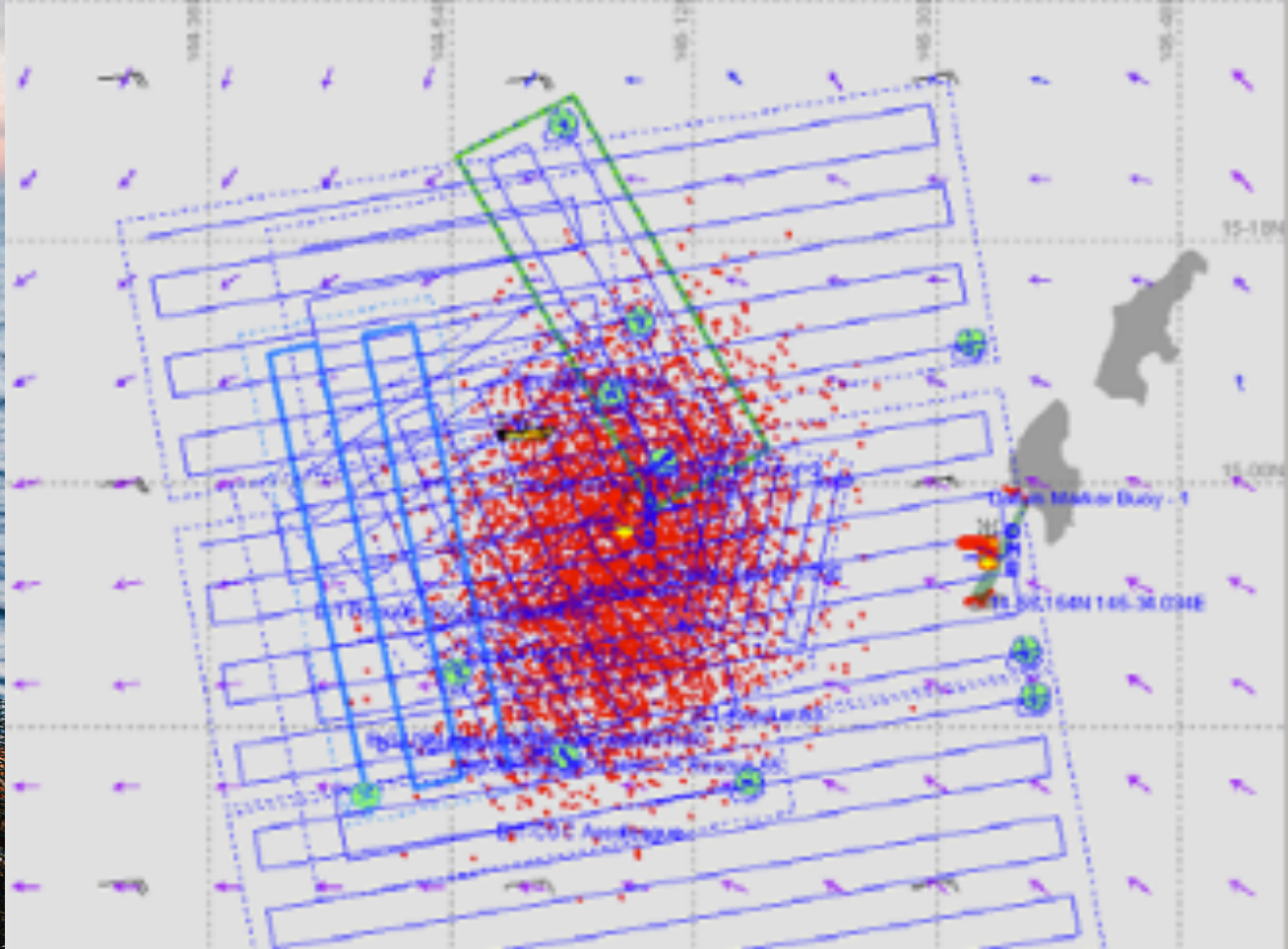


# Ten Lessons From Telemetry

Shelly Cadora, PhD

Cisco Systems



# It's All About Operations

- Precedent is well established

“Today, our time series metric ingestion service handles more than 2.8 billion write requests per minute, stores 4.5 petabytes of time series data, and handles 25,000 query requests per minute.”

<https://blog.twitter.com/2016/observability-at-twitter-technical-overview-part-i>

- Network monitoring has work to do.



# Where We Started (Late 2014)

- Maximize performance
  - Enable “push”
  - Leverage system’s natural order
- Normalize to existing toolchains
  - Flexible format
  - Flexible transport
- Learn Through Experimentation
  - Fail and evolve



