Network Visibility for the Automation Age.

Designing telemetry protocols for datacentre and WAN automation.

Rob Shakir, IP Architect
Google

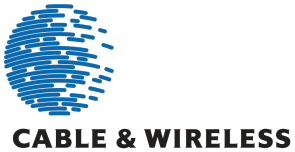
Ecole Polytechnique, Paris - INF566.
March 6, 2019.

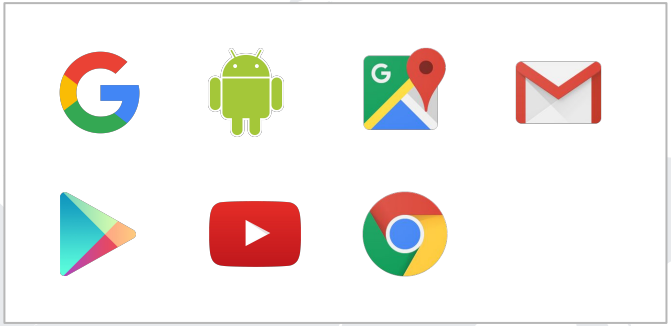
Rob Shakir

IP Architect
Google

Technology development and strategy for Google's global network infrastructure.

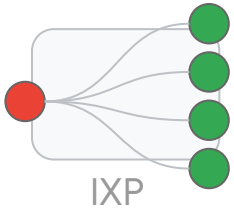
A journey through deploying network infrastructure...



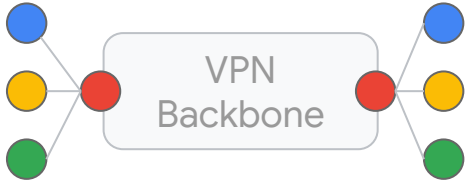


Google™

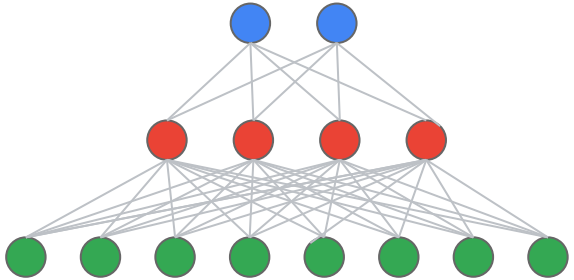
Automate, automate, automate.



Scale

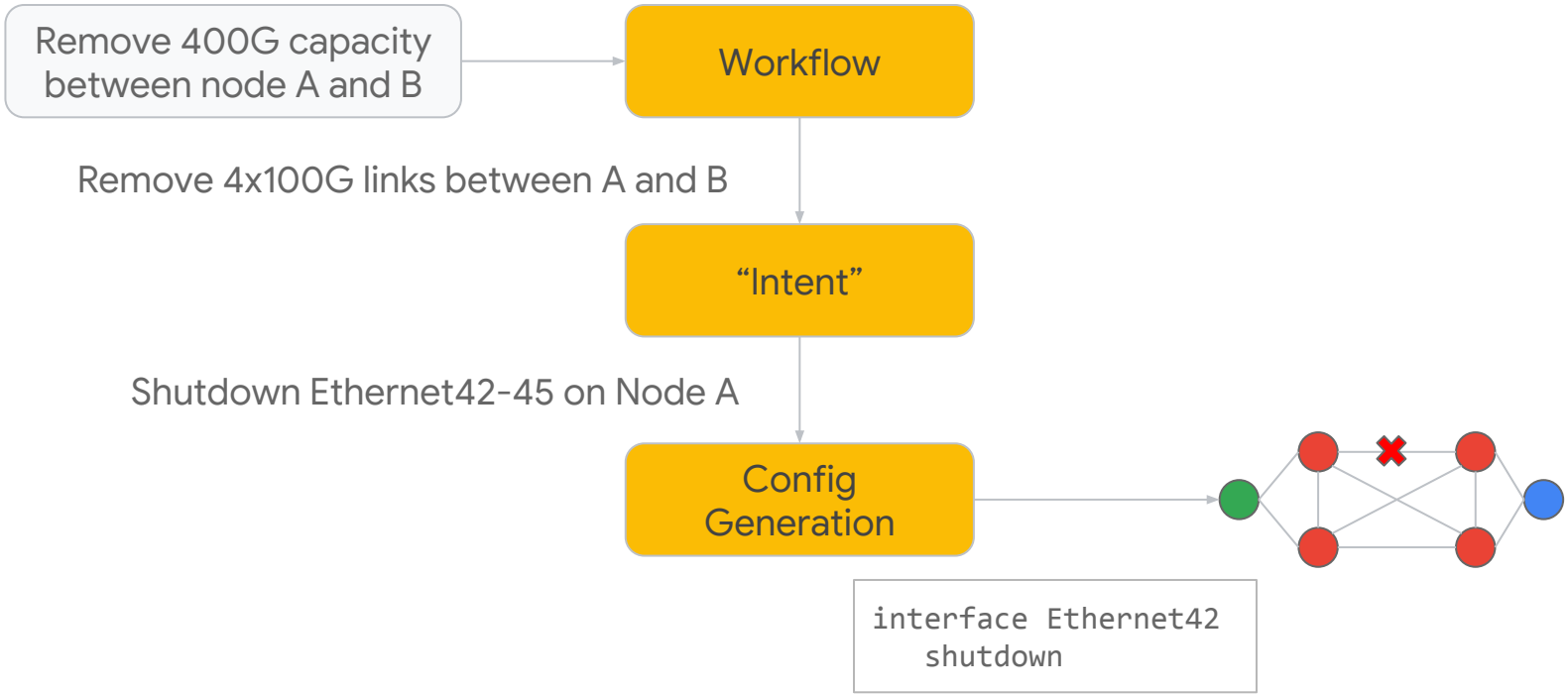


Consistency

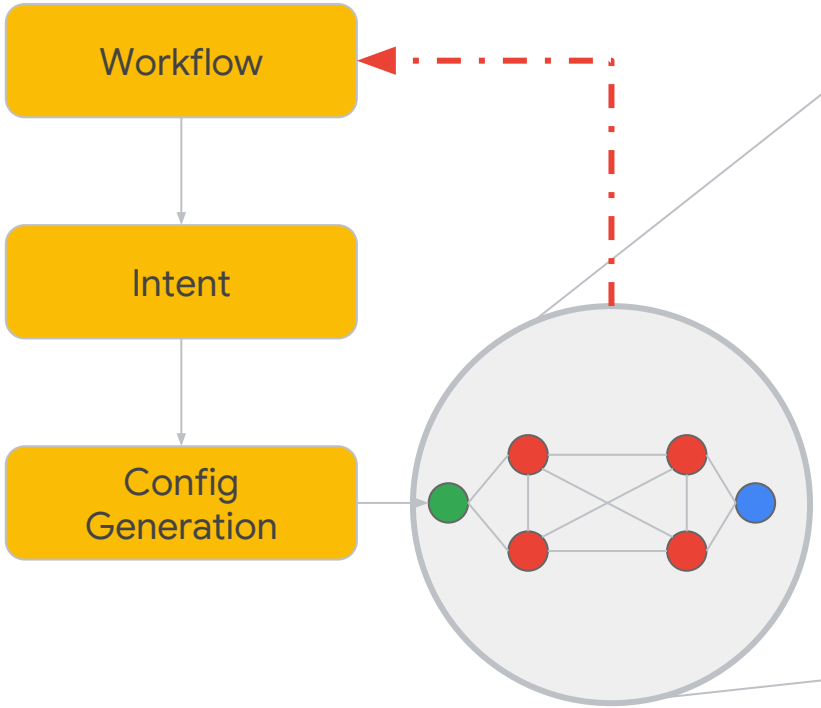


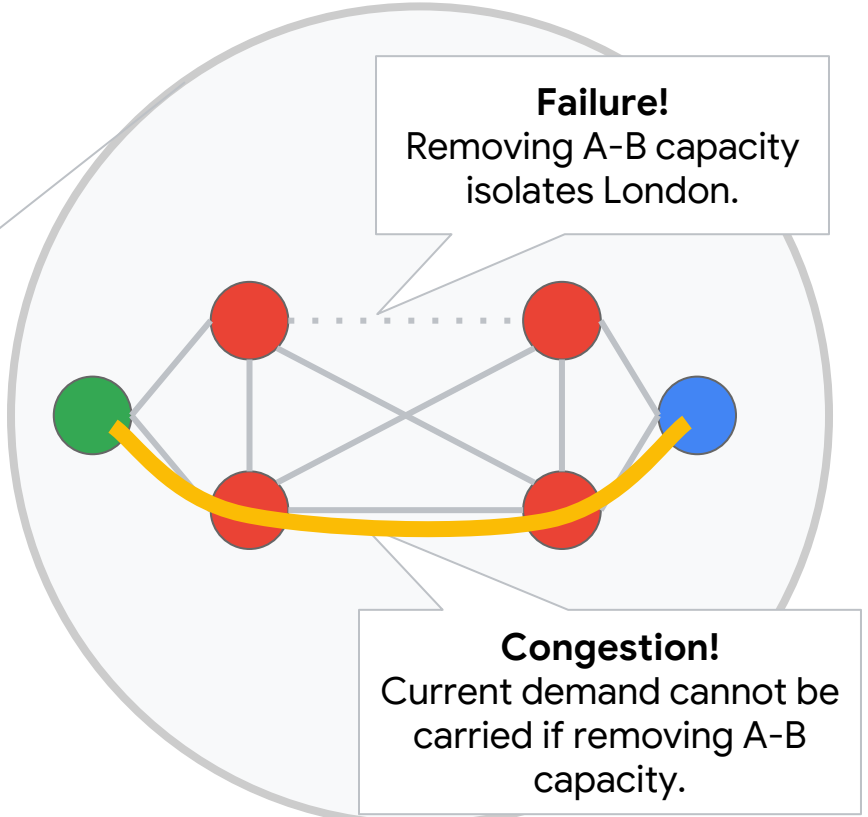
Correctness

The common “NMS” mental model.

