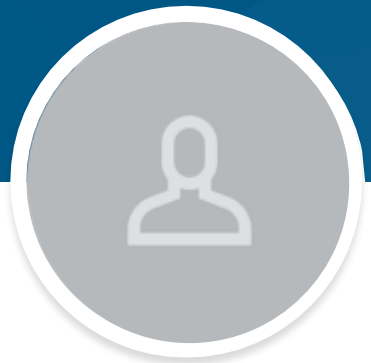
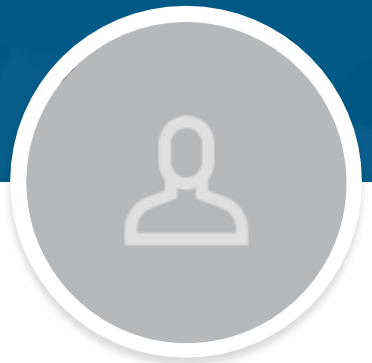
Network Telemetry in SONiC

(2018 August OCP workshop)



Jipan Yang
(Alibaba)



Zhenggen Xu
zxu@linkedin.com

What is network telemetry

- Full visibility of network status down to individual node is needed.
  - To fix or isolate network problem in a timely manner.
  - To take necessary preventive measures for network segment under pressure.

- Network Telemetry provides tools and interfaces to collect the network status data.

Common network telemetry methods

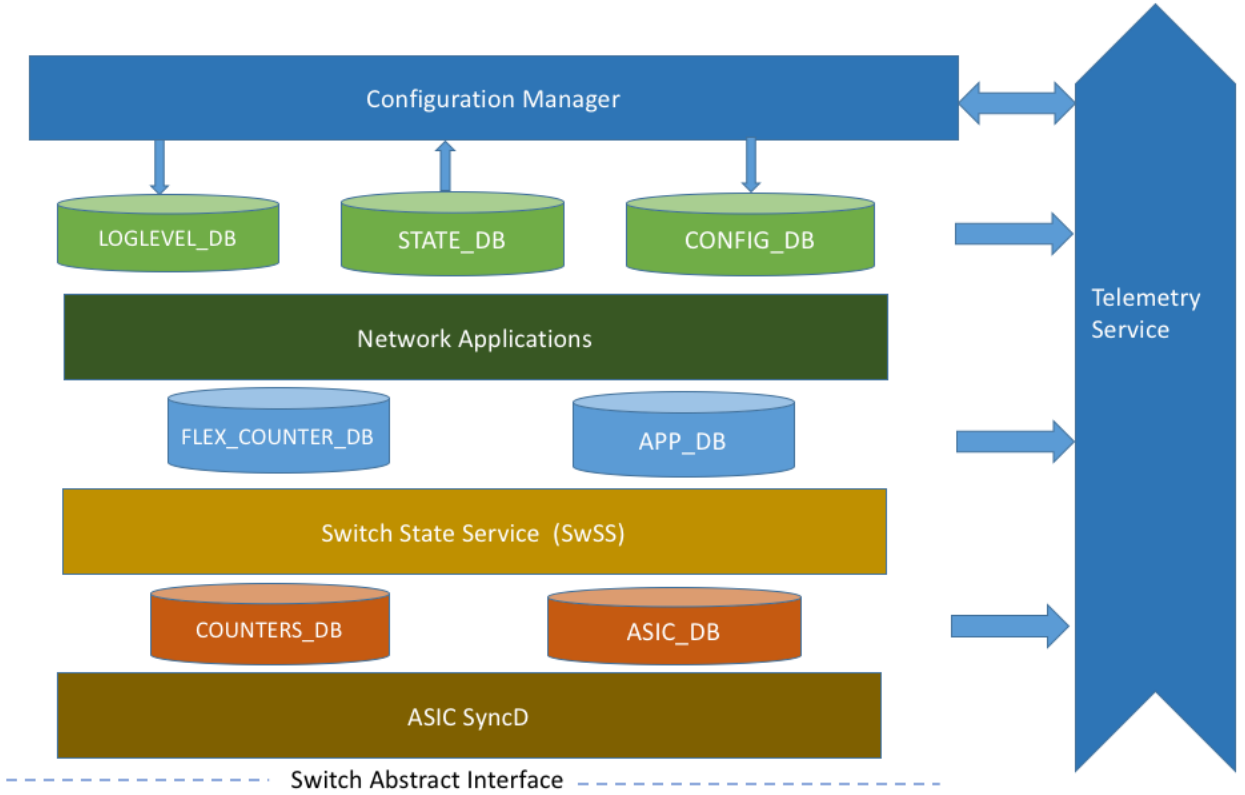
- SNMP
- Syslog
- CLI
- Packet mirroring
- Data plane telemetry (INT)
- …