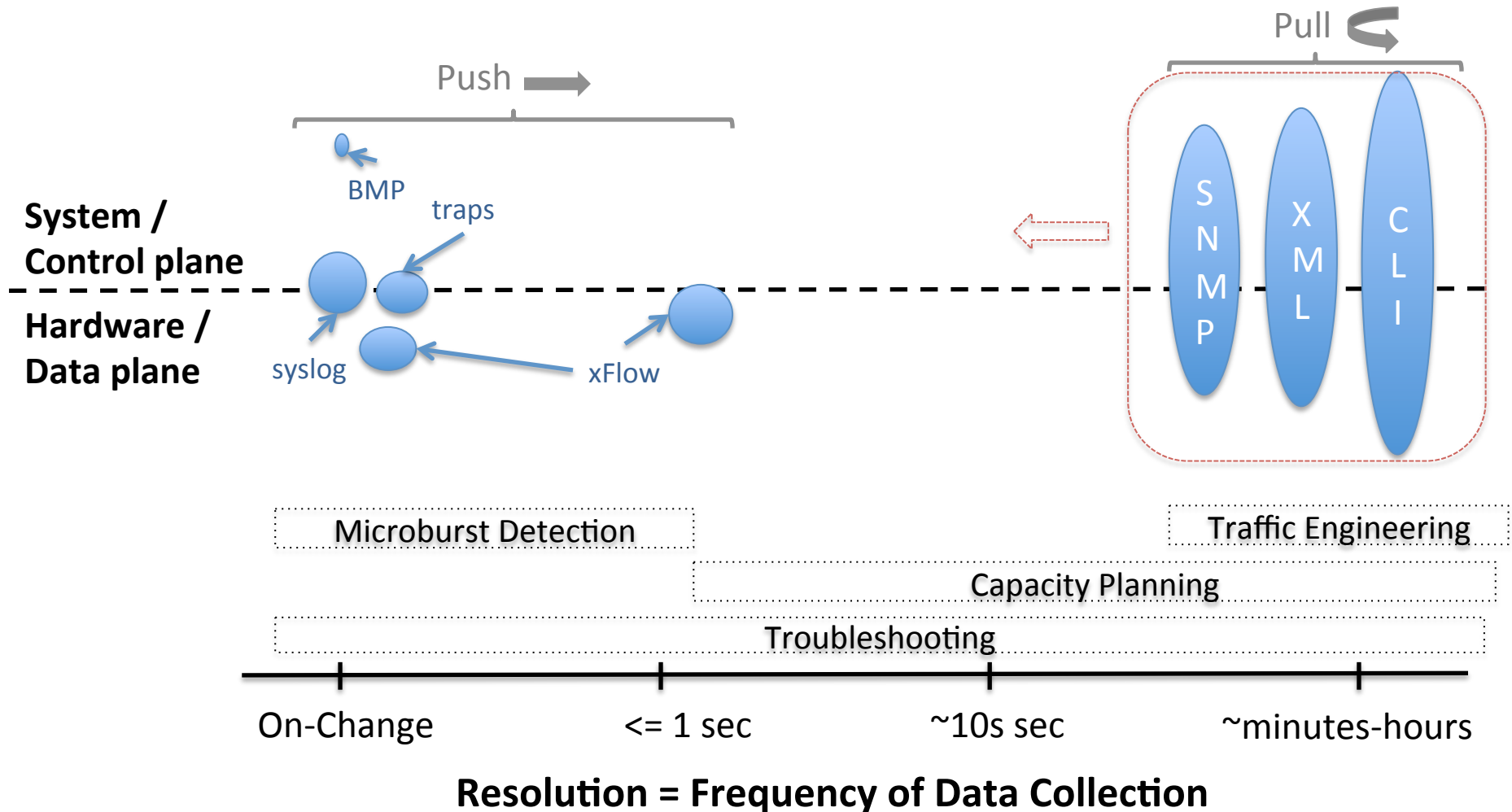
Free the Data

## Where We Are (2016)

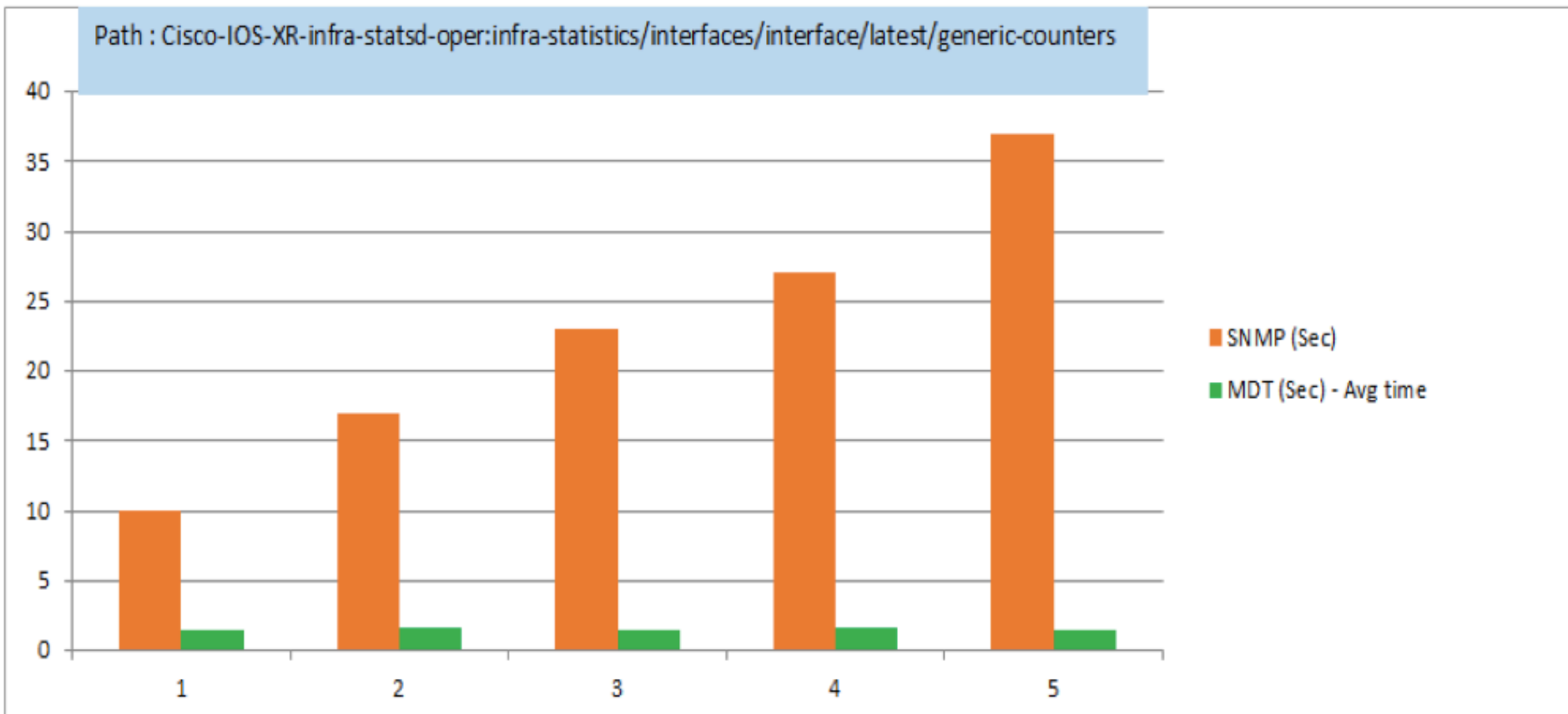
- First Production Deployment
- Telemetry authoritative over SNMP (30 sec)
- Extending coverage/performance/models



# Lesson 1: Telemetry Means Different Things to Different People



# Lesson 2: It's Not Hard to Beat SNMP



- 10 second poll / push
- 3 pollers / telemetry receivers
- 30 minute measurement intervals
- 288 100Gig E Interfaces (Line Rate)
- SNMP: IF-MIB (query by row)

# Head to Head on the Wire

## SNMP GetBulk IF-MIB

| No. | Time     | Source      | Destination | Protocol | Length |
|-----|----------|-------------|-------------|----------|--------|
| 12  | 1.979707 | 172.30.8.11 | 172.30.8.51 | SNMP     | 83     |
| 13  | 1.981296 | 172.30.8.51 | 172.30.8.11 | SNMP     | 1395   |
| 14  | 1.983784 | 172.30.8.11 | 172.30.8.51 | SNMP     | 86     |
| 15  | 2.004446 | 172.30.8.51 | 172.30.8.11 | SNMP     | 1381   |

▶ Frame 13: 1395 bytes on wire (11160 bits), 1395 bytes captured (11160 bits) on 0  
▶ Ethernet II, Src: Vmware\_9b:eb:e4 (00:0c:29:9b:eb:e4), Dst: Vmware\_af:e8:6b (00:0c:29:af:e8:6b)  
▶ Internet Protocol Version 4, Src: 172.30.8.51, Dst: 172.30.8.11  
▶ User Datagram Protocol, Src Port: 161 (161), Dst Port: 49179 (49179)  
▼ Simple Network Management Protocol  
    version: v2c (1)  
    community: cisco  
    ▶ data: get-response (2)

