No packet left behind

Minimising packet loss through automated network operations

John Evans, Fabien Chraim

AWS

NANOG 88, 13 June 2023

No packet left behind

The job of a network is to transport packets

Packet loss is the primary signal of when a network is not doing its job

But some level of packet loss is normal in TCP/IP networks

How can we minimize anomalous packet loss through automated network operations?

Automated network operation

5 stages of auto-remediation

Focus of this presentation

1. Detect and isolate impact
2. Identify root cause
3. Mitigate impact
4. Remediate the underlying problem
5. Return to service

Automated network operations

Foundations – design for operation

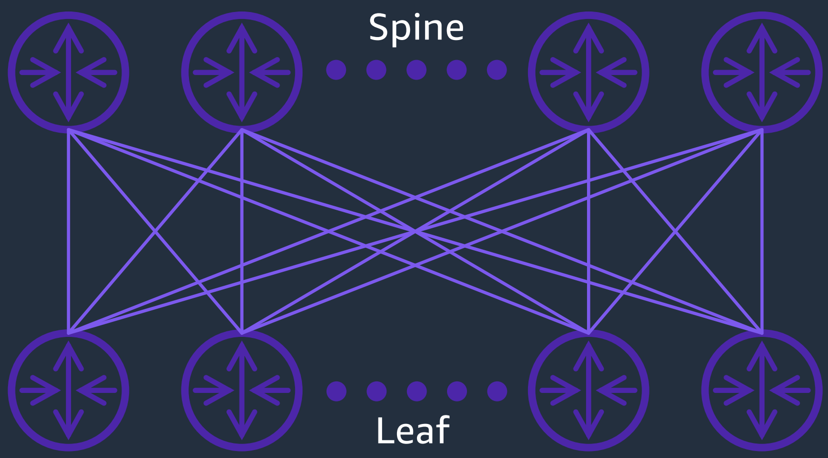
A. Network architecture that supports automated operations

B. Accurate signal of impact which indicates cause

C. Small number of auto-mitigation actions

D. Systems to safely apply those actions

Network architecture that supports automated operations



Multi-tier Clos (a.k.a. fabric) architectures – key building block to support automation

Both within the DC and WAN

Individual devices can be brought into and taken out of service without impact

With the right control plane and data plane support

Large-chassis vs. fixed form factor routers

Large-chassis routers

- Fewer devices to manage
- Multiple-stage forwarding architecture
- More ports, larger failure domain
- Dual monolith: redundancy within boxes

Fixed form factor routers

- Many devices to manage
- Typically single ASIC - simpler forwarding architecture
- Fewer ports, contained failure domain
- 3-tier Clos: redundancy between boxes

Automated network operations

Foundations

A. Network architecture that supports automated operations

B. Accurate signal of impact which indicates cause

C. Small number of auto-mitigation actions ⬅

D. Systems to safely apply those actions

Working backwards from auto-mitigation

There are only a relative small number of auto-mitigation actions

- Take a device / link / set of devices and or links out of service
- Put a device / link / set of devices and or links back into service
- Roll-back a change
- Move traffic
- Escalate to Network Operators

Precise signal of impact is important – taking the wrong action can be worse than taking no action

- Taking a congested device out of service can make congestion worse

Automated network operations

Foundations

A. Network architecture that supports automated operations

B. Accurate signal of impact which indicates cause

C. Small number of auto-mitigation actions

D. Systems to safely apply those actions

Problem statement

How can we measure packet loss …

- … with sufficient accuracy that we can detect anomalies (even low level)
- … and sufficient context that we can apply appropriate auto-mitigation actions
  - … which device?
  - … what’s the cause?
- … at scale

Not all packet loss is equal - what’s anomalous?