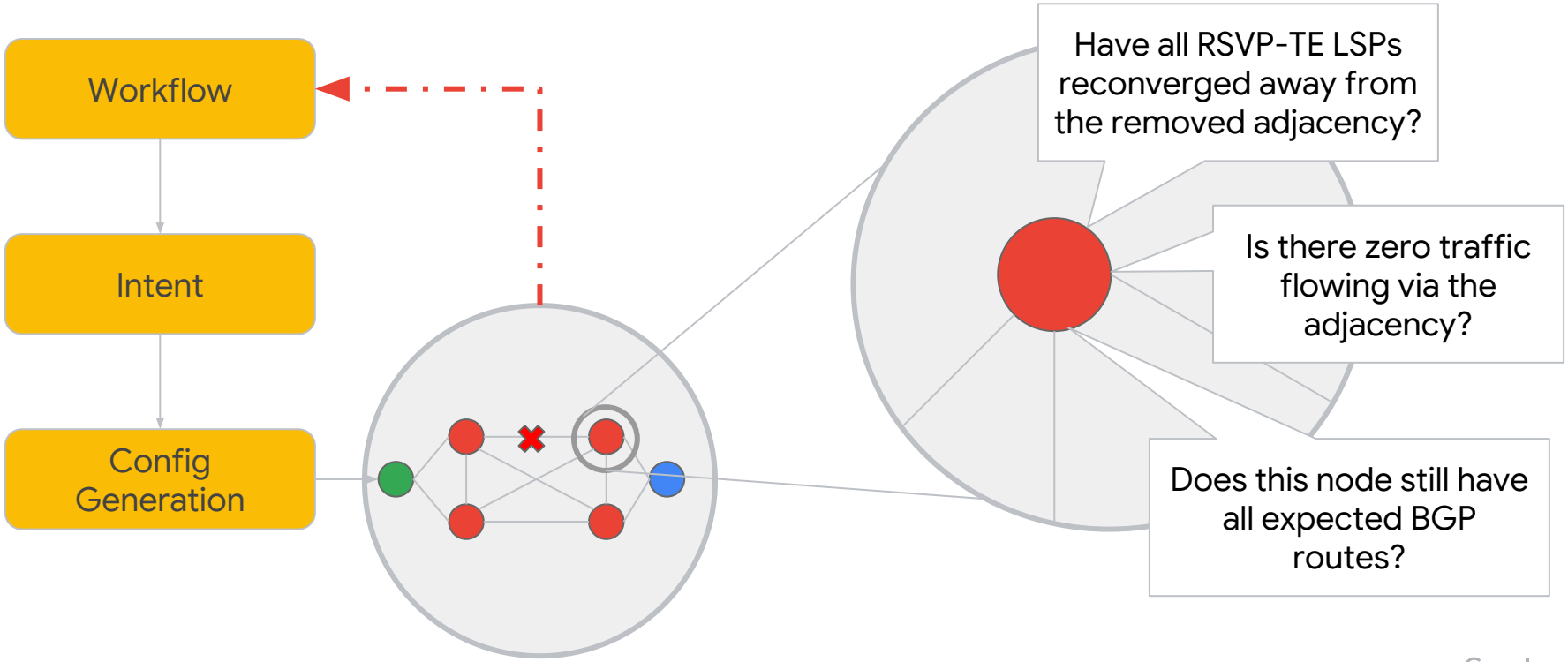


Is this change safe?



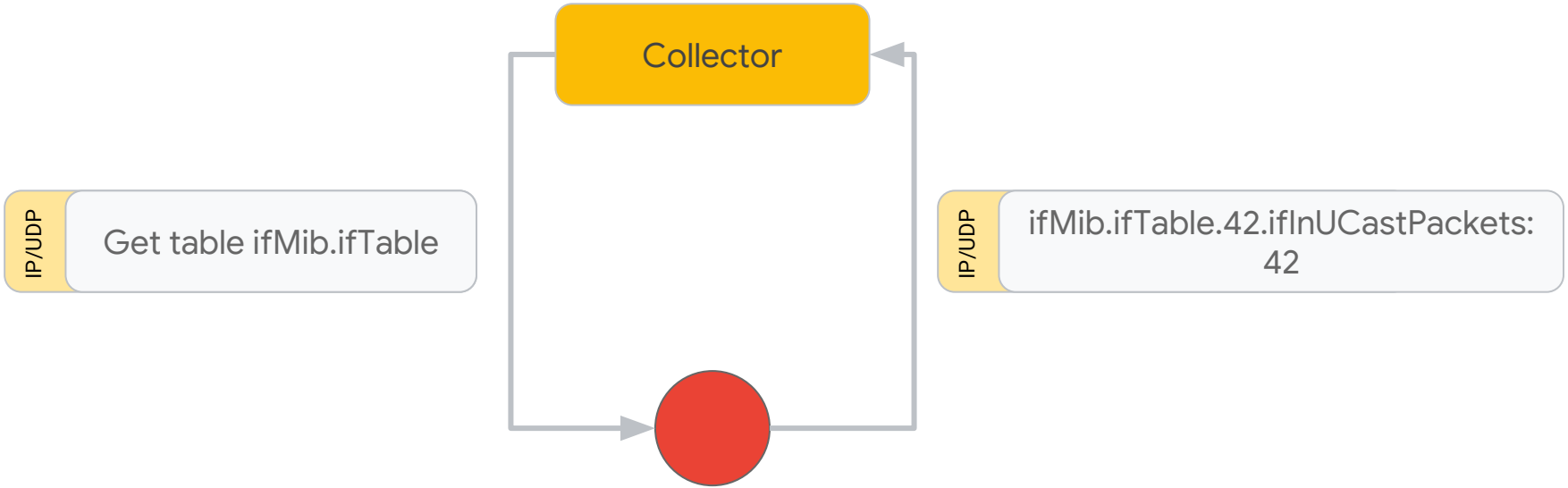


Was this change successful?



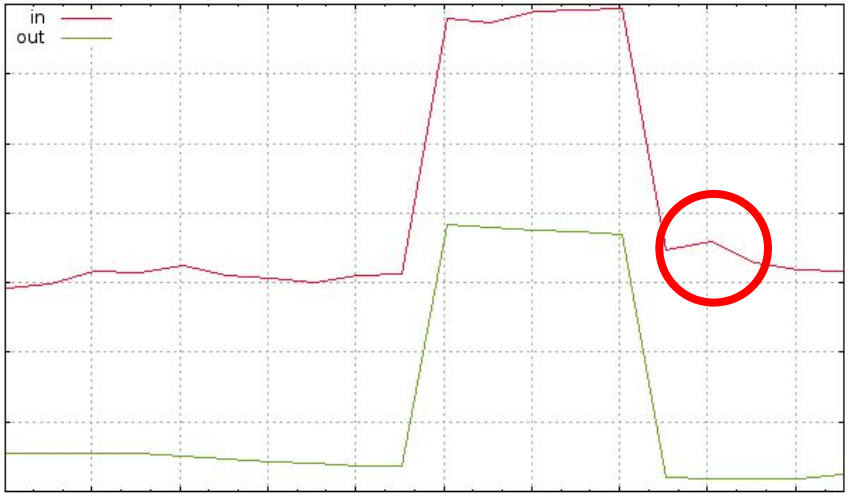
We can’t safely automate the network without understanding its state.

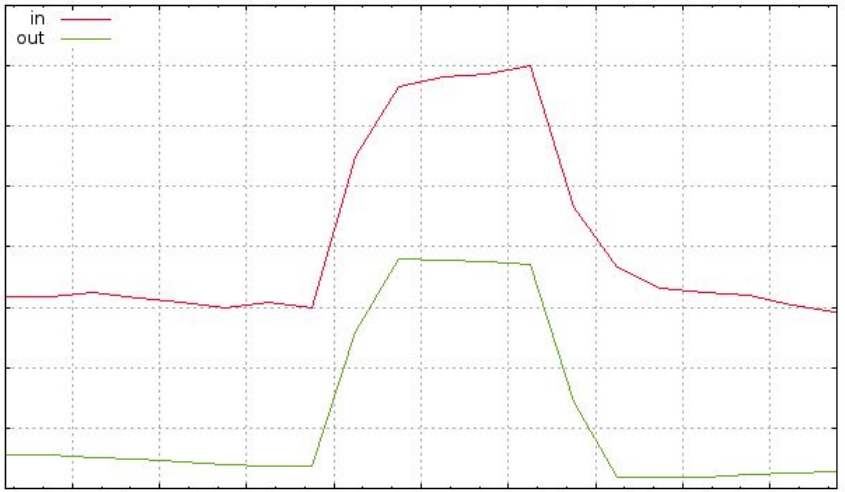
SNMP for monitoring.



Major challenges: No timestamping, over a lossy transport.

Consequences of no timestamps.