|      |                         |                         |                     |
|------|-------------------------|-------------------------|---------------------|
| 0050 | 2b 06 01 02 01 02 02 01 | 01 02 02 01 02 30 0f 06 | +.....0..           |
| 0060 | 0a 2b 06 01 02 01 02 02 | 01 01 03 02 01 03 30 0f | +......0..          |
| 0070 | 06 0a 2b 06 01 02 01 02 | 02 01 01 04 02 01 04 30 | ..+.....0           |
| 0080 | 0f 06 0a 2b 06 01 02 01 | 02 02 01 01 05 02 01 05 | ...+.....           |
| 0090 | 30 0f 06 0a 2b 06 01 02 | 01 02 02 01 01 06 02 01 | 0...+.....          |
| 00a0 | 06 30 0f 06 0a 2b 06 01 | 02 01 02 02 01 01 07 02 | .0...+.....         |
| 00b0 | 01 07 30 0f 06 0a 2b 06 | 01 02 01 02 02 01 01 08 | ..0...+.....        |
| 00c0 | 02 01 08 30 0f 06 0a 2b | 06 01 02 01 02 02 01 01 | ...0...+.....       |
| 00d0 | 09 02 01 09 30 13 06 0a | 2b 06 01 02 01 02 02 01 | ....0...+.....      |
| 00e0 | 02 02 04 05 4e 75 6c 6c | 30 30 21 06 0a 2b 06 01 | ....Null 00!...+    |
| 00f0 | 02 01 02 02 01 02 03 04 | 13 4d 67 6d 74 45 74 68 | .../RP0/CP U0/00\$. |
| 0100 | 30 2f 52 50 30 2f 43 50 | 55 30 2f 30 30 24 06 0a | 0/RP0/CP U0/00\$.   |
| 0110 | 2b 06 01 02 01 02 02 01 | 02 04 04 16 47 69 67 61 | +.....Giga          |
| 0120 | 62 69 74 45 74 68 65 72 | 6e 65 74 30 2f 30 2f 30 | bitEther net0/0/0   |
| 0130 | 2f 30 30 24 06 0a 2b 06 | 01 02 01 02 02 01 02 05 | /00\$.+.....        |
| 0140 | 04 16 47 69 67 61 62 69 | 74 45 74 68 65 72 6e 65 | ..GigabitEtherne    |
| 0150 | 74 30 2f 30 2f 30 2f 31 | 30 18 06 0a 2b 06 01 02 | t0/0/0/1 0...+...   |
| 0160 | 01 02 02 01 02 06 0a 4  | 74 75 6e 6e 65 6c 2d 74 | .....tunnel-t       |
| 0170 | 65 31 30 19 06 0a 2b 06 | 01 02 01 02 02 01 02 07 | e10...+.....        |
| 0180 | 04 0b 74 75 6e 6e 65 6c | 2d 74 65 31 30 30 17 06 | ..tunnel-te100..    |
| 0190 | 0a 2b 06 01 02 01 02 02 | 01 02 08 04 09 4c 6f 6f | +......Loo          |
| 01a0 | 70 62 61 63 6b 30 30 17 | 06 0a 2b 06 01 02 01 02 | pack00...+.....     |
| 01b0 | 02 01 02 09 04 09 4c 6f | 6f 70 62 61 63 6b 32 30 | .....Lo opback20    |
| 01c0 | 0f 06 0a 2b 06 01 02 01 | 02 02 01 03 02 02 01 01 | ...+.....           |
| 01d0 | 30 0f 06 0a 2b 06 01 02 | 01 02 02 01 03 03 02 01 | 0...+.....          |

~20 stats/interface

## Telemetry LatestCounters (Compact GPB)

| No. | Time     | Source      | Destination | Protocol | Length |
|-----|----------|-------------|-------------|----------|--------|
| 1   | 0.000000 | 172.30.8.51 | 172.30.8.4  | UDP      | 1016   |

▶ Frame 1: 1016 bytes on wire (8128 bits), 1016 bytes captured (8128 bits) on 0  
▶ Ethernet II, Src: Vmware\_9b:eb:e4 (00:0c:29:9b:eb:e4), Dst: Vmware\_af:6b:6b (00:0c:29:af:6b:6b)  
▶ Internet Protocol Version 4, Src: 172.30.8.51, Dst: 172.30.8.4  
▶ User Datagram Protocol, Src Port: 0 (0), Dst Port: 2105 (2105)  
▶ Data (974 bytes)

|      |                         |                         |                   |
|------|-------------------------|-------------------------|-------------------|
| 0110 | a8 05 ff ff ff ff 0f 12 | 91 01 0a 13 4d 67 6d 74 | .....Mgmt         |
| 0120 | 45 74 68 30 2f 52 50 30 | 2f 43 50 55 30 2f 30 90 | Eth0/RP0 /CPU0/0. |
| 0130 | 03 00 98 03 00 a0 03 00 | a8 03 00 b0 03 00 b8 03 | .....             |
| 0140 | 00 c0 03 00 c8 03 00 d0 | 03 00 d8 03 00 e0 03 00 | .....             |
| 0150 | e8 03 00 f0 03 00 f8 03 | 00 80 04 00 88 04 00 90 | .....             |
| 0160 | 04 00 98 04 00 a0 04 00 | a8 04 00 b0 04 00 b8 04 | .....             |
| 0170 | 00 c0 04 00 c8 04 00 d0 | 04 00 d8 04 00 e0 04 00 | .....             |
| 0180 | e8 04 00 f0 04 00 f8 04 | 00 80 05 00 88 05 cc 9e | .....             |
| 0190 | b7 ba 05 90 05 00 98 05 | 93 9a b7 ba 05 a0 05 ff | .....             |
| 01a0 | ff ff ff 0f a8 05 ff ff | ff ff 0f 12 94 01 0a 16 | .....             |
| 01b0 | 47 69 67 61 62 69 74 45 | 74 68 65 72 6e 65 74 30 | GigabitE thernet0 |
| 01c0 | 2f 30 2f 30 2f 30 90 03 | 9a 0b 98 03 e5 c8 18 a0 | /0/0/0.. ..       |
| 01d0 | 03 81 0a a8 03 c0 d8 0c | b0 03 e9 03 b8 03 00 c0 | .....             |
| 01e0 | 03 f3 06 c8 03 10 d0 03 | 00 d8 03 00 e0 03 00 e8 | .....             |
| 01f0 | 03 00 f0 03 00 f8 03 00 | 80 04 00 88 04 00 90 04 | .....             |
| 0200 | 00 98 04 00 a0 04 00 a8 | 04 00 b0 04 00 b8 04 00 | .....             |
| 0210 | c0 04 00 c8 04 00 d0 04 | 00 d8 04 00 e0 04 00 e8 | .....             |
| 0220 | 04 00 f0 04 00 f8 04 00 | 80 05 00 88 05 c9 9e b7 | .....             |
| 0230 | ba 05 90 05 00 98 05 d7 | 9a b7 ba 05 a0 05 00 a8 | .....G igabitEt   |
| 0240 | 05 00 12 94 01 0a 16 47 | 69 67 61 62 69 74 45 74 | .....G igabitEt   |
| 0250 | 68 65 72 6e 65 74 30 2f | 30 2f 30 2f 31 90 03 00 | hernet0/ 0/0/1... |
| 0260 | 98 03 00 a0 03 00 a8 03 | 00 b0 03 00 b8 03 00 c0 | .....             |
| 0270 | 03 00 c8 03 00 d0 03 00 | d8 03 00 e0 03 00 e8 03 | .....             |
| 0280 | 00 f0 03 00 f8 03 00 80 | 04 00 88 04 00 90 04 00 | .....             |
| 0290 | 98 04 00 a0 04 00 a8 04 | 00 b0 04 00 b8 04 00 c0 | .....             |
| 02a0 | 04 00 c8 04 00 d0 04 00 | d8 04 00 e0 04 00 e8 04 | .....             |
| 02b0 | 00 f0 04 00 f8 04 00 80 | 05 00 88 05 c9 9e b7 ba | .....             |
| 02c0 | 05 90 05 00 98 05 d7 9a | b7 ba 05 a0 05 ff ff ff | .....             |
| 02d0 | ff 0f a8 05 ff ff ff ff | 0f 12 8d 01 0a 0b 74 75 | .....tu           |
| 02e0 | 6e 6e 65 6c 2d 74 65 31 | 30 90 03 00 98 03 00 a0 | nnel-te1 0.....   |