Whether packet loss is a problem and what actions should be taken as a result, depends on four features:

- Magnitude
- Location
- Duration
- Cause
- (Context)

Packet Loss Monitoring Options

Active Monitoring:

+/- Magnitude: accuracy limited by sampling in time and space

+ Location: can indicate discarding device with triangulation

+ Duration

– Cause: does not indicate root cause

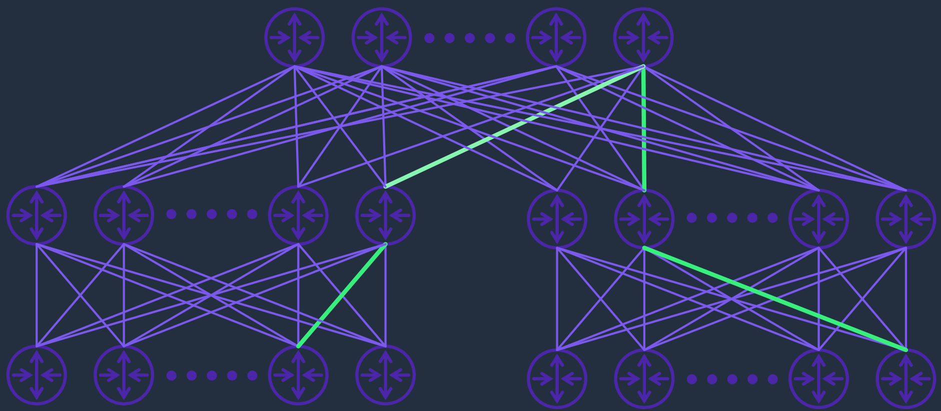
Active monitoring trade off - coverage vs. fidelity vs. probe volume (pick 2 out of 3)

Example: assuming 3-tier clos fabric

- 4 links from host to tier1
- 32 from tier1 to tier2
- 8 from tier2 to tier3
- 1,048,576 possible link-path combinations within this fabric

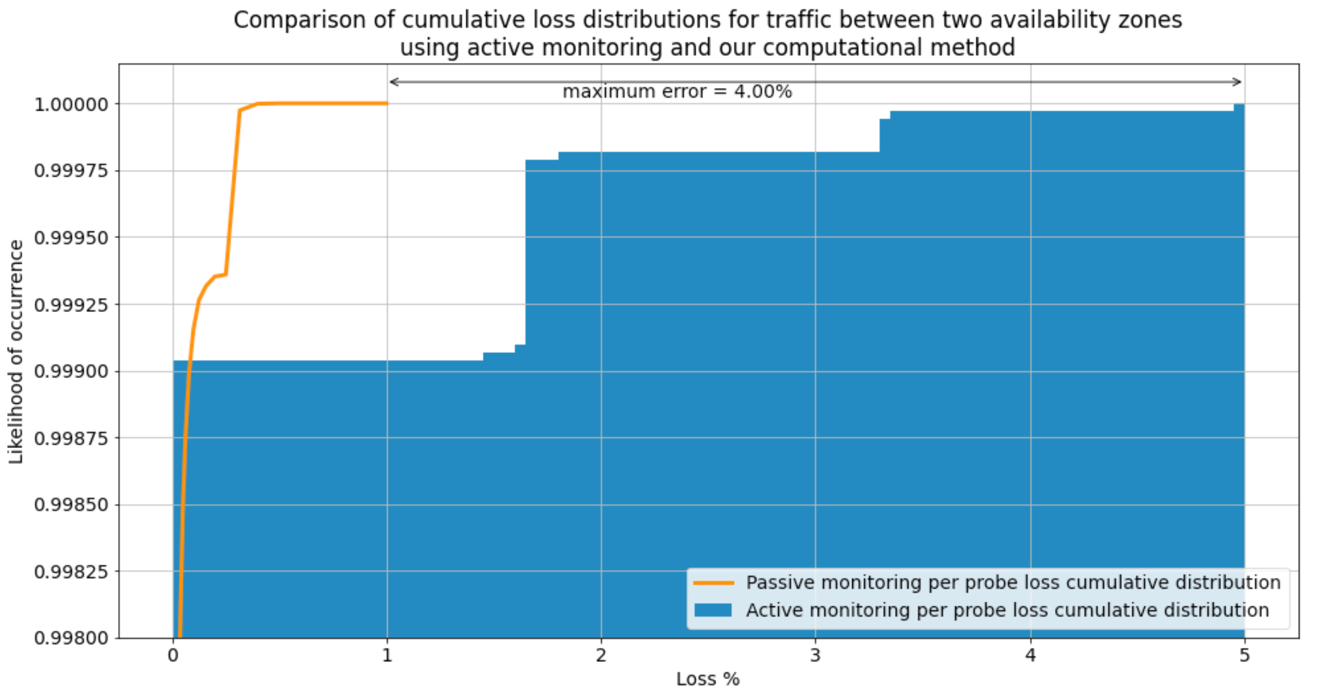
Number of link-path combinations can be much larger in practice:

