More data is good, but good data is better.

Using the router control plane traffic as the foundation an open source vendor agnostic network analytics eco-system.

Brian Field

brian@opensrcanalytics.com

Bio

- 20+ years in the cable service provider space, PhD in CS, 18+ patents
- Network architect for:
  - the convergence of voice, unicast video (VOD) and internet onto the same/single IP infrastructure.
  - laying multicast linear video (TV channels) onto the this "converged" network. Corresponding ops tool development
  - initial commercial services overlay to the converged network
- CDN, network virtualization, streaming telemetry, analytics.
- Focus now— applying above domain knowledge into the analytic space

Problem

- Undetected Network issues:
  - Existing network operational data didn't provide insight as to existence, location, or scope of problem caused by the network
- Examples:
  - Forwarding loops
  - ACL causing legitimate service packets to be dropped
  - Unexpected null route exposure resulting in service black holing
  - Data center v6 off-net issues due to accidental rogue RAs
  - Inconsistent MTUs

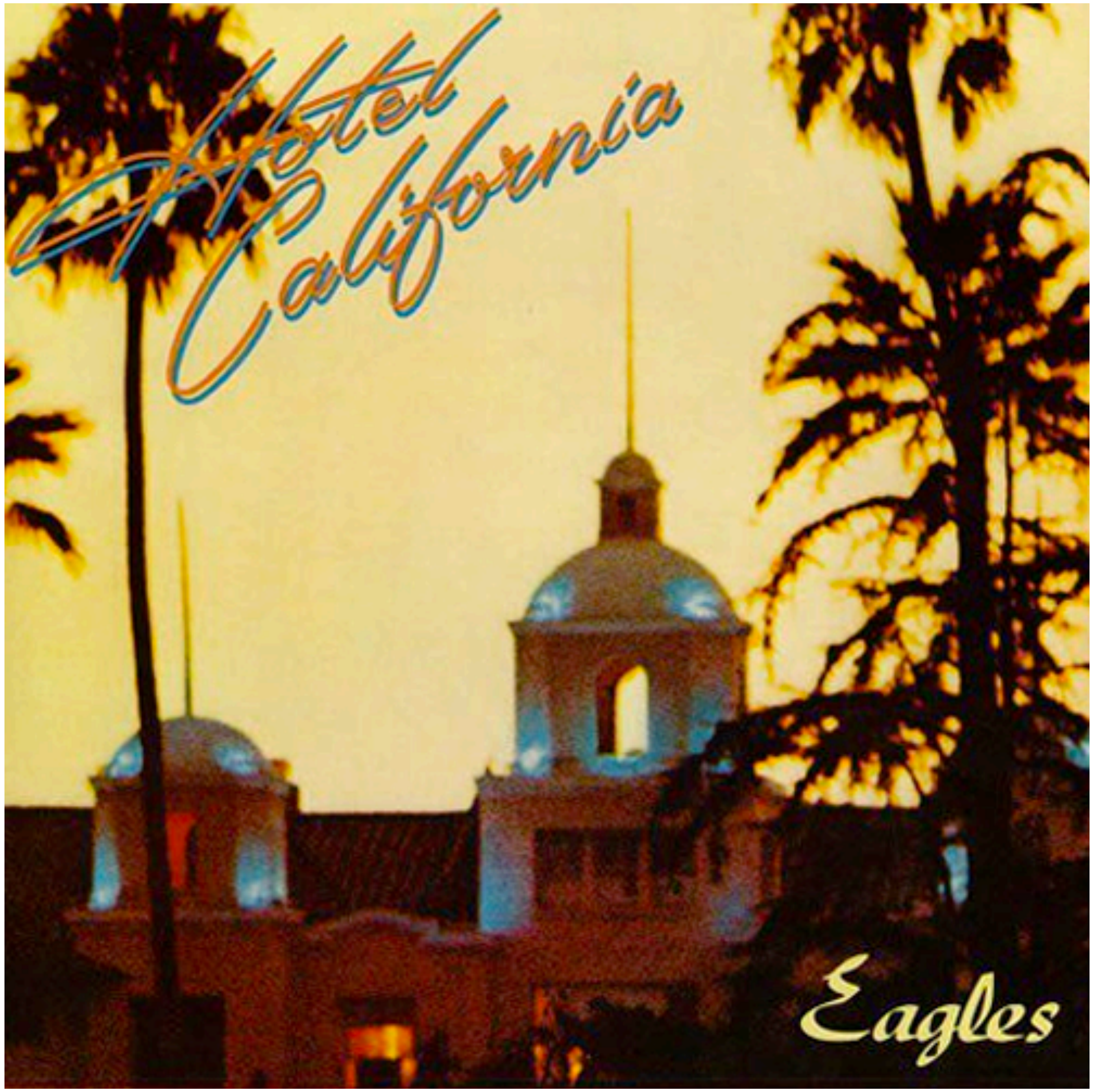
What is the existing network operational data?