~36 stats/interface

# Lesson 2 (cont): How To Beat SNMP

- Push Don't Pull
  - Avoid Request/Response Hits to CPU
  - Scrape the Hardware Once
  - Internal IPC Once
  - Many Receivers: Just Copy The Packet
- Loosen Restrictions on Index & Export
  - Get-Next Semantics are Expensive
  - Retrieve the Data In A Way that is Natural to the System
  - Even Pull Benefits From This



# Lesson 3: You Don't Know What You Love Until It's Gone

- Well-Known Name:

OID: 1.3.6.1.2.1.2.2.1.10.4

MIB: IF-MIB

Object: ifInOctets

In reality, not  
enough of these

- Not-So-Well-Known Name:

```
"RootOper.InfraStatistics.Interface({'InterfaceName'  
: 'GigabitEthernet0/0/0/0'}).GenericCounters",  
{'BytesReceived': 443140326}
```

# Lesson 4: Data Models are Good for Telemetry

- Follow the industry standard: YANG

## Vendor-Specific

```
module: Cisco-IOS-XR-infra-statsd-oper
  +--ro infra-statistics
    +--ro interfaces
      +--ro interface* [interface-name]
        +--ro latest
          | +--ro generic-counters
          |   +--ro packets-received? uint64
          |   +--ro bytes-received?  uint64
          |   +--ro packets-sent?    uint64
          |   +--ro bytes-sent?      uint64
          <snip>
```

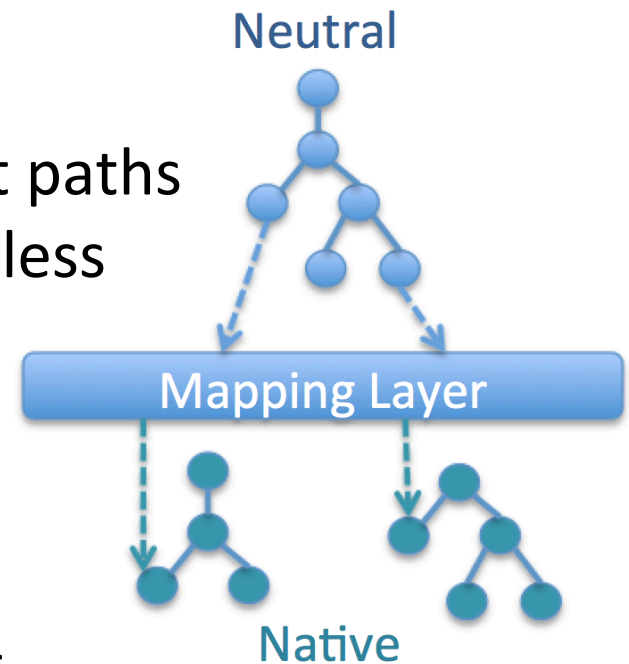
## Vendor-Neutral

```
module: openconfig-interfaces
  +--rw interfaces
    +--rw interface* [name]
      | +--ro counters
      |   +--ro in-octets?          yang:counter64
      |   +--ro in-unicast-pkts?   yang:counter64
      |   +--ro in-broadcast-pkts? yang:counter64
      |   +--ro in-multicast-pkts? yang:counter64
      |   +--ro in-discards?       yang:counter64
      <snip>
```