- Intra-region 2.9E87
- Inter-region 4E176



Cloud network paths are both long and wide

Active monitoring trade off - coverage vs. fidelity vs. probe volume (pick 2 out of 3)



The active monitoring system overestimates the maximum loss by about 4% compared to passive loss measurement

Packet Loss Monitoring Options

Active Monitoring:

+/- Magnitude: accuracy limited by sampling in time and space

+ Location: can indicate discarding device with triangulation

+ Duration

– Cause: does not indicate root cause

Passive Monitoring:

+/- Magnitude: unsampled in time and space – as accurate as device loss reporting is

+ Location: directly indicates the dropping device

+ Duration

– Cause: classification is inconsistent and not useful to root cause

MIB-II (RFC1213, 1991)

ifInDiscards

“The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.”

ifInErrors

“The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.”

Implementation Inconsistency

All vendors support more discard metrics than this – but they are inconsistently implemented

Experience across multiple vendors and hardware platforms:

- Not reporting all discards – appears like a grey failure
- Duplication across discard metrics
- Same OID can account for different discards are different platforms
- ifInErrors can include non-discarded “errors” and discarded errors
- Interface metrics vs. platform metrics

There are no clearly defined semantics for packet loss reporting

Experience defining a new packet discard classification scheme

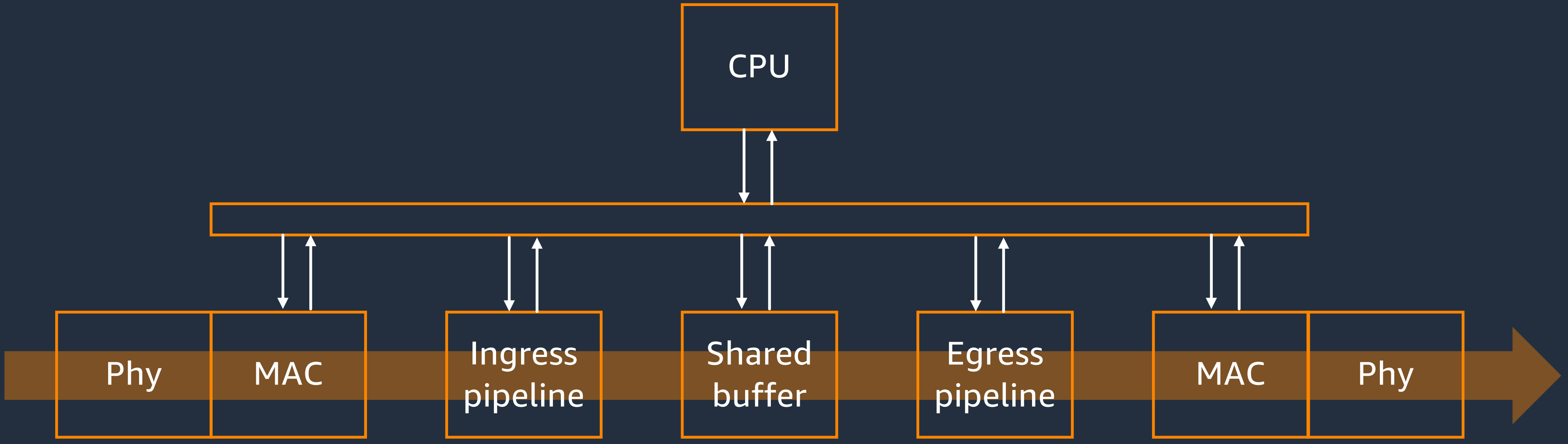
- We defined discard classes working backwards from auto-remediation
- Defined discard semantics
- Mapped the underlying hardware drop counters to the discard classes

Across multiple hardware platforms

From 64 to 256 underlaying hardware drop counters, depending on platform

https://datatracker.ietf.org/doc/draft-evans-discardclass/

Where and why do packets get dropped?