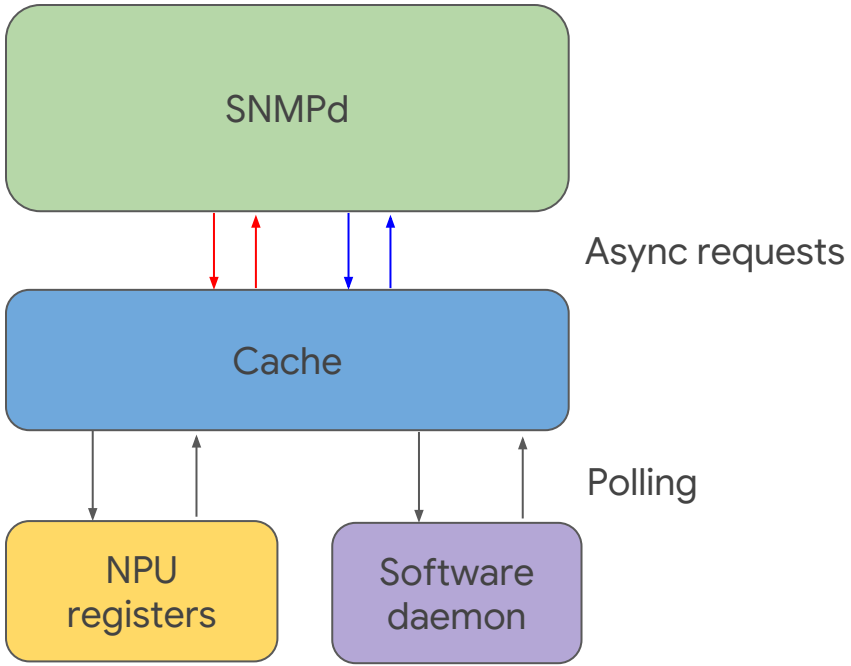




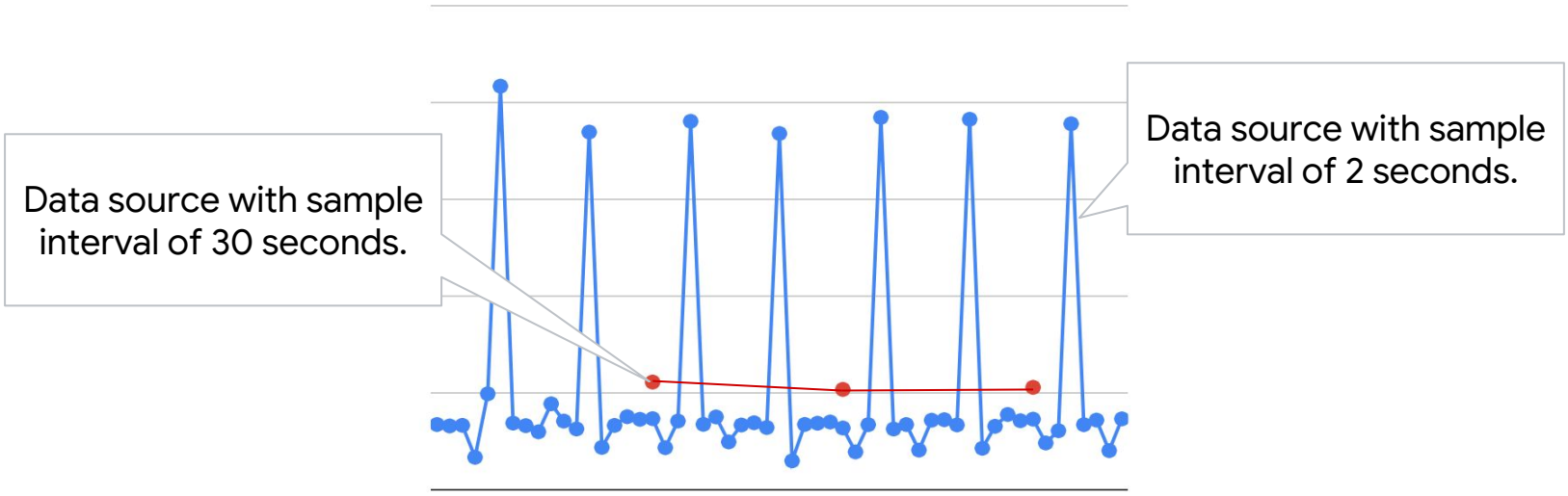
Same traffic profile - different telemetry sources.
Introduces both artifacts and inaccuracies when calculating counter first derivative.

Typical router SNMP implementations.

- Client scheduled requests mean servers implement caching.
- Data is often 30-60+ seconds stale - limits data fidelity.



Implications of under-sampling.



SNMP rates can be out by more an order of magnitude.

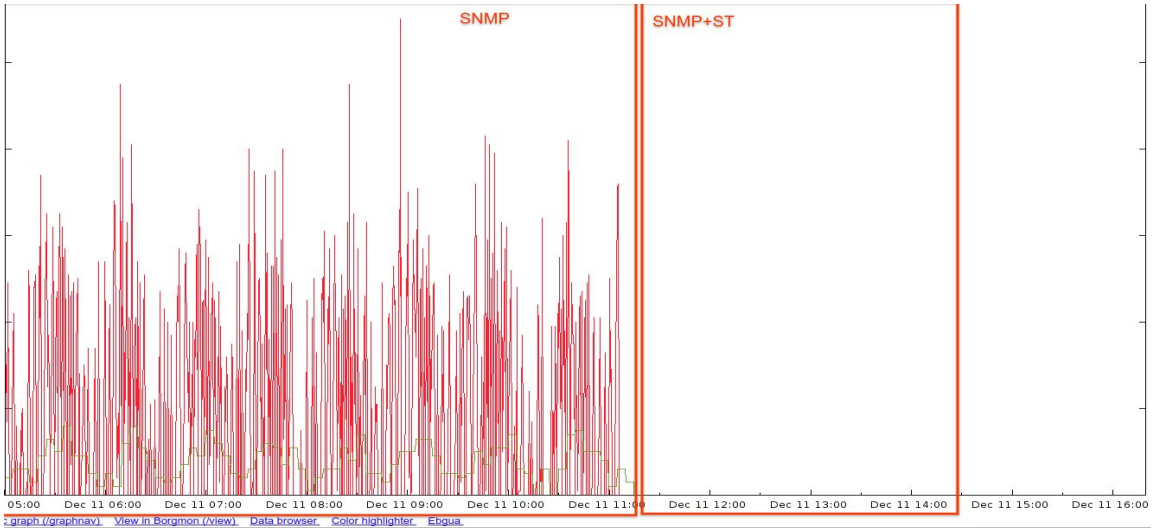
Implications of lossy transport.

Over-polling -- significant additional scale load.

UDP transport means no reliable delivery of traps.

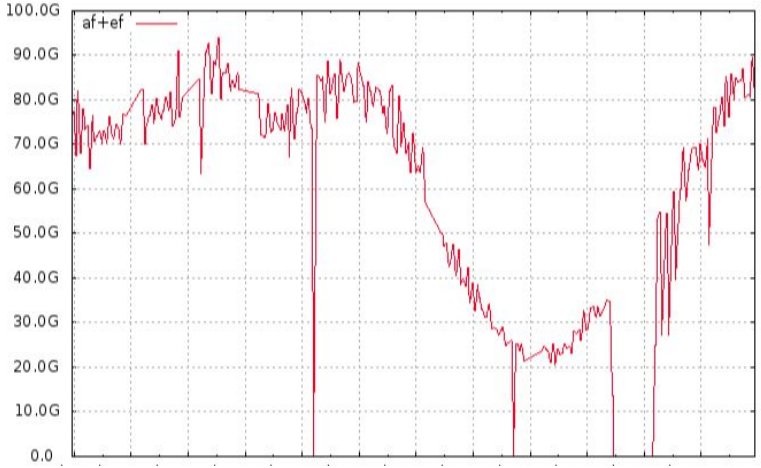
Automation systems that need to detect changes further compound scale challenges - poor technical design makes the solution unviable.

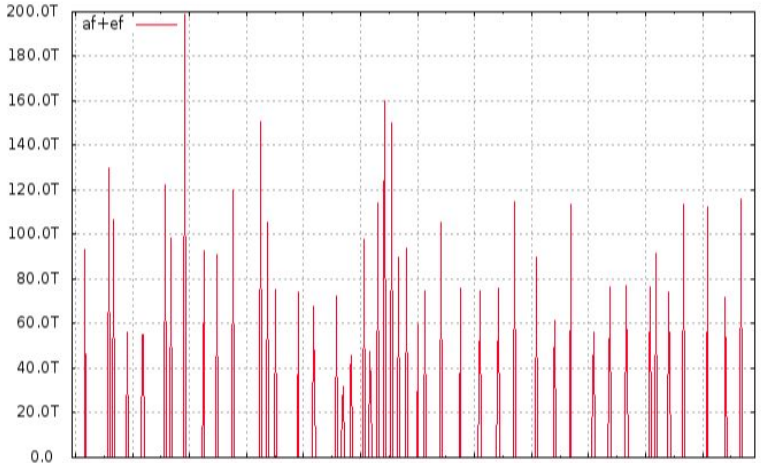
Implementation quality - 30 years on.



Significant number of timeouts across different implementations -- even after years of implementation.

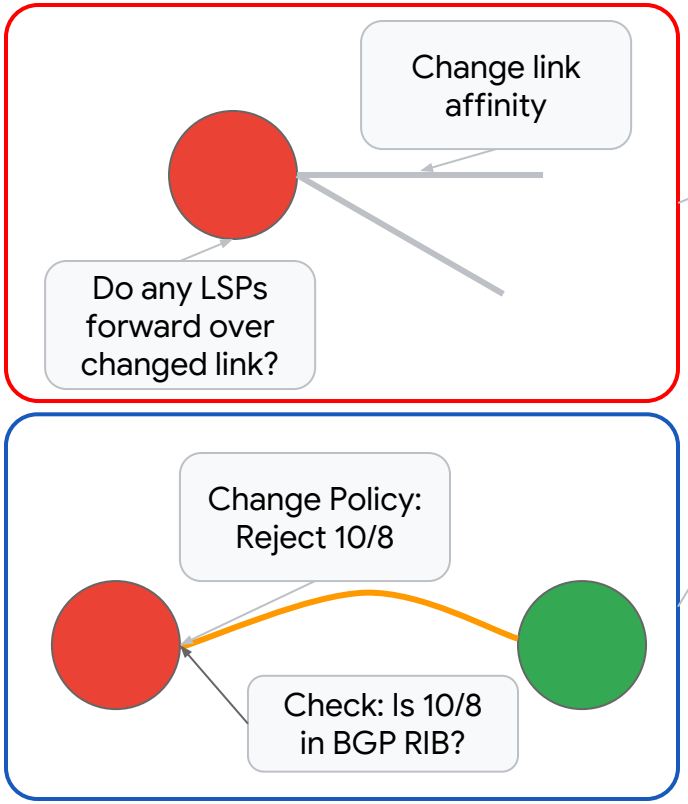
Errors in Rate Calculations

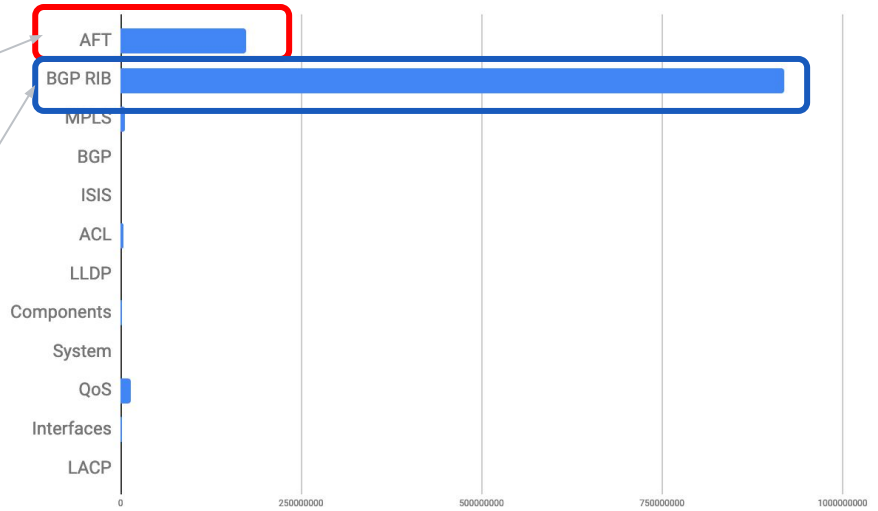




Even with sliding window averaging -- highly erroneous rates calculated.
Zero traffic or hundreds of terabits through one interface!