Benefits of YANG: Order by system, pull containers, readable, extensible, hierarchical, reusable, versioning, modularity, etc

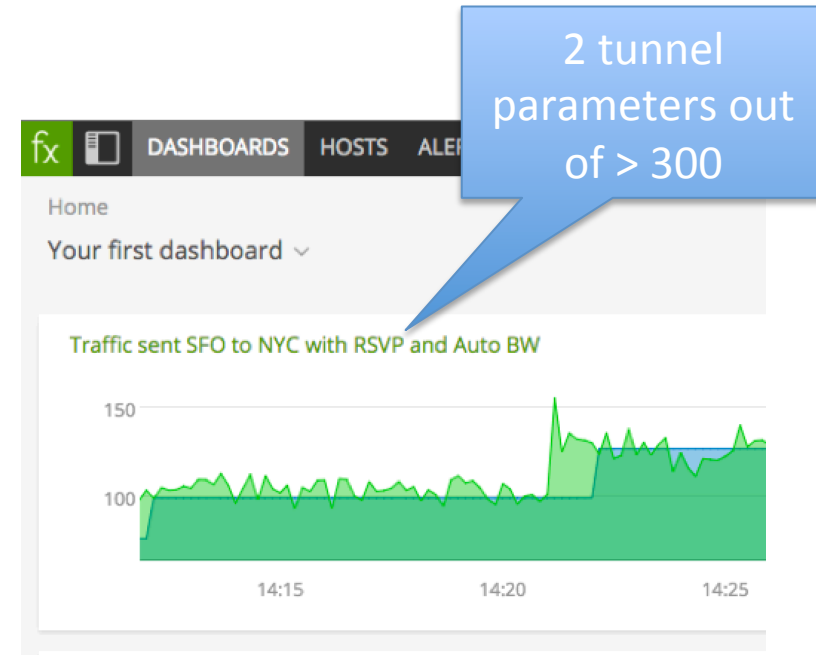
# Lesson 5: Some Models Are Harder Than Others

- Less “Impedance Mismatch” Than Config
  - Diminished Many to One Problem
  - Same underlying objects, different paths
  - Mapping operation data is mostly stateless
- Implications for Performance
  - Scatter-gather problem
  - Model mapping can be resource intensive -> optimize software architecture for high speed export
- Model Development Cycle May Lag



# Lesson 6: Filtering Is A Balancing Act

- Massive Amounts of Data
  - Bulk Collection is Efficient
  - Bulk Processing/Export Not So Much
  - Compression (Lossless) is Good
- On-Box Filtering Loses Data
  - Can't Change Your Mind About What's Important Later
  - Can't Scale Out Embedded Processing
- Options
  - Balance Filtering Between Network and Collector
  - Don't Rely on Periodic Push for "On-Change" Data (e.g. RIB events)



# Lesson 7: Self-Describing Data Has Pros and Cons

## GPB – “compact”

```
1: GigabitEthernet0/0/0/0
50: 449825
51: 41624083
52: 360333
53: 29699362
54: 91299
<snip>
```

2X faster  
Operationally more complex  
(but not relative to SNMP!)

## JSON/GPBKV – “self-describing”

```
{InterfaceName: GigabitEthernet0/0/0/0
  GenericCounters {
    PacketsSent: 449825
    BytesSent: 41624083
    PacketsReceived: 360333
    BytesReceived: 29699362
    MulticastPacketsReceived: 91299
  }
}
```