- Show command, syslog, snmp, streaming telemetry models, open config

- The same data we've been getting last 20 years

- Low fidelity. Haystack.

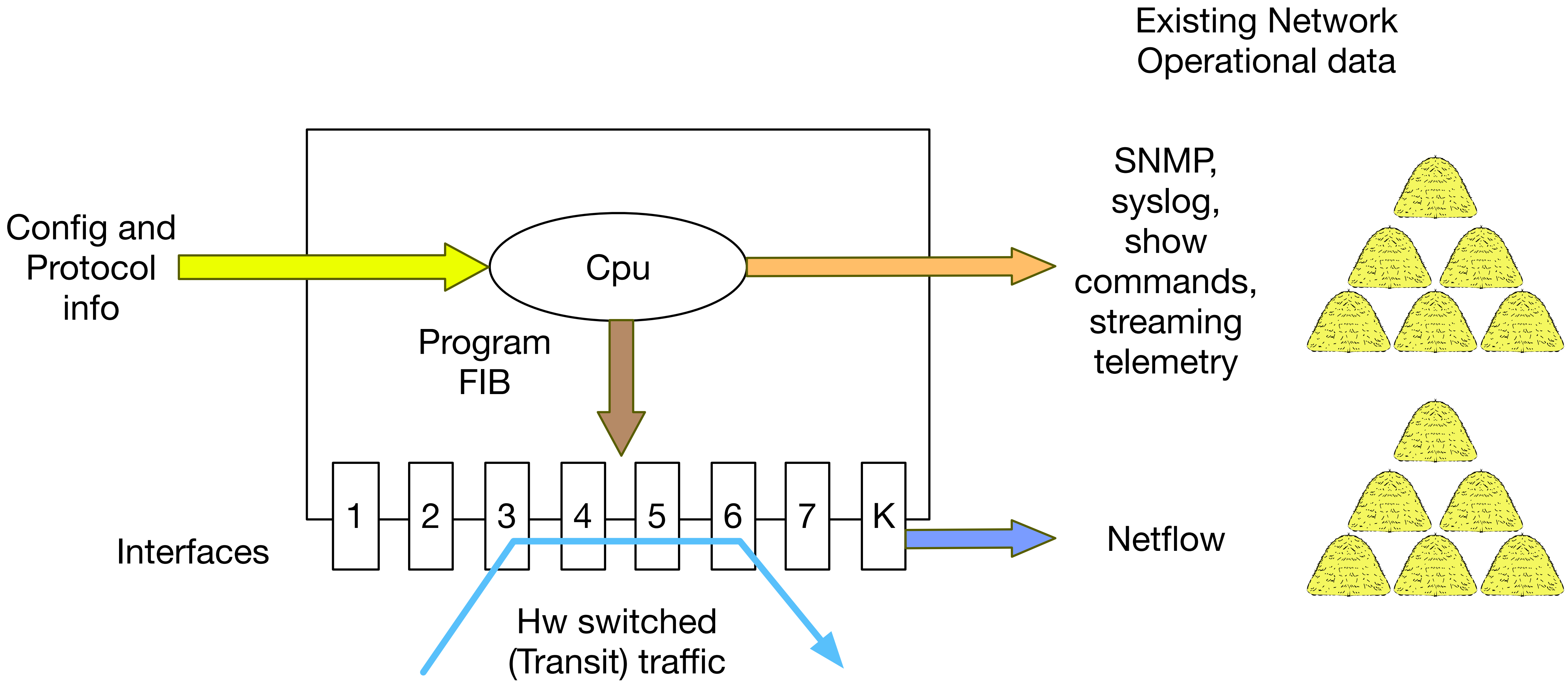
- Streaming is more efficient. More data. More haystacks.