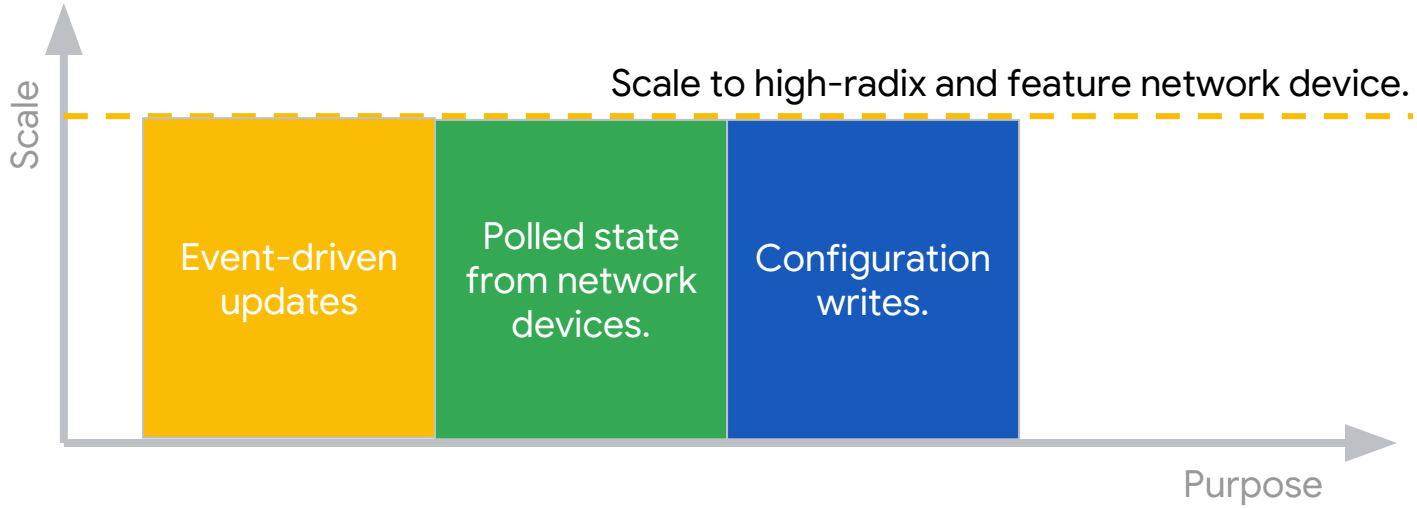
Additional pressures of automation.



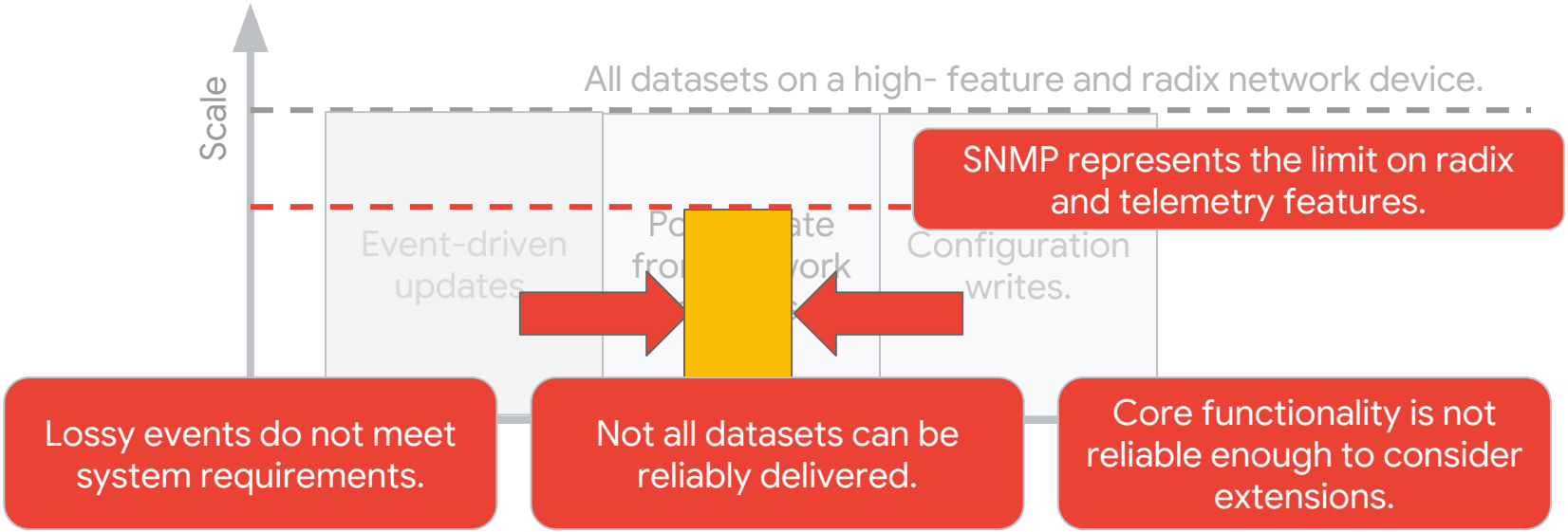


Data sets required for robust state validation are orders of magnitude larger the data we poll today.

SNMP's intended scale and purpose.



SNMP's actual scale and purpose.



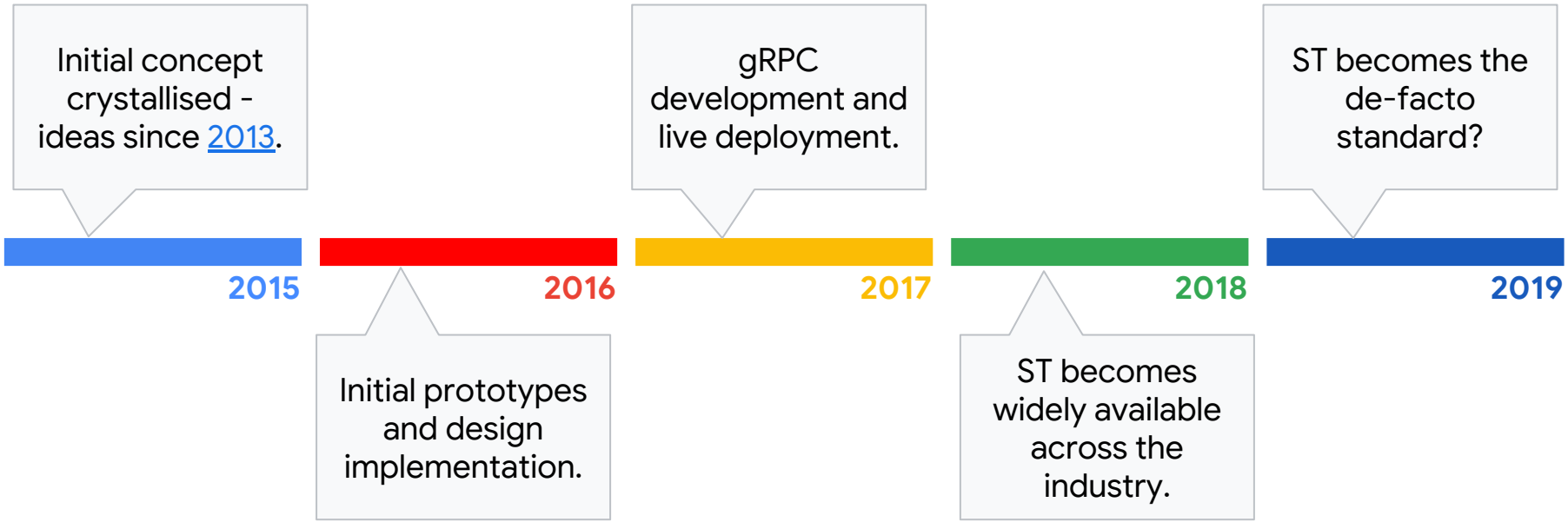
Is SNMP a success?

Positive net value	Incremental deployability	Open code availability
Freedom from usage restrictions	Open specification	Open maintenance
Good technical design	Extensibility	No hard scalability bound

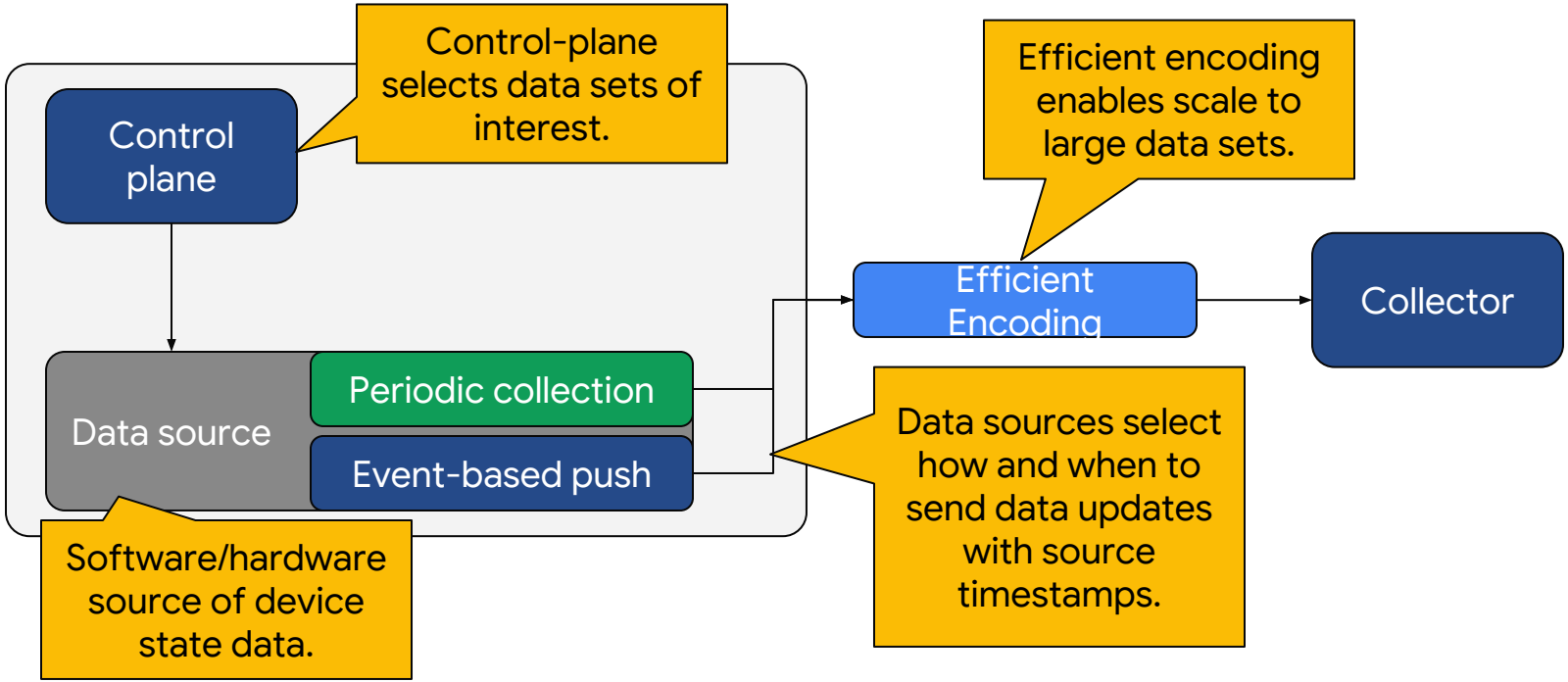
SNMP has been widely deployed, but is a limited success for its entire scope. Despite this success, it is currently hitting its design and scale limitations.