3X larger  
Native models: still need  
heuristics for key names

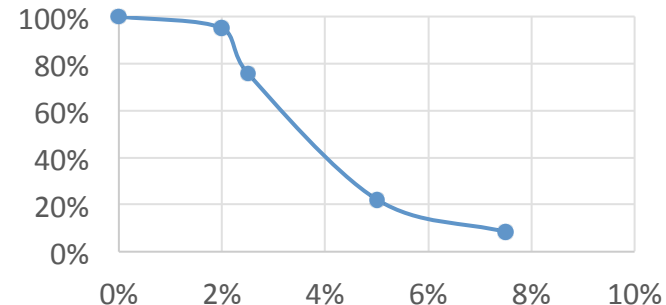


# Lesson 8: Transports Have Trade-offs

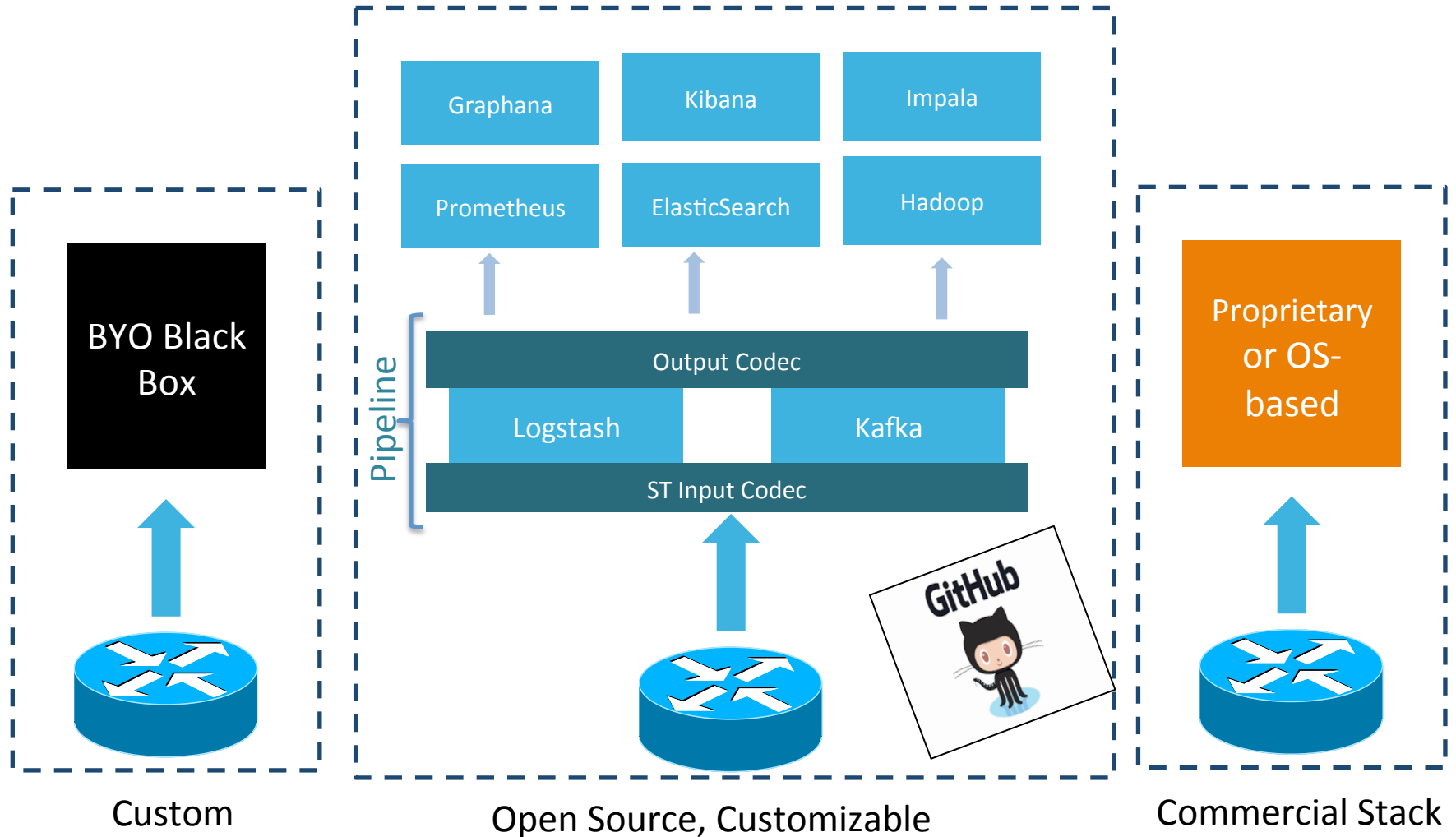
- UDP – can't fragment, packets might get lost, insecure, fast
- TCP – can fragment, guaranteed delivery, insecure, vulnerable to loss
- Two motivations for gRPC
  - Unification of management protocol
  - Secure, two-way streaming

| Data Model | Encoding        | Transport | Score |
|------------|-----------------|-----------|-------|
| Native     | Binary          | UDP       | 100%  |
|            |                 | TCP       | 88%   |
|            | Self-describing | TCP       | 42%   |
|            |                 | gRPC      | 38%   |