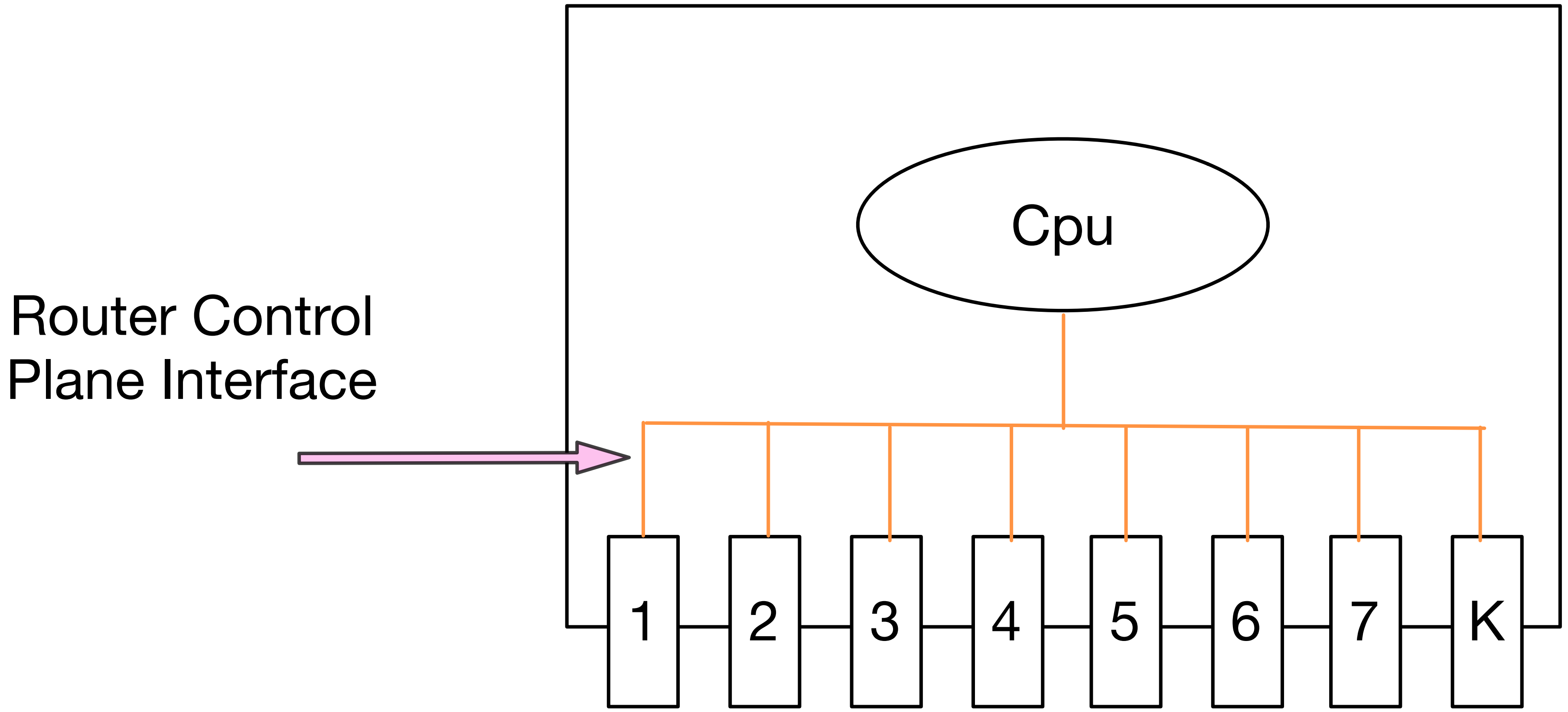
ML/AI to the rescue?

- Lots of energy and some hype.
- Basic principle to ML— no matter how great the ML algorithm or how much compute, if you're feeding it the wrong data, there's not much value.
- ML/AI has its place but its not magic pixie dust.
- Is the existing network operational data "the right data"?
- Is there other, higher fidelity network operational data we are overlooking?

How does a router work?