Replacing SNMP with Streaming Telemetry.

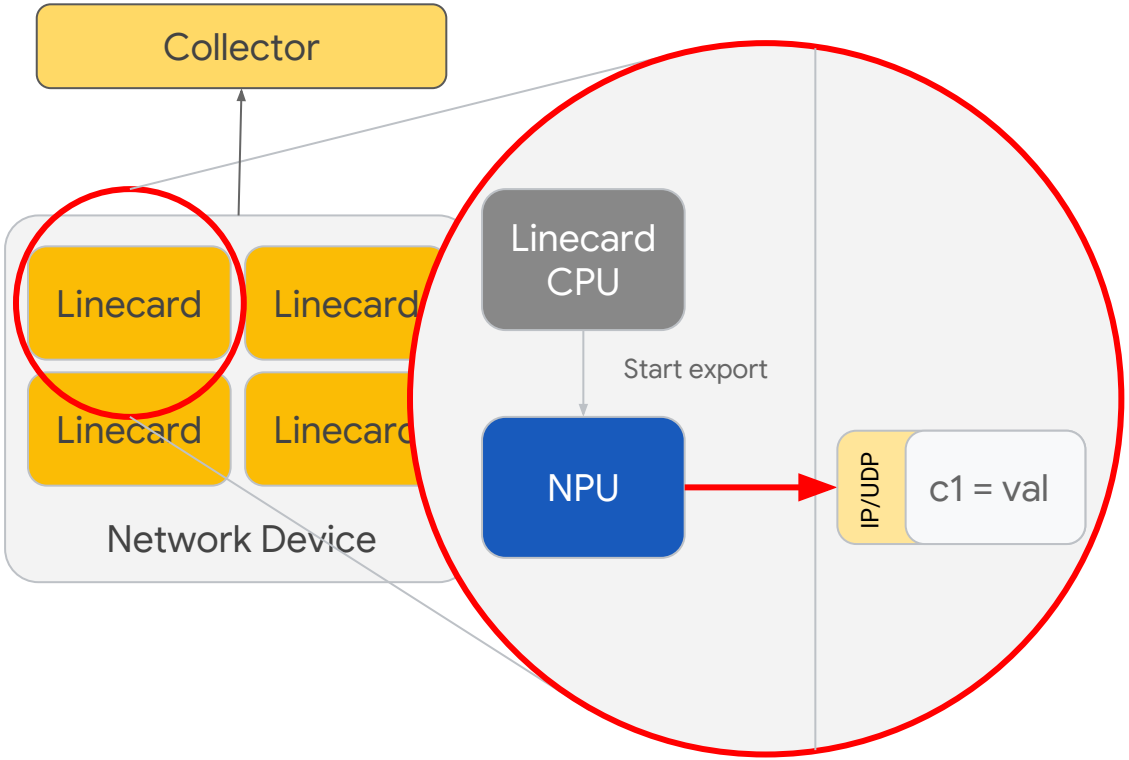
ST's Development Timeline.



Streaming Telemetry (ST) concept.



An initial solution: UDP ST.



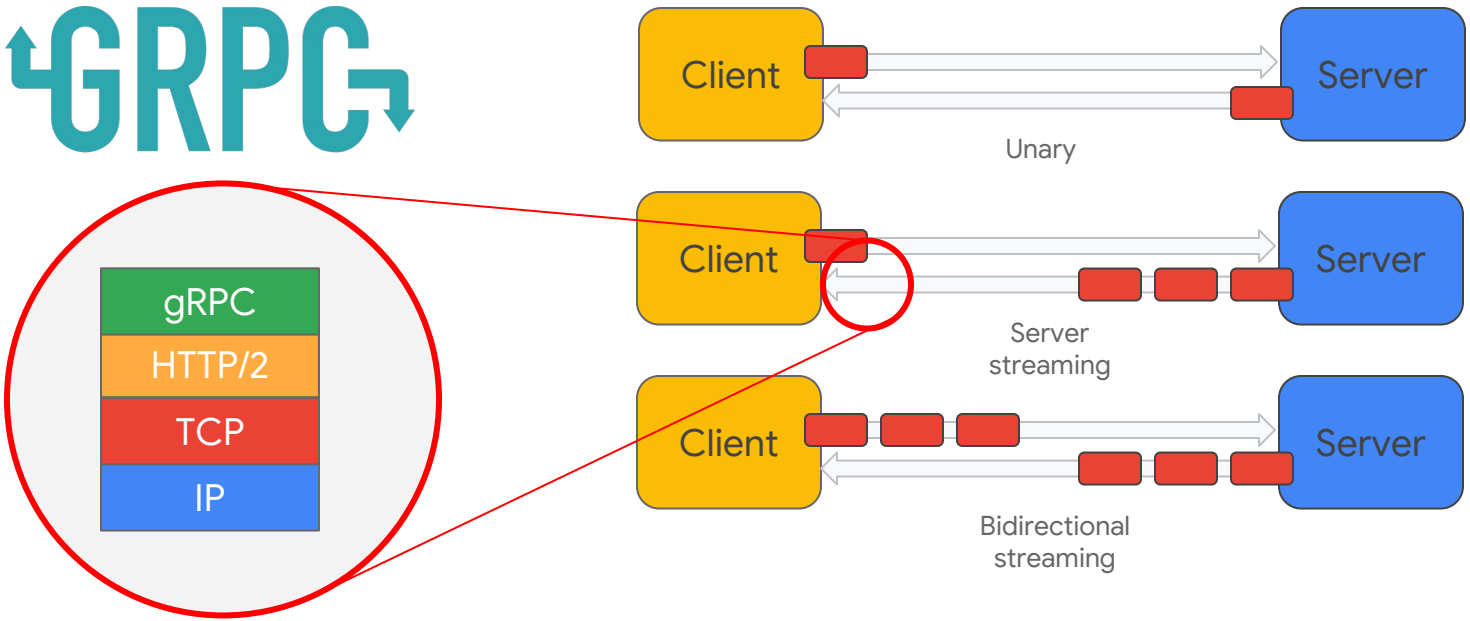
- Efficient for device.
- Timestamped at source.

- No ability to distinguish between data loss and device overload.
- Unreliable transport - oversending still required.
- No knowledge of device health.

Learnings of note:

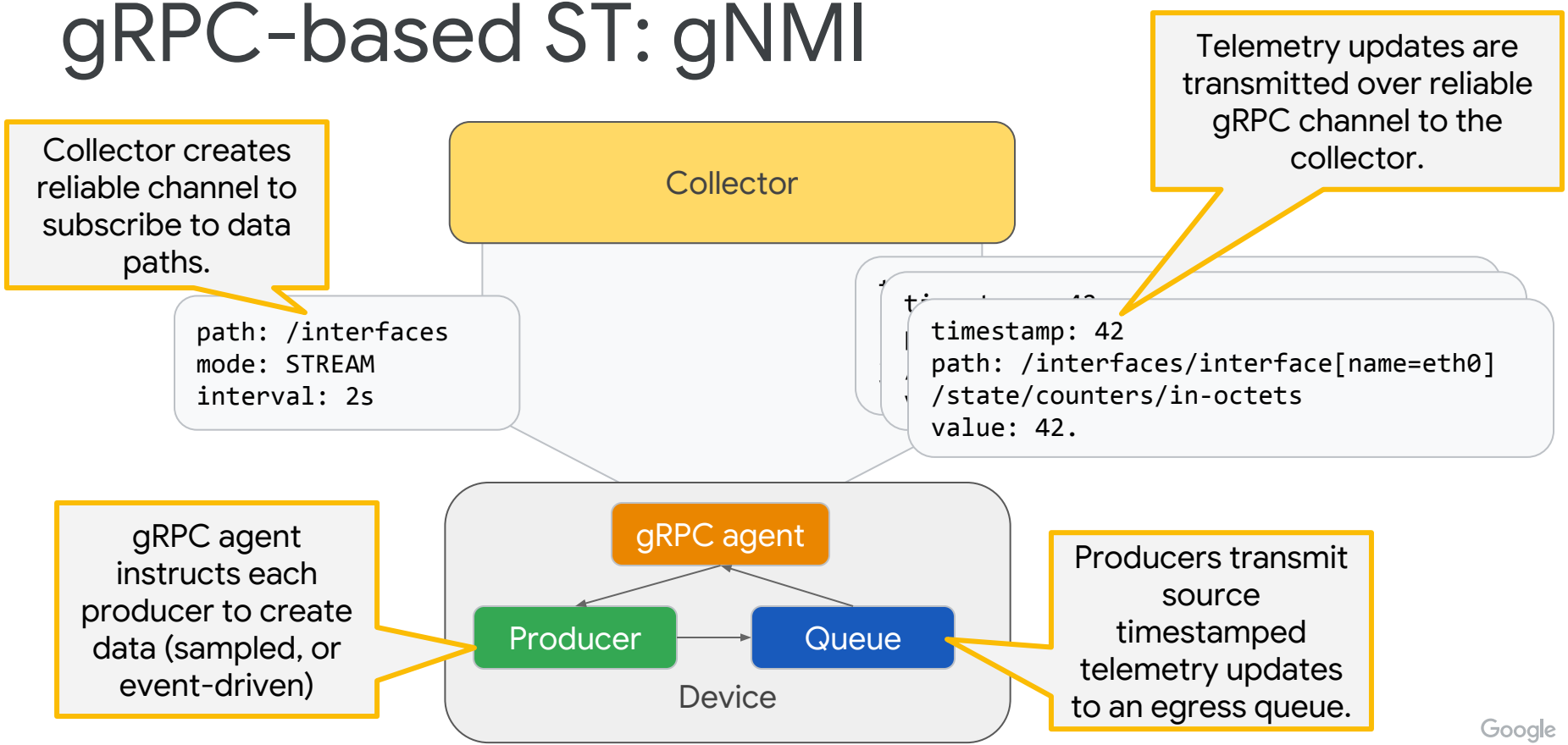
- Don’t solely focus on implementation complexity.
- Sometimes the operational complexity and implications of an approach are not immediately evident - prototypes make them clear.

Choosing a reliable transport.



Re-use existing transport -- with open specification, and extensive implementations.

gRPC-based ST: gNMI



Addressing key learnings from SNMP.

High scale

Increasing volumes of data as sampling cadence, and streamed datasets increases.

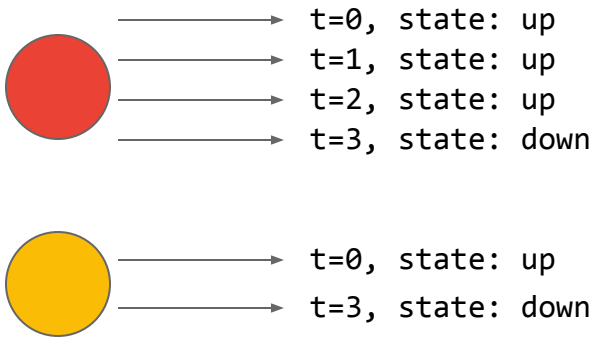
How do we avoid hitting a hard scale limit?