Requirements for hype scale data center network telemetry

- Extensible and flexible data collection
  - Whatever, whenever and each node, to pin point any problem.
- Structured data
  - To easily consume and correlate.
  - Unstructured data like syslog is hard to analyze.
- Efficient data transportation and encoding
  - Well understood
  - Little overhead

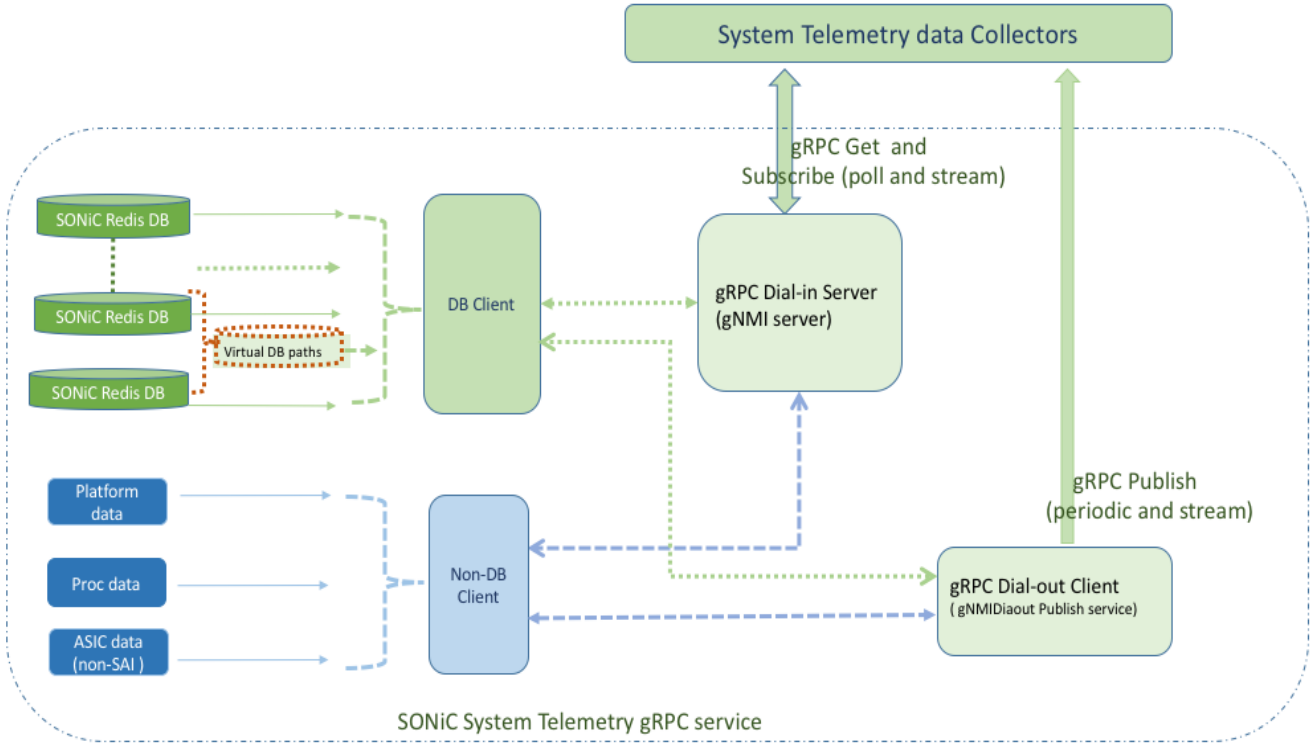
Network telemetry in SONiC

- SONiC is a database centric network OS.
  - Most of the critical control plane data could be found in redis DB.
  - The data in redis DB may be subscribed or polled as needed.
- SONiC telemetry service provides interfaces to collect control plane states.