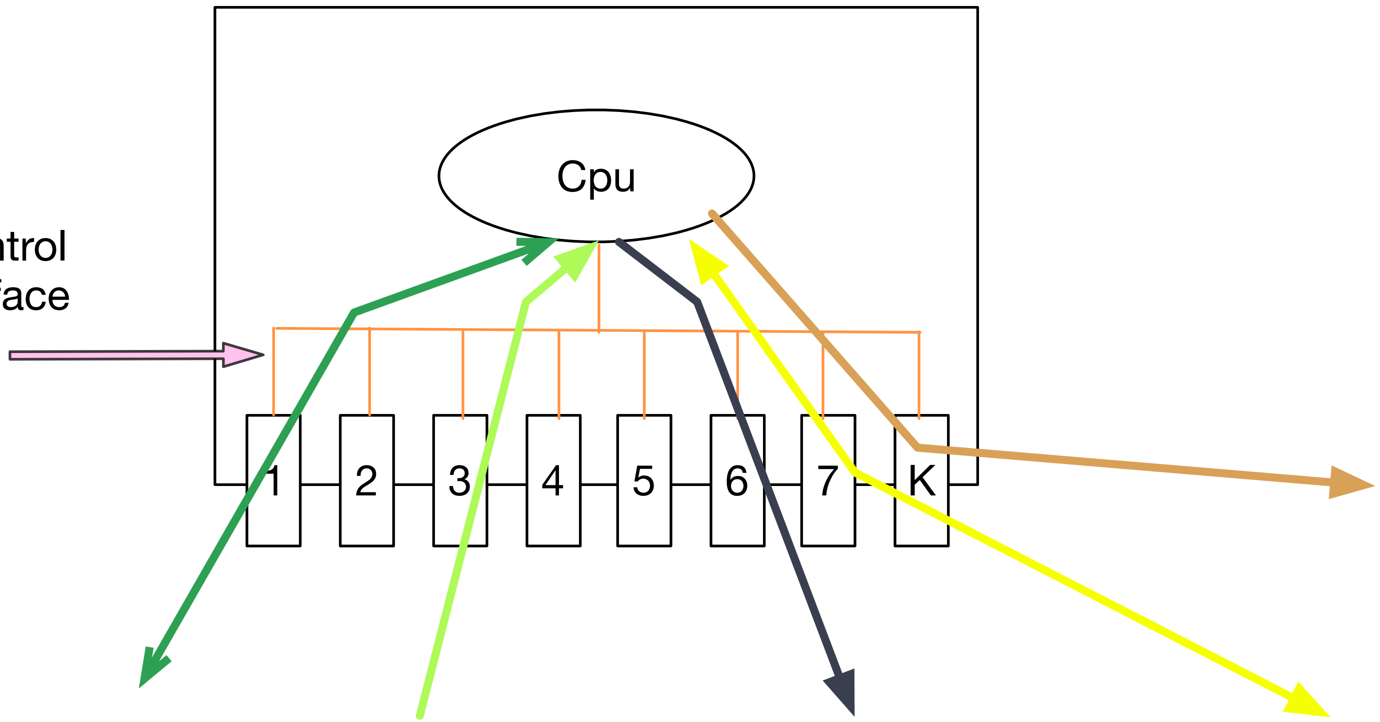


Yes, the router's internal control plane interface



What flows over this interface?

Router Control Plane Interface



Existing Operational traffic
Syslog, snmp, show commands, streaming telemetry

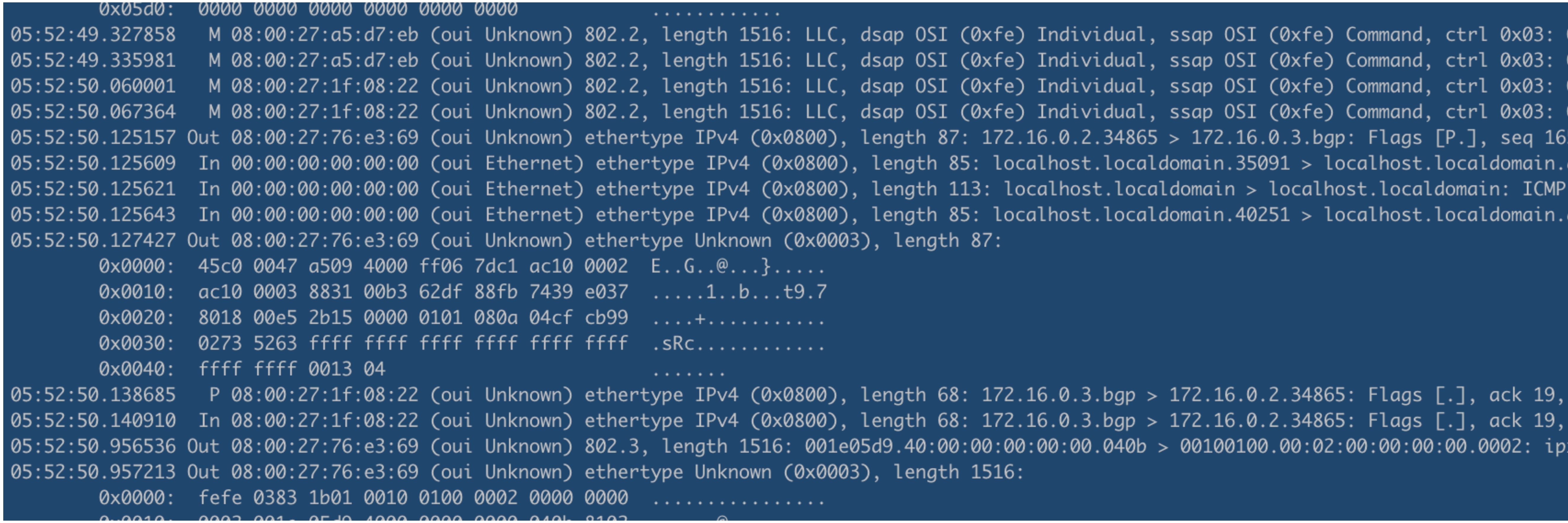
Routing traffic
Bgp, isis, ospf, lldp, ldp, pim, igmp, icmp, etc.