CPU

Phy

MAC

Ingress pipeline

Shared buffer

Egress pipeline

MAC

Phy

ErrorRxL2

ACL
Policer
NullRoute

ErrorRxL3
ErrorLocal
NoRoute
ttl

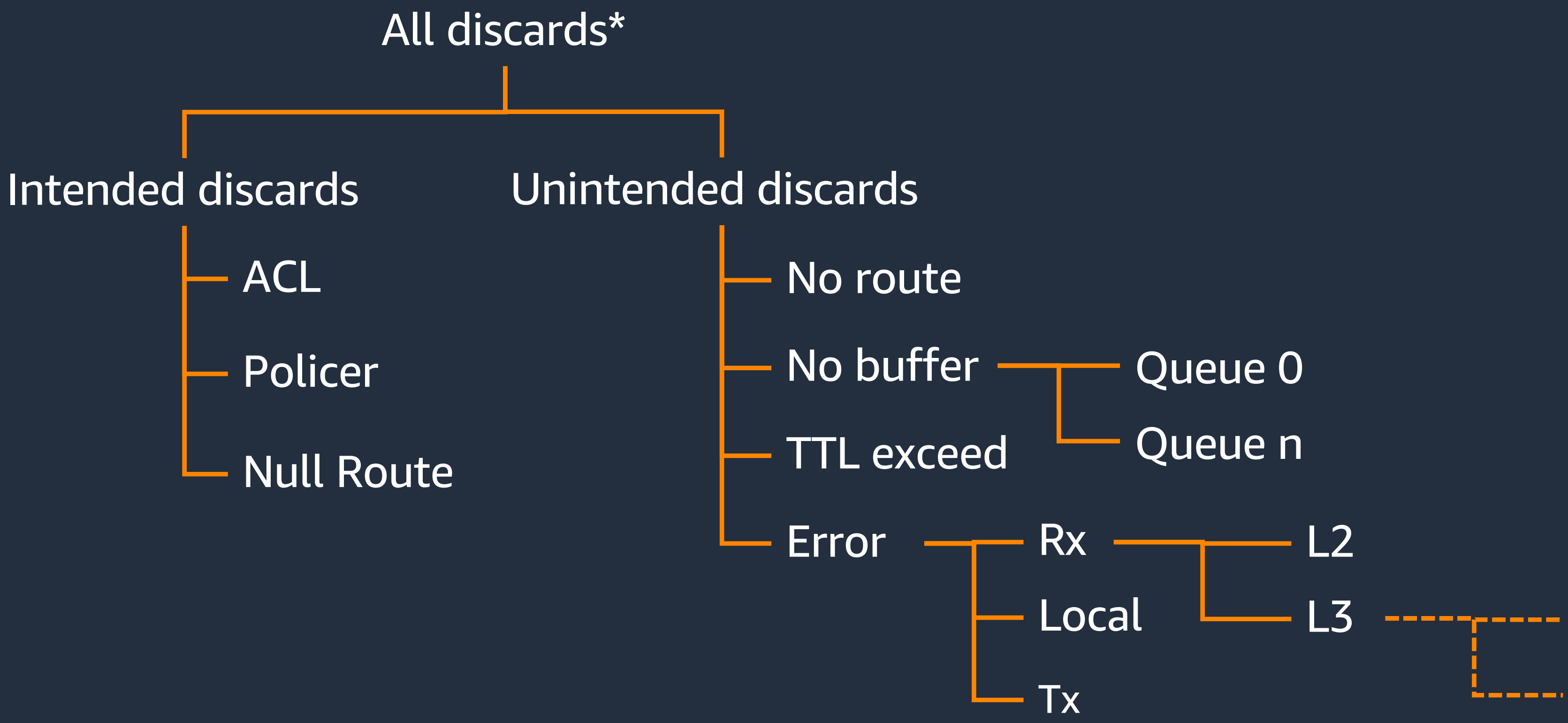
NoBuffer

ACL
Policer

ErrorTxL3

NO PACKET LEFT BEHIND

Minimal viable discard taxonomy for auto-mitigation … ymmv



All discards*

- Intended discards
  - ACL
  - Policer
  - Null Route
- Unintended discards
  - No route
  - No buffer
    - Queue 0
    - Queue n
  - TTL exceed
  - Error
    - Rx
      - L2
      - L3
    - Local
    - Tx

* Also need packets sent

Semantics Matter

TLDR:

Report all packet drops …

… once and only once …

… where they occur …

… in the right class

Long version:

https://datatracker.ietf.org/doc/draft-evans-discardclass/

Implementation experience

- Number of discard classes is a compromise
  - Enough granularity to take the right action
  - Too much information – can slow down resolution rather than help to surface the problem quickly
  - Volume of data for per interface metrics
- Null route vs. no route discards
- To CPU ACL vs. transit ACL discards
- Responded TTL expired vs total TTL expired
- Cannot detect config error without additional context

