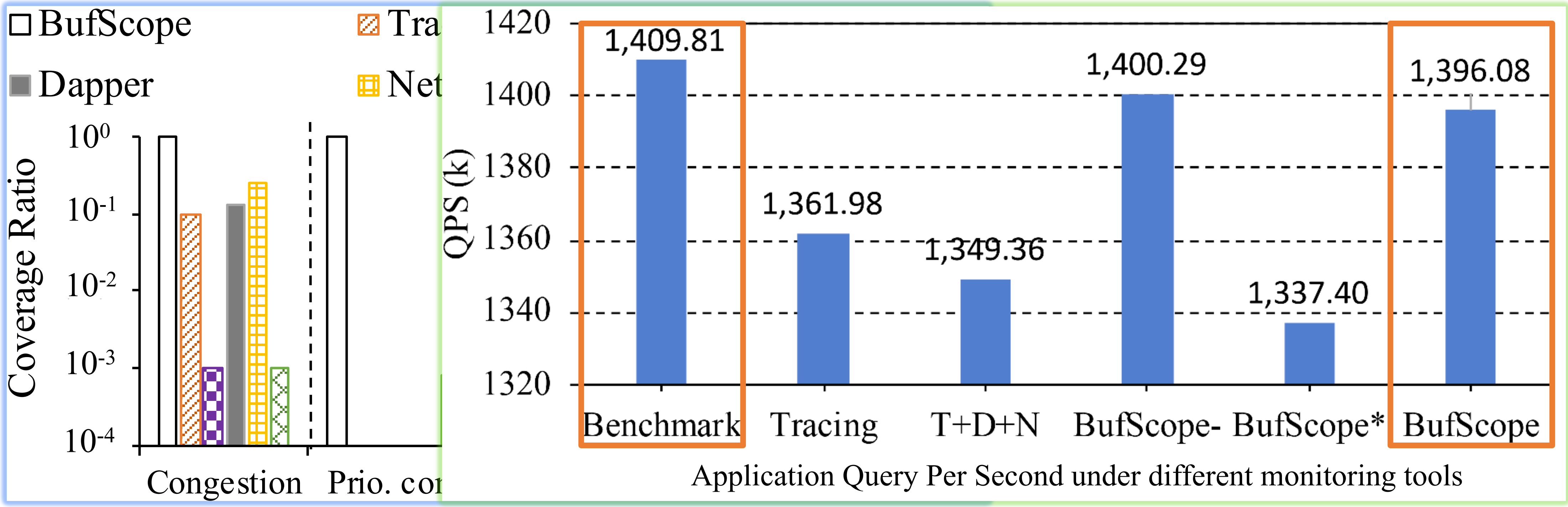
Microbenchmarks

Event Coverage

100% event coverage

Performance overhead

<1% throughput drop



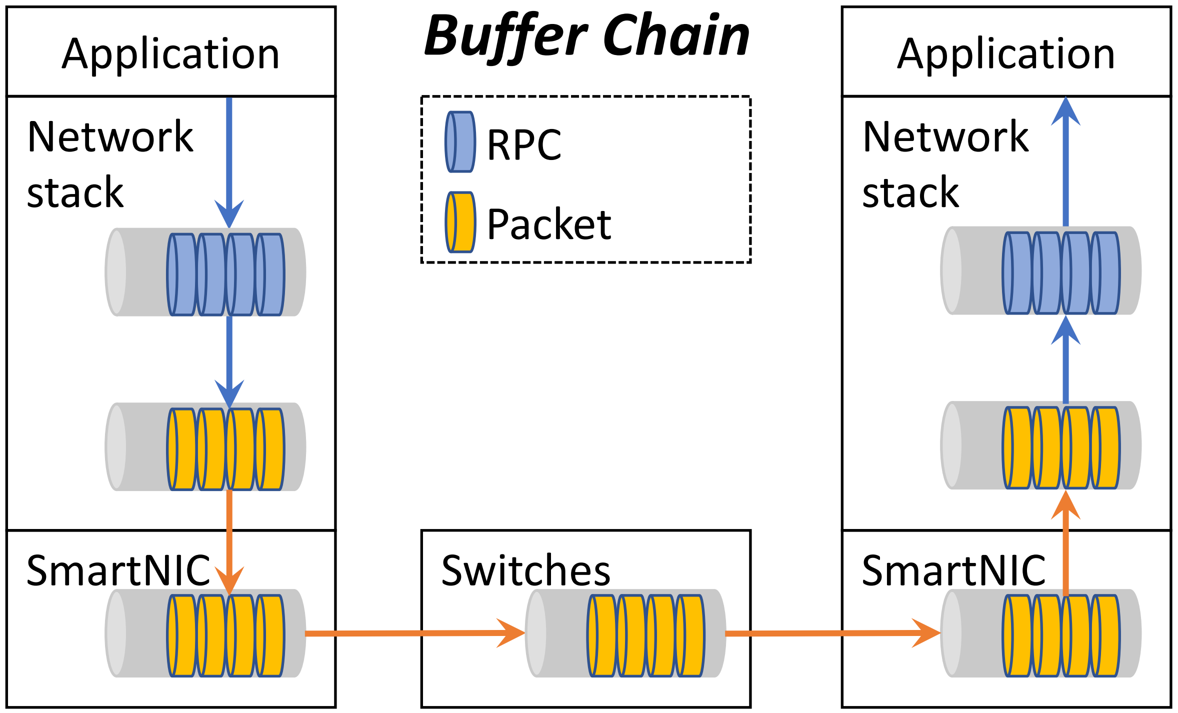
BufScope: Diagnosing RLAs based on **Buffers**

Boost RLA locating from days to minutes

Buffer Classification

Semantic Injection

Offloading & Compression



Thanks for your interest in BufScope

gkh18@mails.tsinghua.edu.cn