Punt Traffic
Ping to router, ttl=1, acl drop, no route drop, multicast issues, MTU issues

Negative Traffic
Icmp responses (ping response, TTL expired, admin denied, no route, MTU issue, etc.

Management traffic
Ssh, grpc, etc.

Example of router control plane data



```
        0x05d0:  0000 0000 0000 0000 0000 0000          ............
05:52:49.327858   M 08:00:27:a5:d7:eb (oui Unknown) 802.2, length 1516: LLC, dsap OSI (0xfe) Individual, ssap OSI (0xfe) Command, ctrl 0x03: (
05:52:49.335981   M 08:00:27:a5:d7:eb (oui Unknown) 802.2, length 1516: LLC, dsap OSI (0xfe) Individual, ssap OSI (0xfe) Command, ctrl 0x03: (
05:52:50.060001   M 08:00:27:1f:08:22 (oui Unknown) 802.2, length 1516: LLC, dsap OSI (0xfe) Individual, ssap OSI (0xfe) Command, ctrl 0x03: (
05:52:50.067364   M 08:00:27:1f:08:22 (oui Unknown) 802.2, length 1516: LLC, dsap OSI (0xfe) Individual, ssap OSI (0xfe) Command, ctrl 0x03: (
05:52:50.125157 Out 08:00:27:76:e3:69 (oui Unknown) ethertype IPv4 (0x0800), length 87: 172.16.0.2.34865 > 172.16.0.3.bgp: Flags [P.], seq 16
05:52:50.125609  In 00:00:00:00:00:00 (oui Ethernet) ethertype IPv4 (0x0800), length 85: localhost.localdomain.35091 > localhost.localdomain.
05:52:50.125621  In 00:00:00:00:00:00 (oui Ethernet) ethertype IPv4 (0x0800), length 113: localhost.localdomain > localhost.localdomain: ICMP
05:52:50.125643  In 00:00:00:00:00:00 (oui Ethernet) ethertype IPv4 (0x0800), length 85: localhost.localdomain.40251 > localhost.localdomain.
05:52:50.127427 Out 08:00:27:76:e3:69 (oui Unknown) ethertype Unknown (0x0003), length 87:
        0x0000:  45c0 0047 a509 4000 ff06 7dc1 ac10 0002  E..G..@...}.....
        0x0010:  ac10 0003 8831 00b3 62df 88fb 7439 e037  .....1..b...t9.7
        0x0020:  8018 00e5 2b15 0000 0101 080a 04cf cb99  ....+...........
        0x0030:  0273 5263 ffff ffff ffff ffff ffff ffff  .sRc............
        0x0040:  ffff ffff 0013 04                        .......
05:52:50.138685   P 08:00:27:1f:08:22 (oui Unknown) ethertype IPv4 (0x0800), length 68: 172.16.0.3.bgp > 172.16.0.2.34865: Flags [.], ack 19,
05:52:50.140910  In 08:00:27:1f:08:22 (oui Unknown) ethertype IPv4 (0x0800), length 68: 172.16.0.3.bgp > 172.16.0.2.34865: Flags [.], ack 19,
05:52:50.956536 Out 08:00:27:76:e3:69 (oui Unknown) 802.3, length 1516: 001e05d9.40:00:00:00:00:00.040b > 00100100.00:02:00:00:00:00.0002: ip
05:52:50.957213 Out 08:00:27:76:e3:69 (oui Unknown) ethertype Unknown (0x0003), length 1516:
        0x0000:  fefe 0383 1b01 0010 0100 0002 0000 0000  ................
        0x0010:  0003 001e 05d9 4000 0000 0000 040b 8103  ......@.........
```