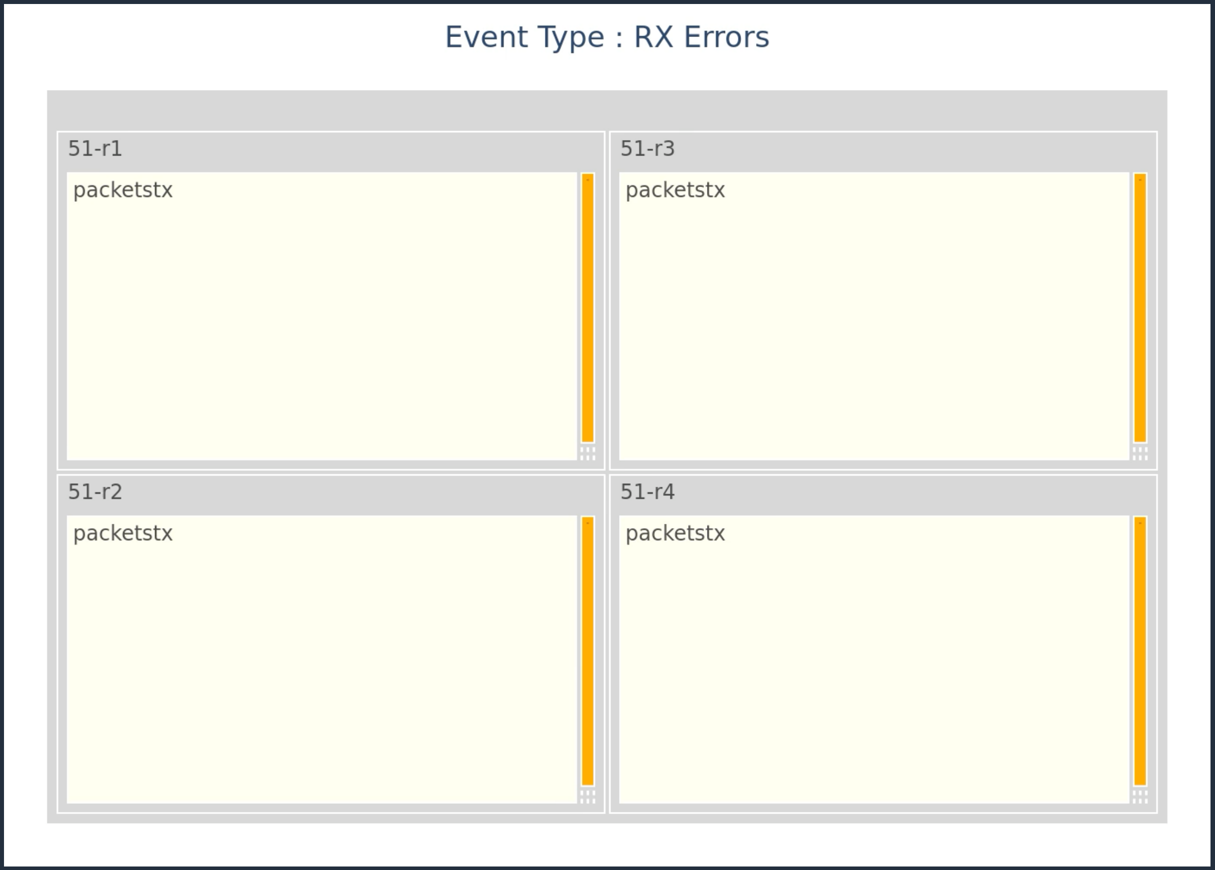
Cannot detect config error without additional context

e.g. How can we determine if ACL discards are intended or due to a misconfigured ACL?