TCP Throughput vs Loss

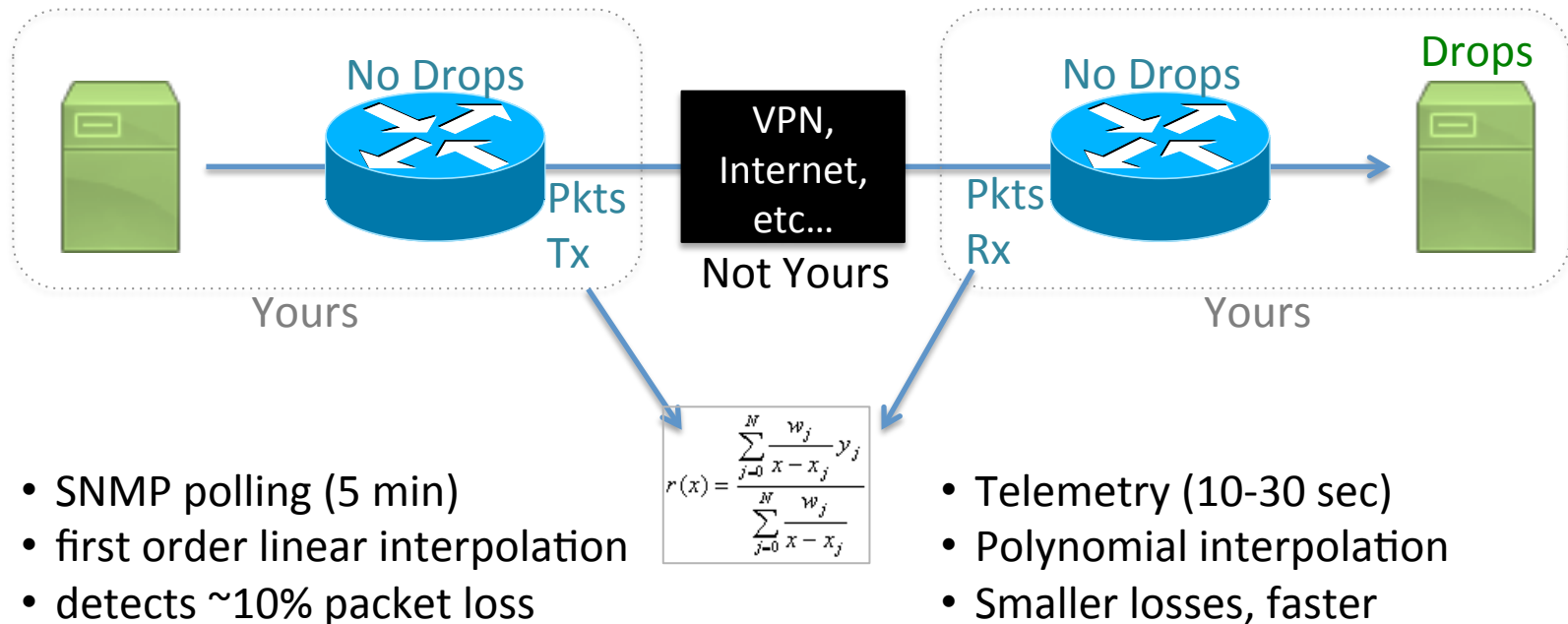


# Lesson 9: Time to Think About Analytics Pipeline & Platform



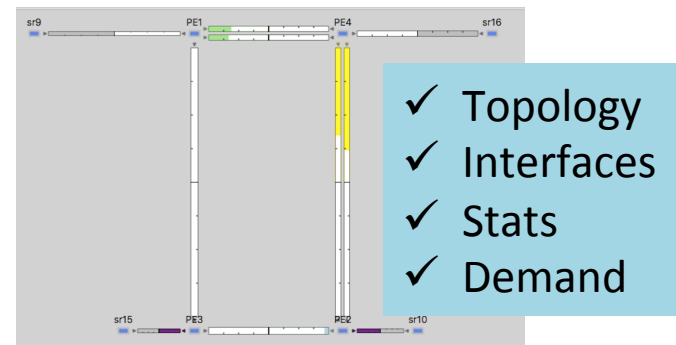
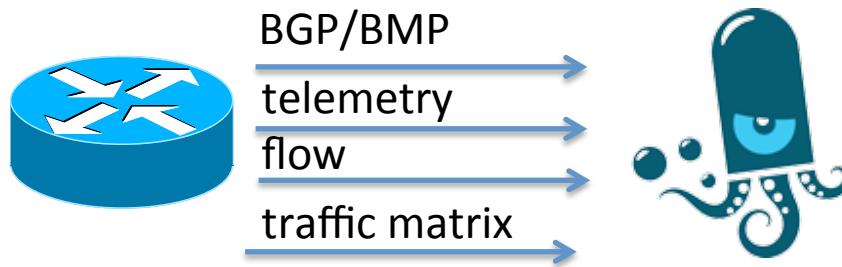
# Lesson 10: Lots of Data is Good, Diverse Data is Multiplicative

- More Data, Better Interpolation

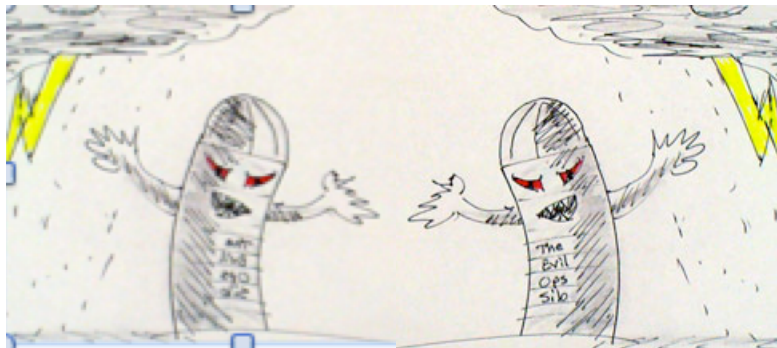


# Lesson 10: Diversity Is In the Eye of Beholder

- No Single Metric Describes the Network



- Application + Networking Data ?



# More Lessons

- Timing is everything
  - Accuracy is (usually) more important than latency
- Best telemetry is baked into feature
  - E.g. Interface stats heavily optimized for moving around quickly, QoS not so much, Eventing
- Telemetry policy should be inline & configurable
  - Evolution: manage policy like any other YANG config object
- Open Source Isn't Free

# Conclusion

- Telemetry Is Real...But Not Done.
- Performance Is King...Until You Achieve It
- Don't Re-Invent the Wheel





**THANK YOU**