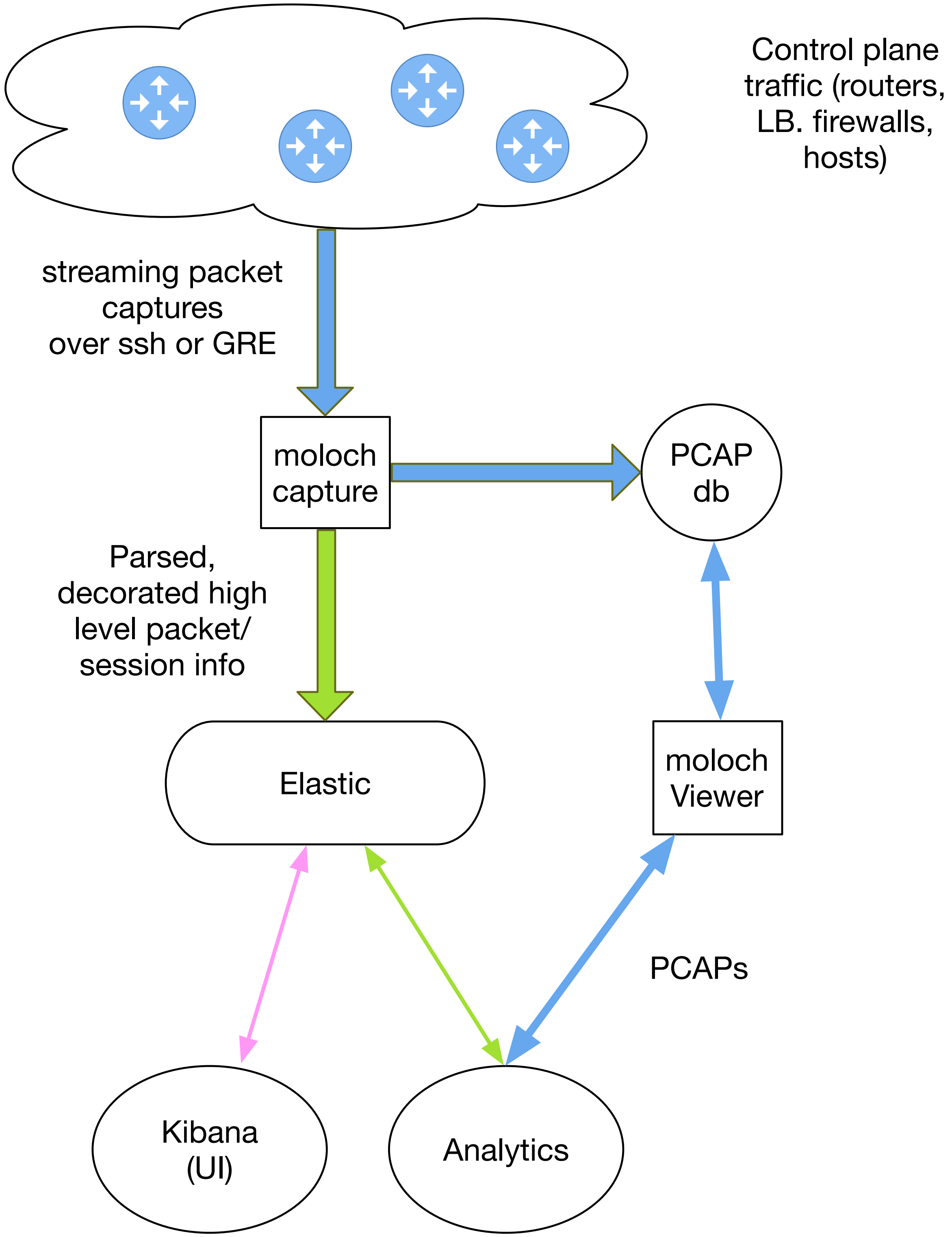
Control plane data dimensionality

- Inter-arrival of packets within a stream
- Inter-arrival of packets across all streams
- Src IP
- Dst IP
- Payload contents.

Wouldn’t it be great if there was an..