- i.e. when intended loss becomes unintended loss
- We can’t tell that from device level metrics alone
- This needs additional context and other detection methods
    - e.g. with config validation before deployment
    - or detecting a significant change in ACL discards before/after a change

NO PACKET LEFT BEHIND

What’s anomalous? Signatures of impact

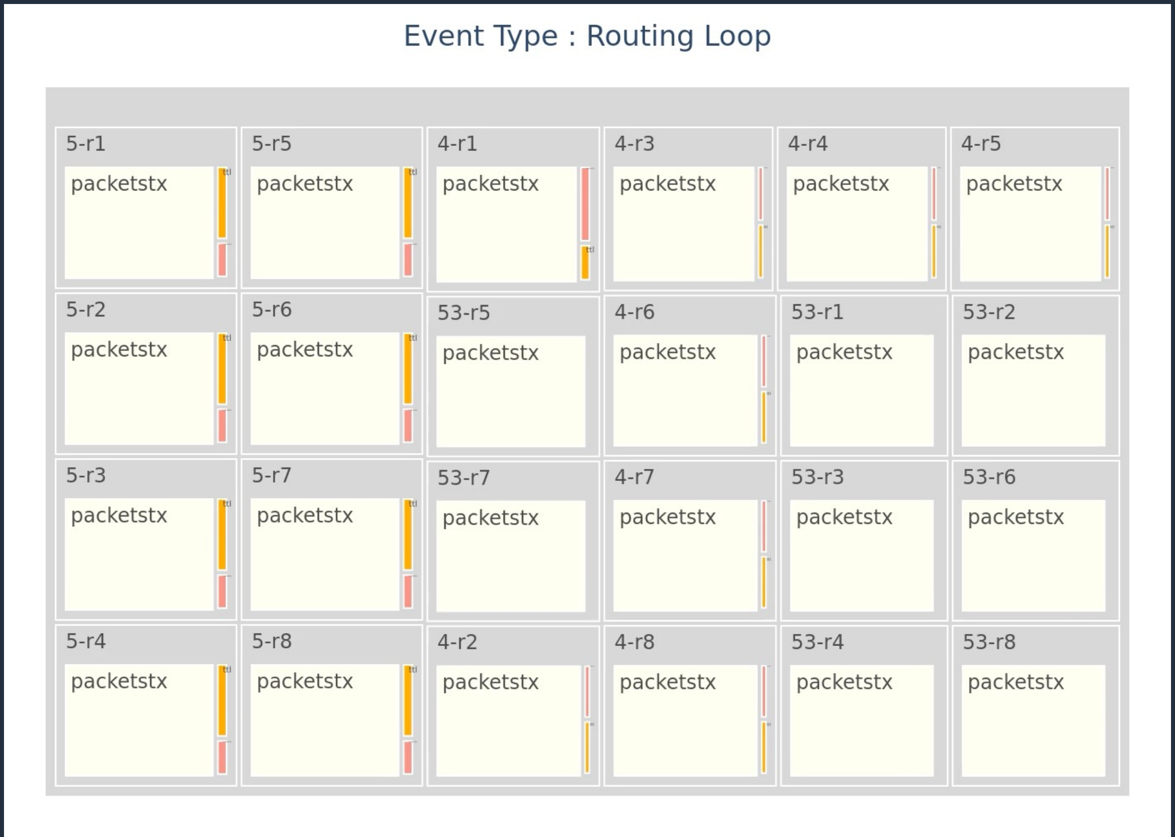


Event Type : RX Errors

Legend

- ErrorRx
- ErrorLocal
- ttl
- Null route

What’s anomalous? Signatures of impact



Event Type : Routing Loop

5-r1 packetstx	5-r5 packetstx	4-r1 packetstx	4-r3 packetstx	4-r4 packetstx	4-r5 packetstx
5-r2 packetstx	5-r6 packetstx	53-r5 packetstx	4-r6 packetstx	53-r1 packetstx	53-r2 packetstx
5-r3 packetstx	5-r7 packetstx	53-r7 packetstx	4-r7 packetstx	53-r3 packetstx	53-r6 packetstx
5-r4 packetstx	5-r8 packetstx	4-r2 packetstx	4-r8 packetstx	53-r4 packetstx	53-r8 packetstx

Legend

- ErrorRx
- ErrorLocal
- ttl
- Null route

Reason → Cause → Action mappings

Drop reason	Direction	Drop Cause	Loss rate	Loss duration	Customer impacting?	Possible actions
ErrorRxL2Discards	Ingress	Upstream device or link errror	>0(Anomaly)	O(1min)	Y	Take upstream link or device out-of-service
TTLDiscards	Ingress	Tracert	<=Baseline		N	no action
TTLDiscards	Ingress	Convergence	>Baseline	O(1s)	Y	no action
TTLDiscards	Ingress	Routing loop	>Baseline	O(1min)	Y	Roll-back
…	…	…	…	…	…	…