Congestion

Reliable transport means that bottlenecks can occur due to throughput.

How should we handle congestion?

Managing high telemetry scale.



Exploit reliable transport to trigger streaming based on events (ON_CHANGE).

Reduces steady state update volume by **>99%**.

```
path1, value
path2, value
```

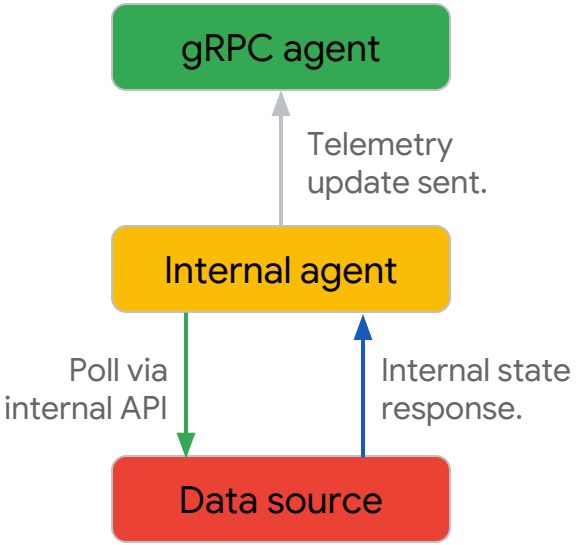
```
prefix
{path1, value}
{path2, value}
```

```
prefix
[binary - path[12]]
```

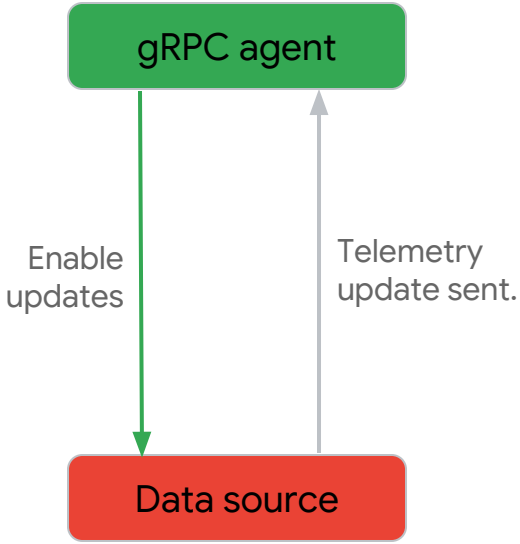
Ensure efficient encoding and payload flexibility to reduce on the wire volume.

Simple prefixing reduces data throughput required by **>85%**.

Implementing Event-Driven Updates.

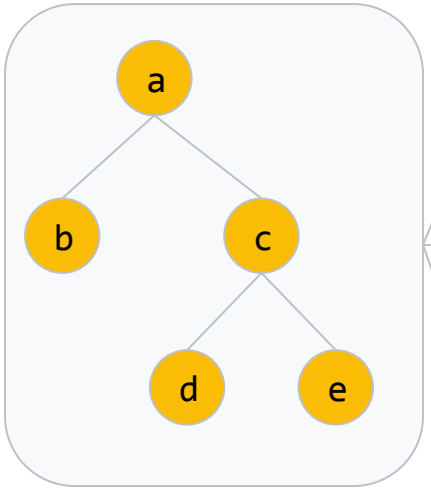


Periodic updates require limited data source changes -- faster time to market - but significant scaling impact.



Making data source changes results in significant reduction in volume, and improved latency.

Improving Encoding Efficiency.



Naïve encoding

```
/a/b = 42
/a/c/d = 42
/a/c/e = 84
```

String de-duplication - significant savings with low computational overhead.

85% data reduction

Prefixed encoding

```
/a/b = 42
prefix: a/c
     d = 42
     e = 84
```

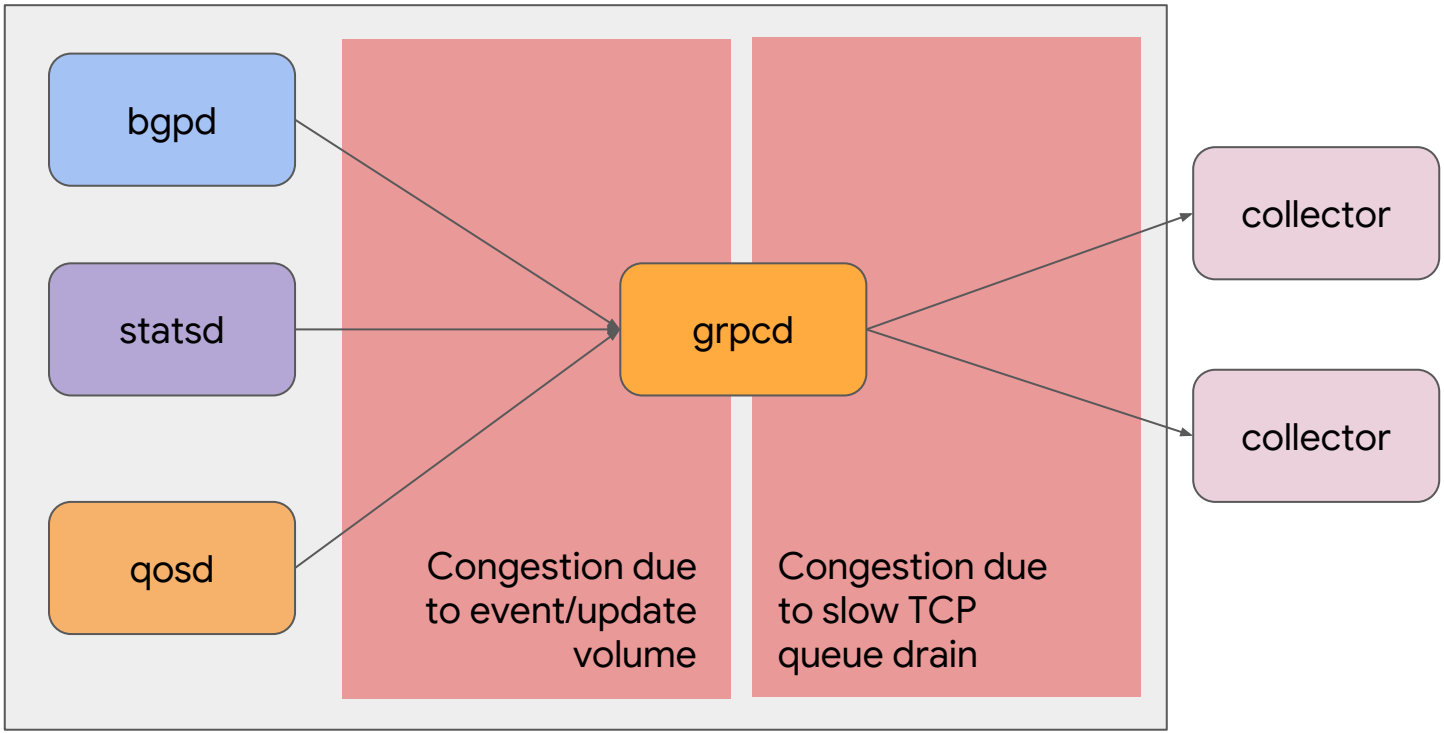
String elimination - requires a defined binary data structure, and higher computational overhead.

Binary encoding

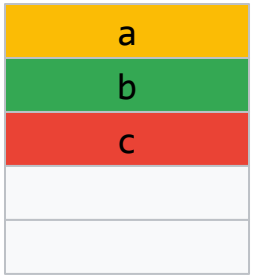
```
{0:
 {1:42
  2: {
      1: 42
      2: 84
  }
}
```

c75% additional savings for high volume schemas

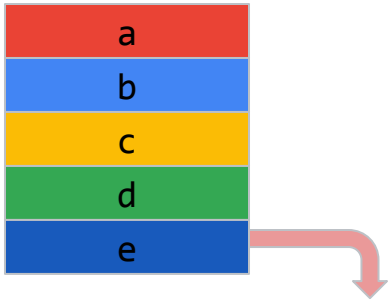
Congestion causes in ST.



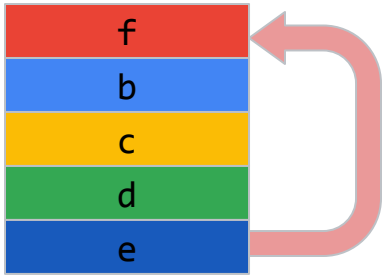
Congestion management approaches.



Telemetry updates to be sent are queued -- does not impact fidelity based on source timestamping.



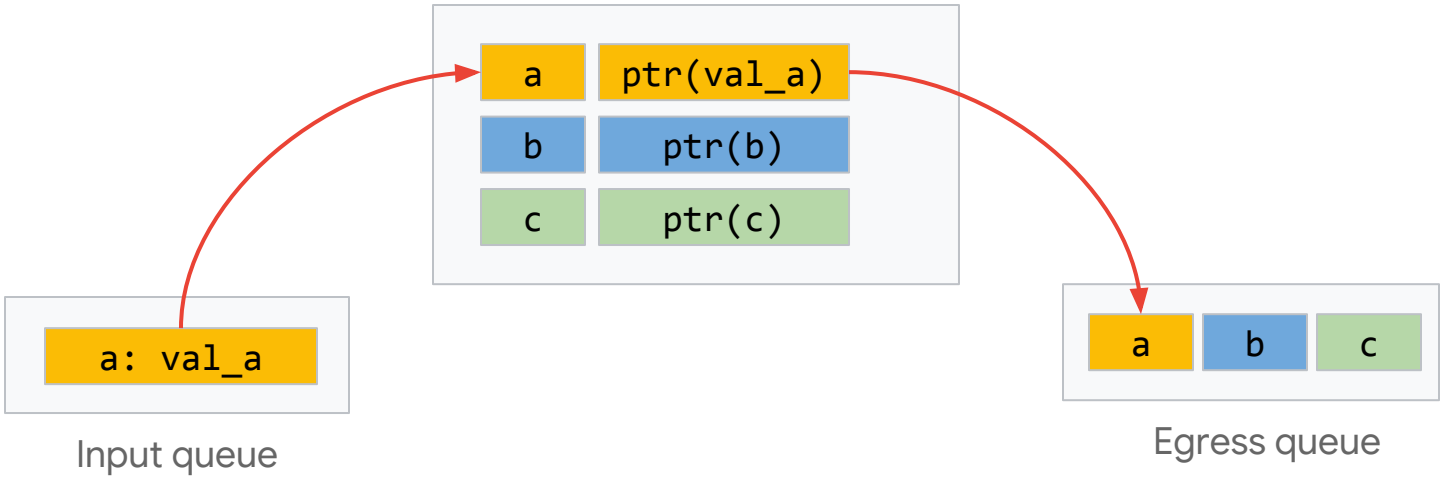
Tail-dropping ensures that there is no overflow - but results in data loss whilst stale data is sent.