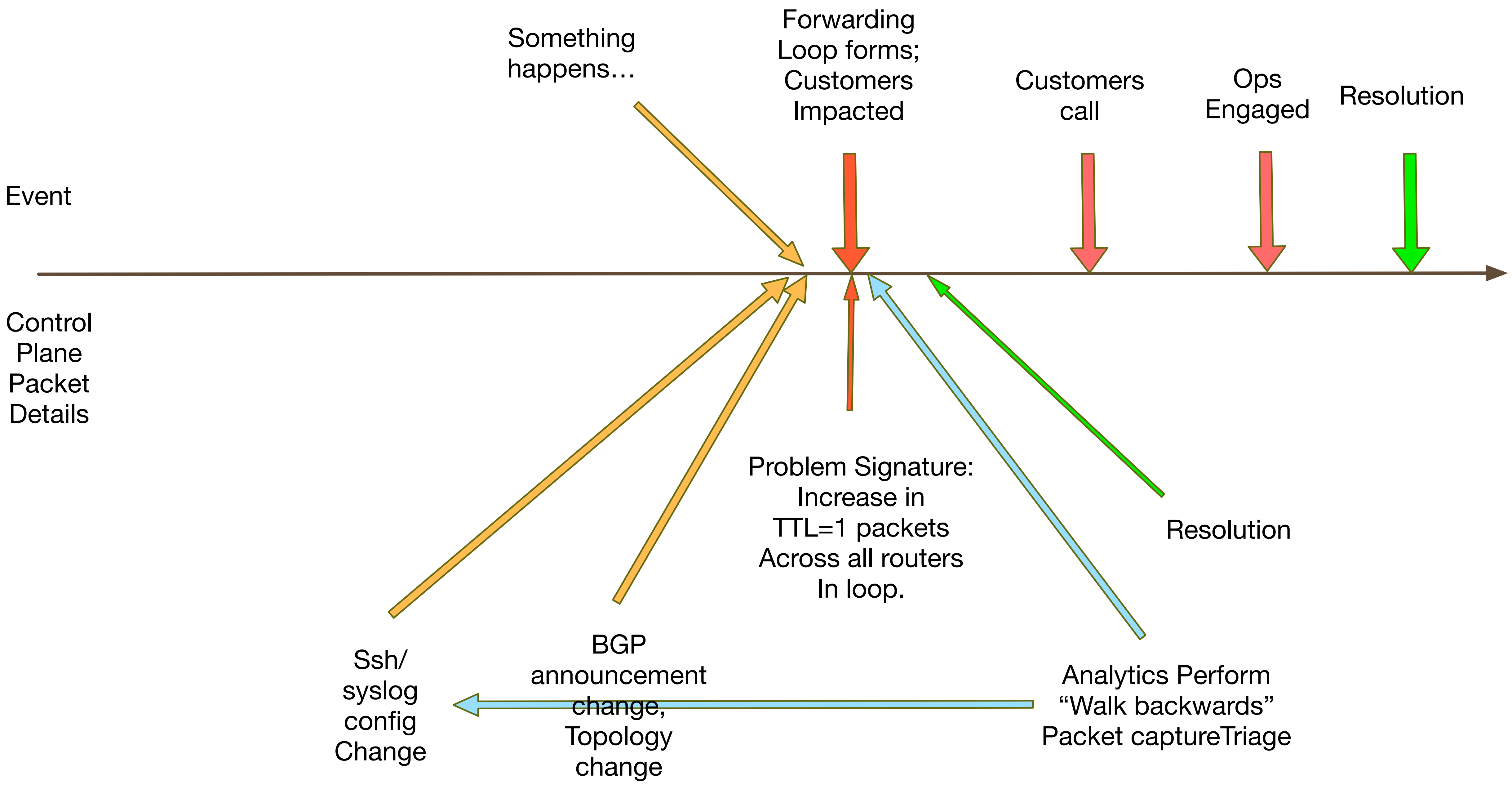
- open source
- scaleable
- streaming packet capture
- parsing, decorating,
- and storage platform?

High-level packet capture streaming and analytic pipeline (and it's vendor agnostic!)



Control plane traffic (routers, LB. firewalls, hosts)
streaming packet captures over ssh or GRE
moloch capture
PCAP db
Parsed, decorated high level packet/ session info
Elastic
moloch Viewer
PCAPs
Kibana (UI)
Analytics