Network telemetry in SONiC

- gRPC, the right framework for network telemetry
  - High performance.
  - Data streaming.
  - Tooling and libraries.
- gNMI pushed it further in the network domain for config and telemetry.
  - SONiC utilized gRPC with reference to gNMI for telemetry implementation
- Support both DB data and nonDB data



gRPC vs others

- gRPC vs snmp
  - Faster (1% cpu of white-box to poll ~6K counter per second)
  - Subscribe and streaming/event mode
  - Easy to expand
  - Security

- gRPC vs syslog/CLI
  - More data
  - Formatted data

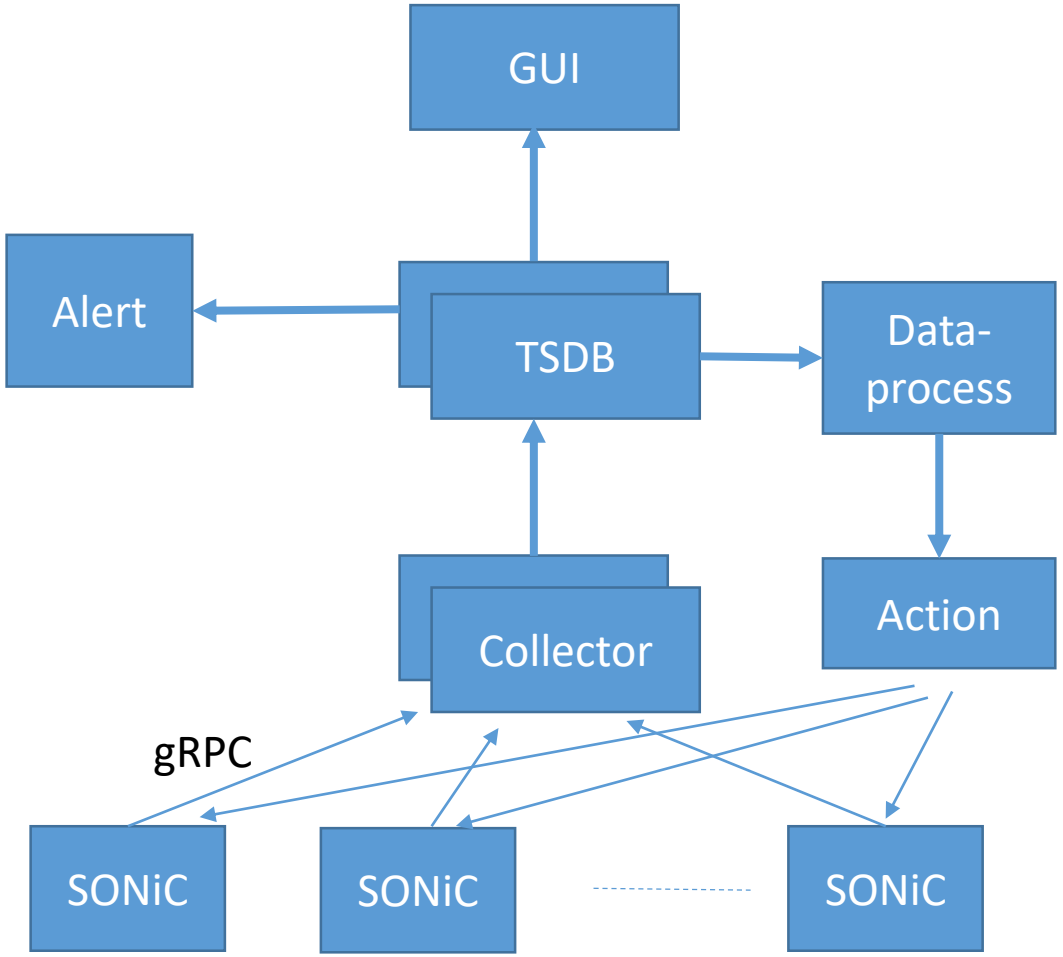
Database and Virtual Pathes

DB name	DB No.	Description
APPL_DB	0	Application running data
ASIC_DB	1	ASIC configuration and state data
COUNTERS_DB	2	Counter data for port, lag, queue
LOGLEVEL_DB	3	Log level control for SONiC modules
CONFIG_DB	4	Source of truth for SONiC configuration
FLEX_COUNTER_DB	5	For PFC watch dog counters control and other plugin extensions
STATE_DB	6	Configuration state for object in CONFIG_DB

DB target	Virtual Path	Description
COUNTERS_DB	"COUNTERS/Ethernet*"	All counters on all Ethernet ports
COUNTERS_DB	"COUNTERS/Ethernet*/ `<counter name>` "	One counter on all Ethernet ports
COUNTERS_DB	"COUNTERS/Ethernet `<port number>` / `<counter name>` "	One counter on one Ethernet port
COUNTERS_DB	"COUNTERS/Ethernet*/Queues"	Queues stats on all Ethernet ports
COUNTERS_DB	"COUNTERS/Ethernet `<port number>` /Queues"	Queue stats on one Ethernet ports

Collector, Data-Store and More

- Collector cluster collects all telemetry data through gRPC
  - Polling
  - Streaming
  - Dial in
  - Dial out
- Data is saved to time-series DB.
  - Schema-less
  - Preserve history
- Data displayed on GUI
- Rules defined to send events to alert system.
- Data could be used for data-processing (ML etc) and apply actions back to the devices.