A circular buffer results in the stalest values being dropped first, but still risks path starvation.

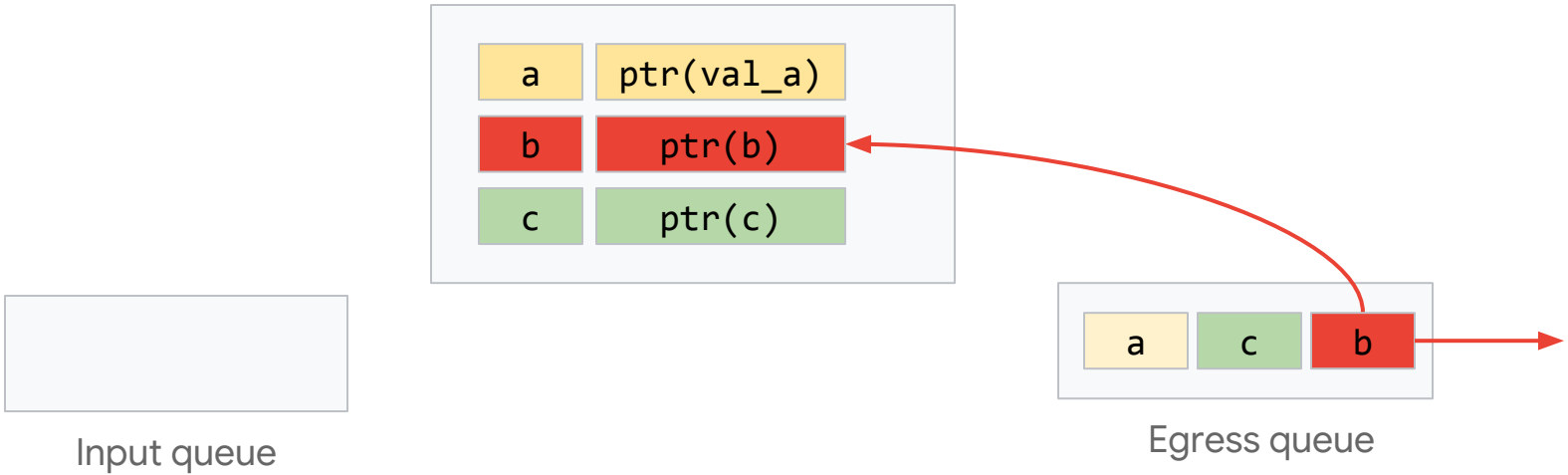
The ideal solution ensures that the most up-to-date value for a path is sent, and ensures no path is starved.

Using a coalescing queue.



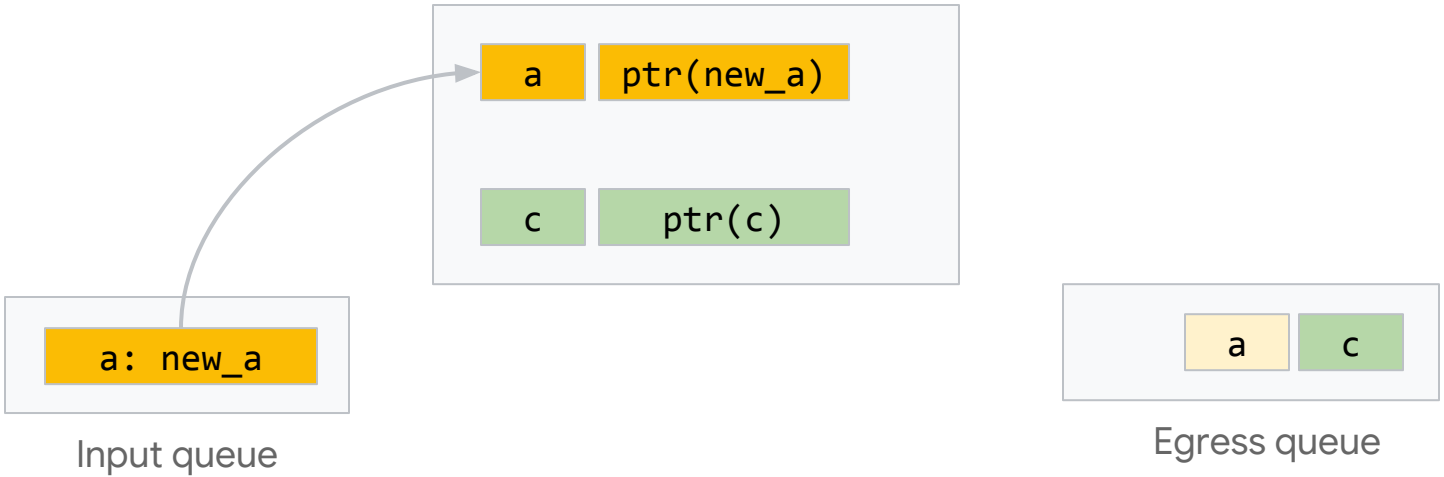
A new update is appended to a map which is keyed by path - pointing to the latest value.
New paths to be sent are appended to an egress queue.

Using a coalescing queue.



When a value is to be sent, the value from the map is retrieved and sent to clients.

Using a coalescing queue.



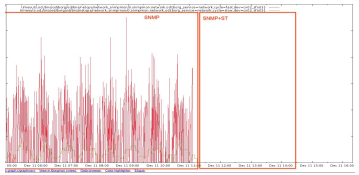
If the value already exists within the map, the pointer is updated, but the egress queue is not altered.
Thus, when ‘a’ is sent, it is sent with the latest value.

Learnings of note:

- Ensuring extensibility to ensure scaling bottlenecks can be addressed is key - but avoid premature optimisation.
- Features of good technical design might actually be internal to an implementation - must consider how prescriptive to be.

Considering 5218's Success Criteria.

Addressing a real need.

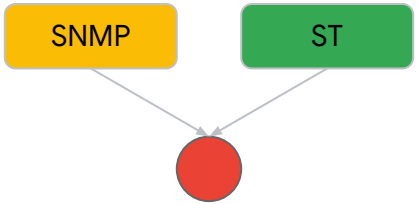


Targeted to address known operational limitations -- particularly those limiting network deployments.

Open specification and code.

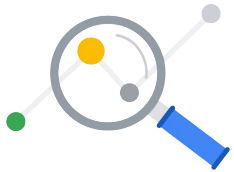
Both specification and reference implementations developed as open source code.

Incremental deployability



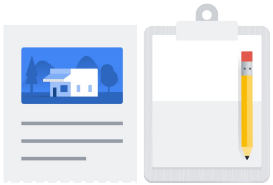
Deployment approaches focused towards being able to run a hybrid between SNMP and ST during development.

Success criteria not in 5218.



Ability to test interpretation of specification is important for a usable product.

Published an open source test framework.

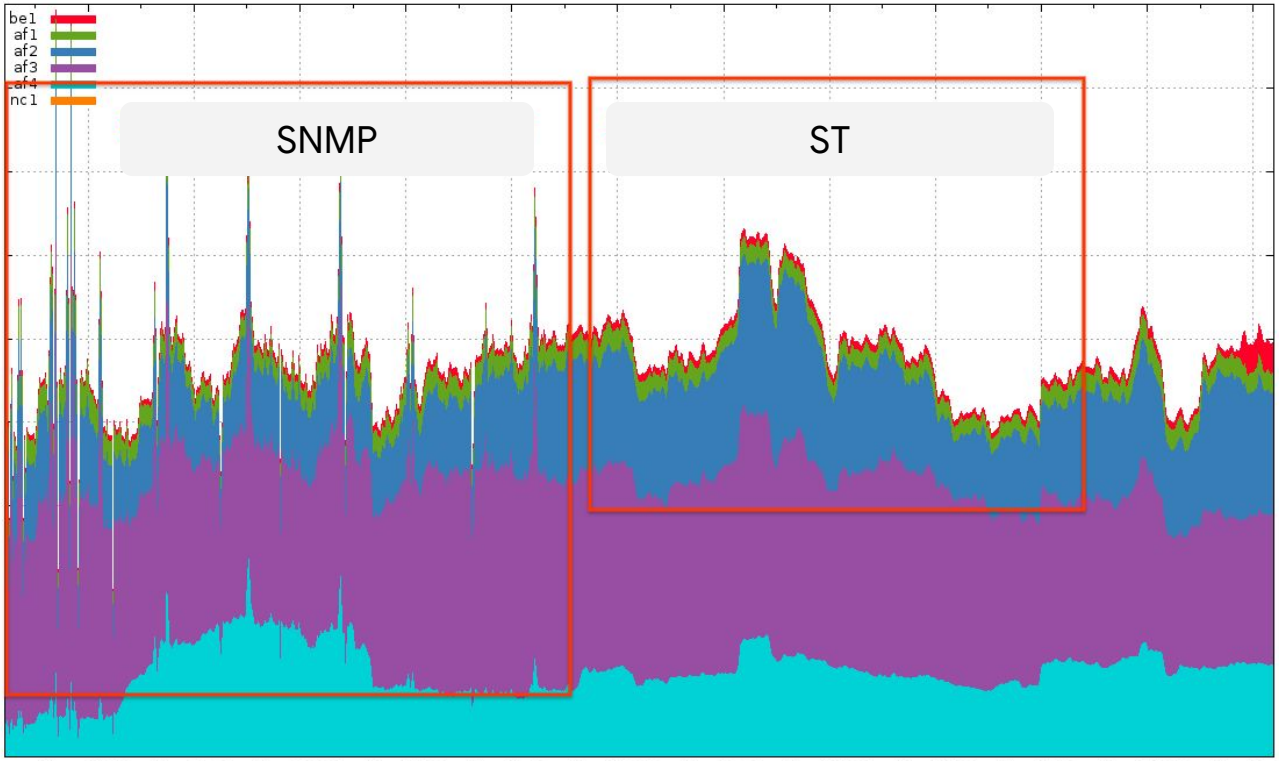


Constraining complexity of base implementations -- *avoiding feature creep*.

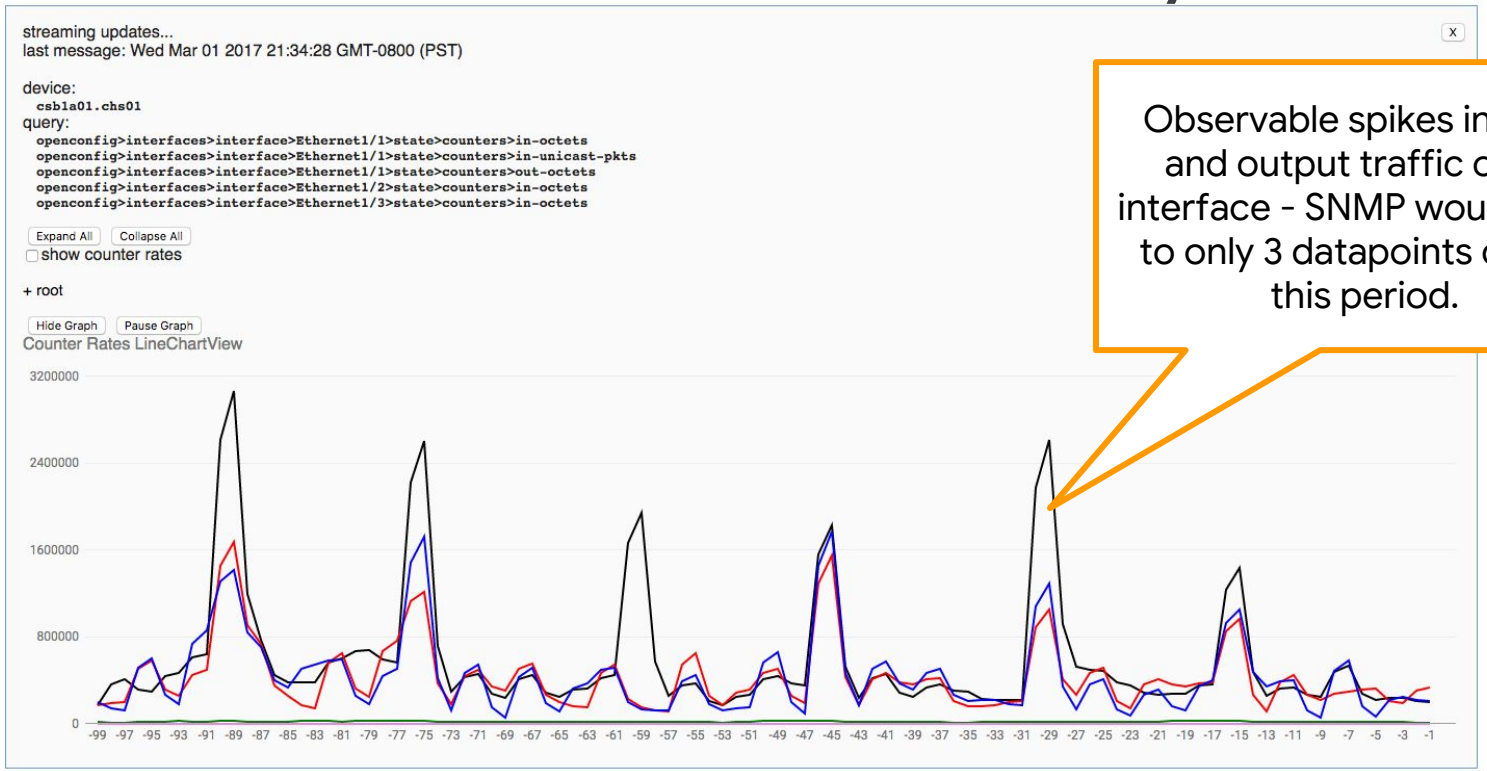
Co-maintained by Google - with key group of implementation partners.

Is ST successful?

Improving fidelity of network statistics.

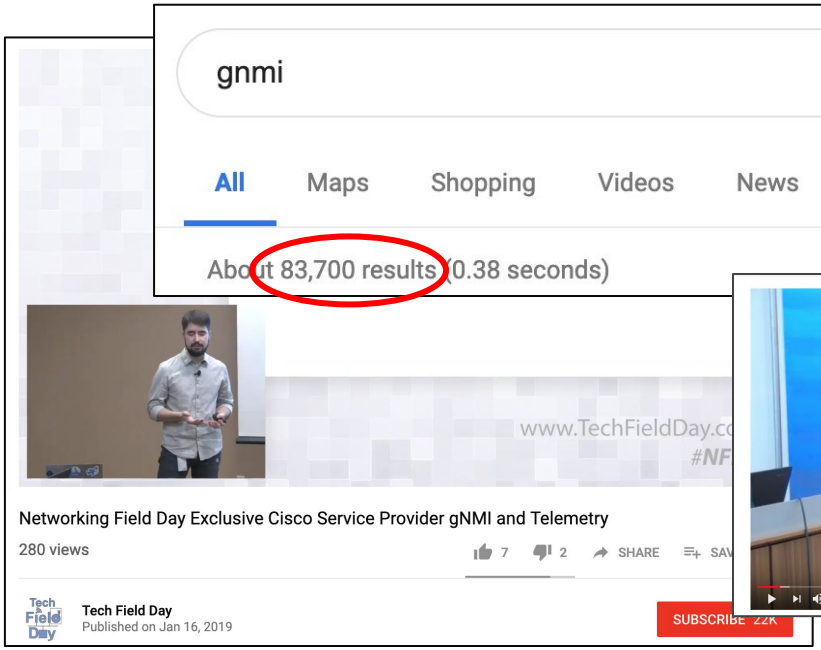


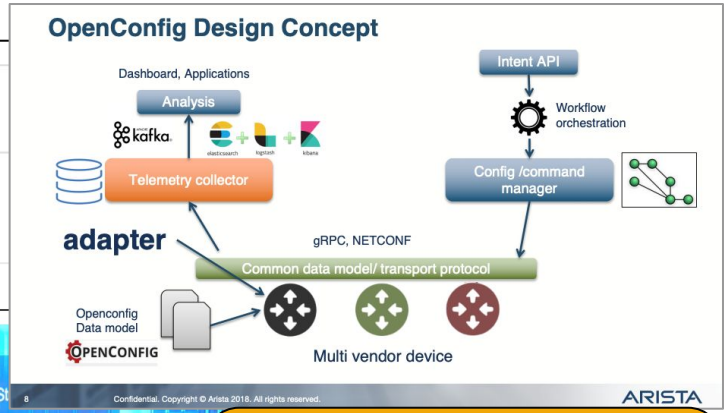
Achievable scale for telemetry data.



Observable spikes in input and output traffic on an interface - SNMP would scale to only 3 datapoints during this period.

Industry Adoption of ST.



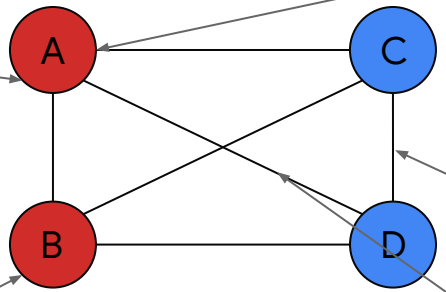


Multiple vendor implementations and operators consuming ST!

Using ST: Validating Network State.

Pre-check in ST data:
Are the interfaces on A connected per the topology model?

Pre-check in ST data:
Are all resilient links to B, C and D carrying traffic?



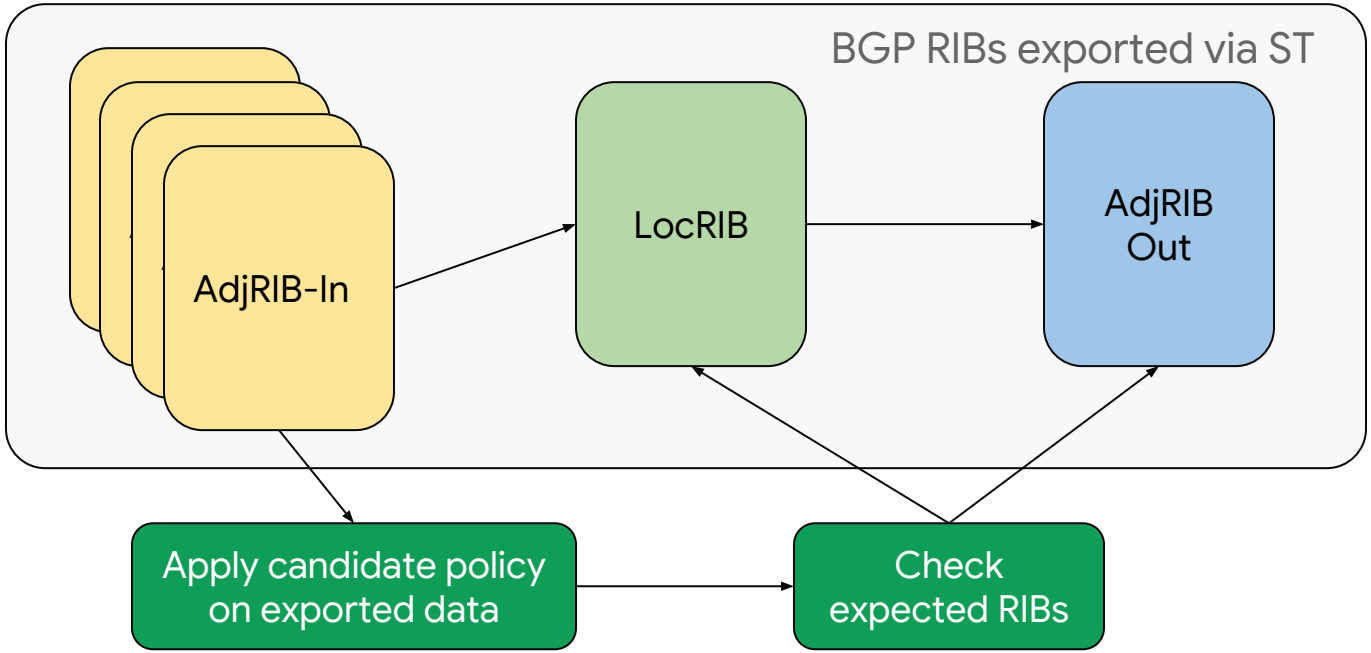
Drain operation -- remove traffic from device A.

Post-check in ST data:
Is the IS-IS overload bit set for device A?

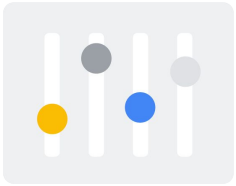
Post-check in ST data:
Are any active next-hops on B, C and D forwarding towards A?

Post-check in ST data:
Are links towards A carrying 0 traffic?

Using ST: Pre-checking BGP policy.



Concluding Observations...



The lessons in RFC5218 continue to be good guidelines for designing new technologies and approaches.

Iterate, iterate, iterate...
Operational and real-world experience are as, if not more, important than these design lessons.

Cross-company and industry collaboration continues to be critical to driving new technologies - snowflakes aren't good for anyone.