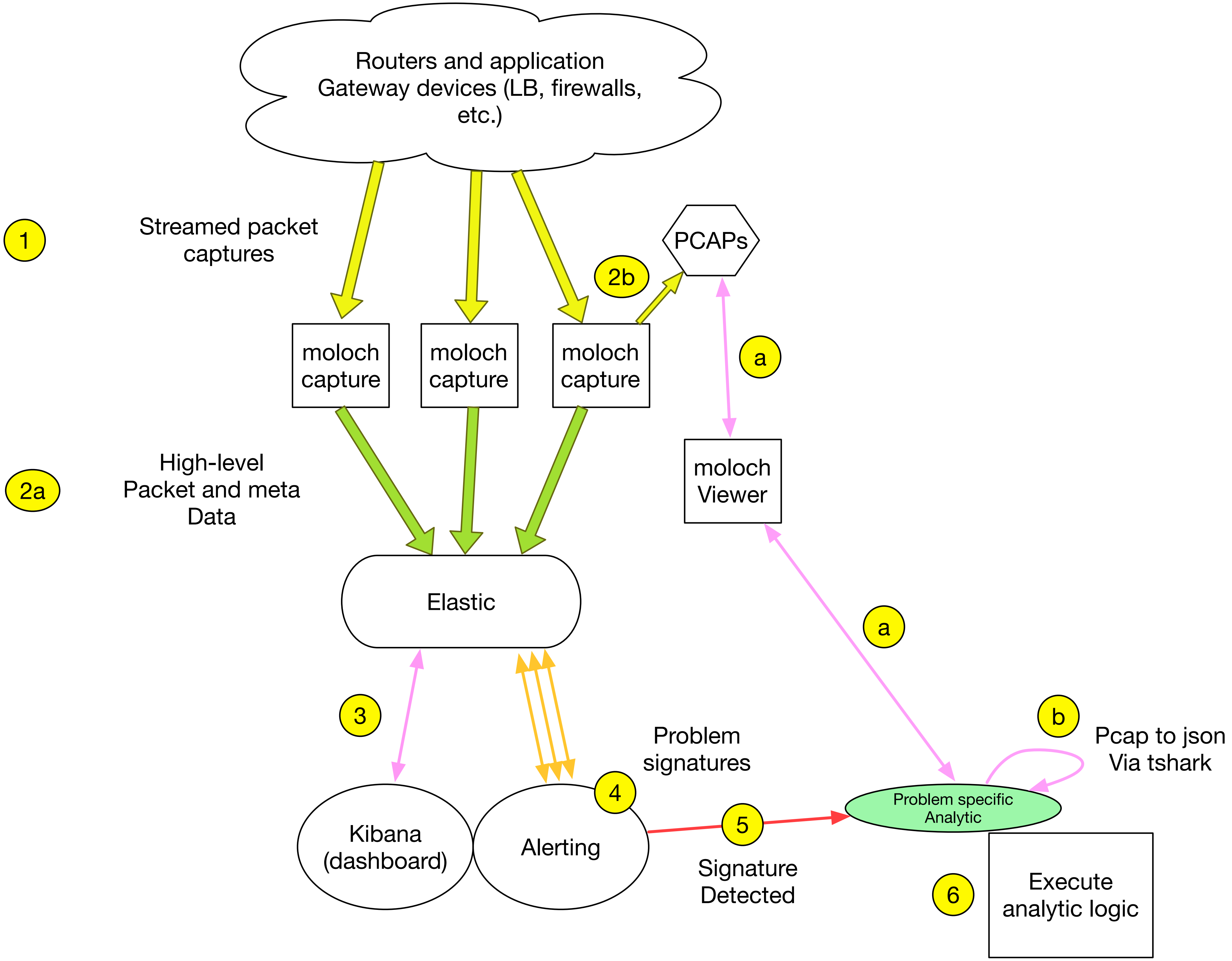
High-level analytic use case


Something happens…
Forwarding Loop forms; Customers Impacted
Customers call
Ops Engaged
Resolution
Event
Control Plane Packet Details
Problem Signature: Increase in TTL=1 packets Across all routers In loop.
Resolution
Ssh/ syslog config Change
BGP announcement change, Topology change
Analytics Perform "Walk backwards" Packet captureTriage

Network data and analytics requirements on moloch

- Needs:
  - Support for ISIS and LLDP
  - real-time
  - correlate across “streams” (BGP, ISIS, ICMP) per device
  - correlate across data from multiple devices
  - efficiently "walk backwards” through packet data.
  - “lingua analytics”.
- Moloch refinements:
  - Ethernet support
  - refined “session” model.

OpenSrcNetwork analytics eco-system



Routers and application Gateway devices (LB, firewalls, etc.)

1 Streamed packet captures

moloch capture

moloch capture

moloch capture

2b PCAPs

a

moloch Viewer

2a High-level Packet and meta Data

Elastic

a

3

b Pcap to json Via tshark

Problem signatures

4

Kibana (dashboard)

Alerting

5

Problem specific Analytic

Signature Detected

6 Execute analytic logic

Make router platforms simpler…

- Reduce platform feature needs
  - No need for BMP or BGP-LS
  - dumping/streaming FIB.

- Lower the bar

- Lots of opportunities here…

Summary

- Router control plane data is high fidelity compared to existing data
- Leverage moloch as packet capturing, parsing and decorating platform
- Build monitoring/alerting triggers off elastic
- Analytics “walk backwards” and pull PCAPs from moloch
- Analytics use Wireshark decode (structure and naming) as the “lingua analytics”

Thank you!