Network Streaming Telemetry

November, 2017

Network Flow Export protocols

NetFlow
- IPFIX
- CFlow
- JFlow

sFlow

AWS - VPC flow logs

NetFlow is a Cisco technology for network flow export - parts of it were originally called cflow on cisco, jflow is Juniper's implentation but it's exactly the same. IPFIX is a IETF protocol, with an RFC, trying to create a universal standard for the technology. It's based on NetFlow v9.

In netflow, each and every network communication can be logged with millisecond precision, so it's especially good for forensic analysis of network activity.

sFlow was created by inmon, and it's used by a bunch of vendors including Brocade, so it's what I work with. sFlow uses packet sampling, so it's possible to miss some short lived communications. It also doesn't timestamp the sampled packets, so the result includes some uncertainty about the timing and volume of the flows.

The advantage is that it requires less load on the router, and if network traffic increases you can decrease sampling frequency, to lower load. With netflow, at scale you'll need to use port mirroring or similar to a device that can be dedicated to generating the netflow data.

AWS VPC is interesting

High precision metrics

We wanted real time metrics on transit traffic to/ from our network

Found SNMP collection non-deterministic

Created network telemetry pipeline

SNMP Collection

Collection is slow
- Need to make multiple GetBulk requests for ifTable
- SNMP data model different from Router data model
- The more pollers, the slower the response (need at least 2 for redundancy)

Port Counters only update every 1s

SNMP comes from the world of 5m network metrics.

Data Models

ifIndex	ifDescr	ifType	ifMTU	ifSpeed
3	GigibitEthernet0	6	1514	1000000000
4	GigabitEthernet1	6	1514	1000000000
5	Loopback0	24	1500	0

SNMP force router to traverse the table column by column

Routers store internal data in a way that’s efficient for them. This is usually that interface statistics are indexed by interface name. So it needs to re-order that data into a table and walk the columns to fulfil the SNMP request.

So each request adds this processing load onto the router.

Also inefficient, 4x8 more description of data rather than actual data

Telemetry

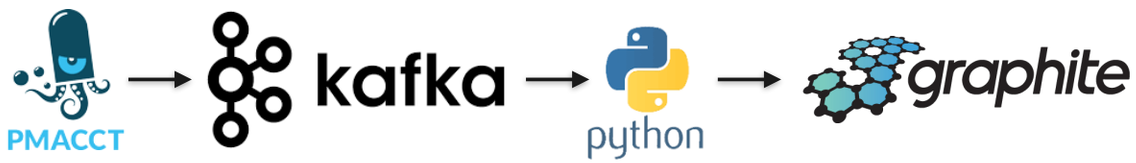
- Cisco Model-driven Telemetry
- Juniper Telemetry Interface
- Arista Telemetry Framework
- … more vendor specific solutions?
- sFlow Counters
- Openconfig

Many vendors now provide solutions to emit data from the router in the structure that makes sense for it, minimising the processing overhead.

Much like with a VM or server, running an agent like collectd and emitting the data

This works with a large estate or number of interfaces, or frequent updates - both things that SNMP polling will fail at. And if you have multiple receivers, the router cal duplicate the packet for different destinations - which is a simple and efficient operation for a router.

Our Solution



Good: it works, we now have high precision metrics for all our core network

Bad: Kafka consumer rebalances stop processing pipeline