Packet Loss Monitoring Options

Active Monitoring:

– **Magnitude:** accuracy limited by sampling in time and space

+ **Location:** can indicate discarding device with triangulation

+ **Duration**

– **Cause:** does not indicate root cause

Passive Monitoring:

+ **Magnitude:** unsampled in time and space ~~as accurate as device loss reporting is~~

+ **Location:** directly indicates the dropping device

+ **Duration**

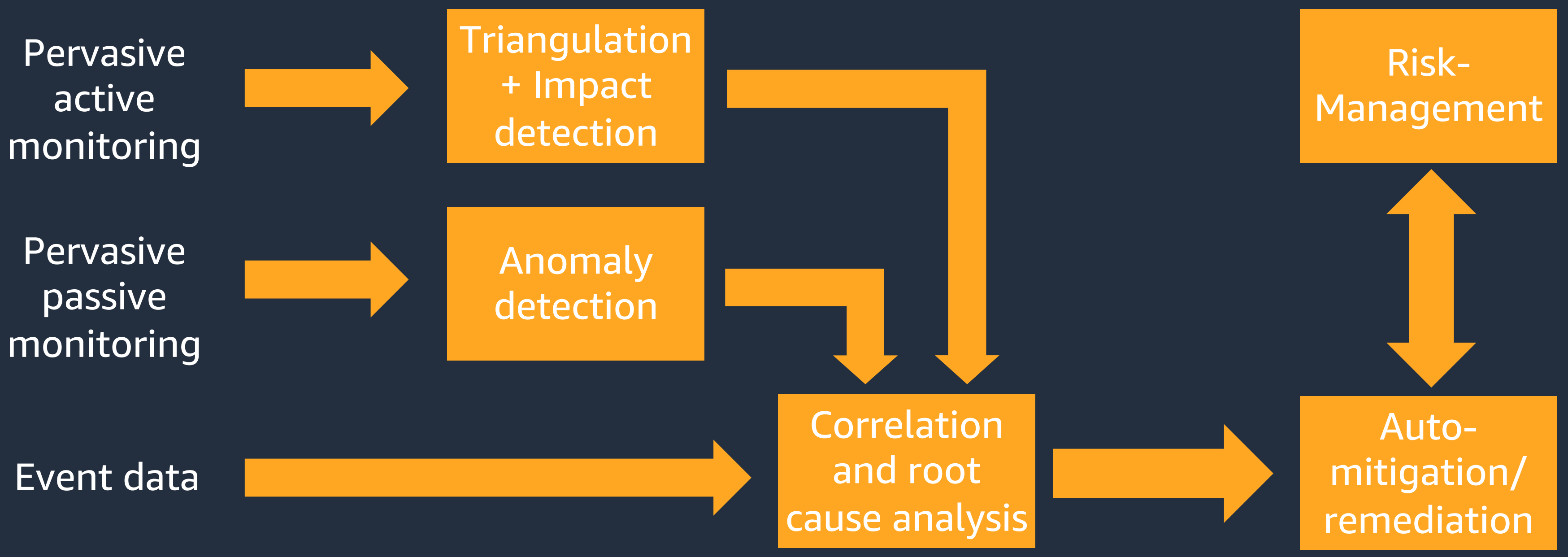
+ **Cause:** classification is ~~inconsistent and not~~ useful to root cause

Automated network operations

Foundations

1. Network architecture that supports automated operations
2. Accurate signal of impact which indicates cause
3. Small number of auto-mitigation actions
4. Systems to safely apply those actions

Systems to safely apply those actions


Pervasive active monitoring
Triangulation + Impact detection
Pervasive passive monitoring
Anomaly detection
Event data
Correlation and root cause analysis
Risk-Management
Auto-mitigation/ remediation

No packet left behind

To fully benefit from automated network operations – need to design the network for it

- A driver for Clos fabric architectures both in the DC and the wide area

Need precise signals – fixing the data drives better outcomes than inference from low quality data

Feedback welcomed:

https://datatracker.ietf.org/doc/draft-evans-discardclass/

Thank you!

John Evans Fabien Chraim

NANOG 88, 13 June 2023