Demo setup and Demo cases

- gRPC streaming mode is used.
- Components are built on top of open-source
- Built it fast!

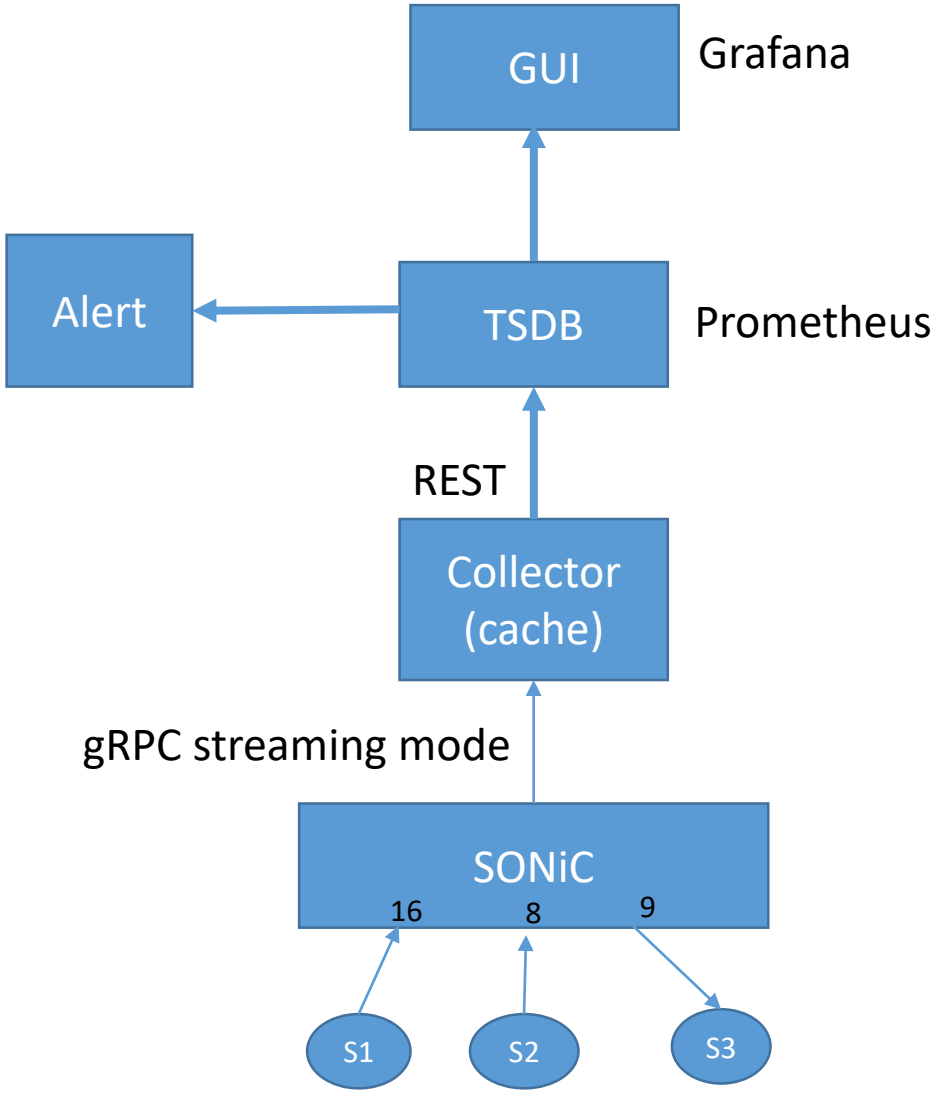
Demo scenarios:

- Traffic flowing from S1->S3, S2->S3

- Displaying Interface speed
  - Speed changes when changing the app data
  - Historic data

- Monitoring Link state changes
  - Flap the link once or multiple times
  - Interface speed changes
  - Link flap counters in 1m
  - Alert sent to email

- Monitoring route nexthop member counters
  - Add/delete nexthops in vtysh
  - Flap links so nexthops got added/deleted automatically



Demo snapshot

Rules:

InterfaceFlaps (1 active)

```
alert: InterfaceFlaps
expr: changes(oper_status[1m])
  / 2 >= 5
for: 20s
labels:
  severity: ticket
annotations:
  summary: interface flaps.
```

Email alerts received: