Visualizing Telemetry with TIG: Deployment and Concepts

Hasan-Uz-Zaman Ashik

Fiber@Home Limited.

Agenda

- Network Monitoring Technologies.
- Introduction to TIG Stack.
- What is Telemetry?
- Drawback of SNMP.
- Need for Telemetry based monitoring.
- Micro Burst traffic monitoring.
- Customized threshold based alert generation.
- Legacy device monitoring support in TIG stack using SNMP.
- Things to consider before deployment.

Network Monitoring Technologies

➢1990s & 2000s:

- Simple Network Management Protocol (SNMP) became widely adopted for network monitoring.
- RMON (Remote Monitoring) was introduced, allowing for remote monitoring of network devices and traffic.
- NetFlow was introduced by Cisco, providing detailed network traffic analysis and flow monitoring.
- Application Performance Monitoring (APM) emerged as a new focus in network monitoring, providing application-specific monitoring and user experience monitoring.

Network Monitoring Technologies(Contd.)

➢2010s:

- Flow-based monitoring became more popular, providing comprehensive network traffic analysis and real-time monitoring.
- Telemetry emerged as a new technology in network monitoring, offering real-time visibility and more efficient troubleshooting.

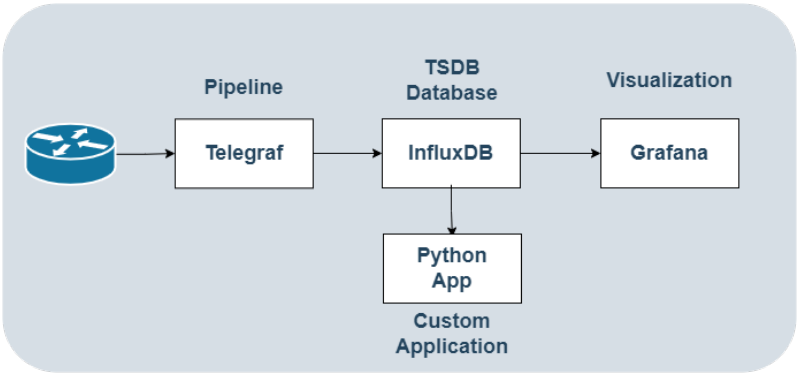
➢2020s:

- Cloud-based network monitoring became increasingly important, offering scalability and ease of deployment.
- New technologies continue to emerge, such as 5G network monitoring and software-defined networking (SDN) monitoring.

Introduction to TIG Stack

TIG stack is a combination of three open-source tools used for monitoring and analyzing data in real-time:

- **Telegraf**: A plugin-driven server agent used for collecting and reporting metrics from various sources, including systems, applications, and databases.
- **InfluxDB**: A time-series database used for storing and querying large amount of data in real-time.
- **Grafana**: A data visualization and analysis platform used for creating and sharing real-time dashboards, and alerts.



What is Telemetry?

Telemetry is a remote data collection technology that provides real-time, high-speed, and accurate network monitoring.

It uses YANG models to organize data and encodes it in GPB format, transmitted through gRPC protocol, improving data collection efficiency and facilitating intelligent interconnection. Unlike traditional technologies that interact in pull mode, Telemetry can operate in push mode.

Drawback of SNMP

- Due to pull model, collector initiates request and device process it to provide response.
- Cannot provide data in millisecond interval.
- Due to periodic query-based system, it cannot accurately monitor the network status.
- MIB is unstructured, that creates overhead for collector.

Why Telemetry?

Advantages of Telemetry over traditional network monitoring technologies:

- Pushes data periodically in milliseconds.
- Streams data, gRPC establishes a single long-lived TCP connection between the router and receiver for telemetry streaming.
- Supports advanced monitoring i.e. application-specific monitoring.
- Easily scalable as device initiates the connection.
- Uses YANG models to organize data structurally, resulting in improved data collection.
- Encodes data in the Google Protocol Buffers (GPB) format, which provides an efficient, flexible, and scalable solution for data serialization.
- Real-time traffic optimization.

Methods of Telemetry

There are two methods to stream telemetry data:

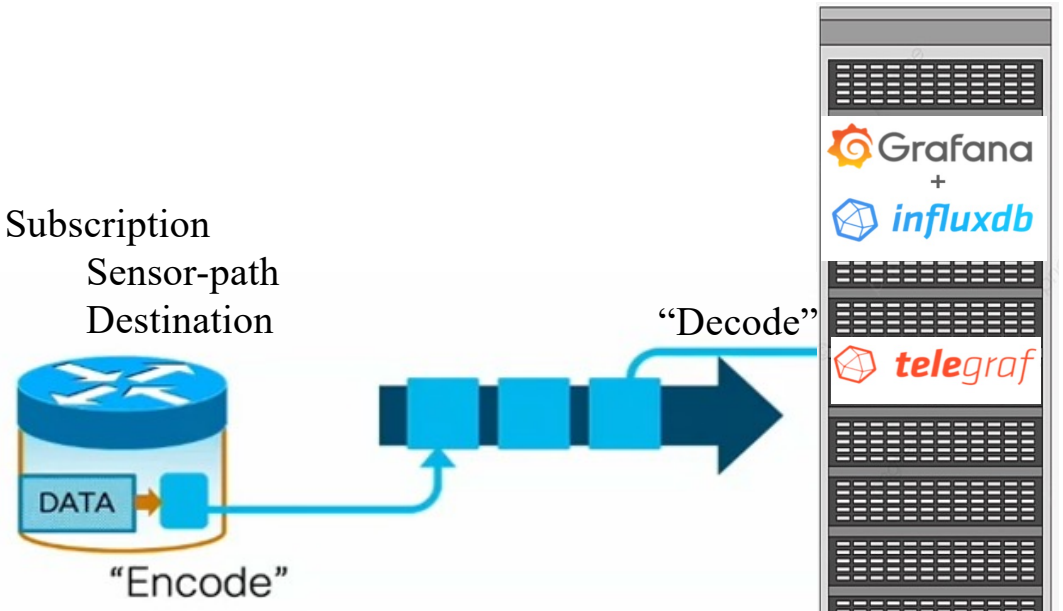
- Model-driven telemetry:

  Subscription based streaming from an MDT-capable device.

- Policy-based telemetry:

  Data and frequency is defined in a policy file.

Device to Telegraf data flow

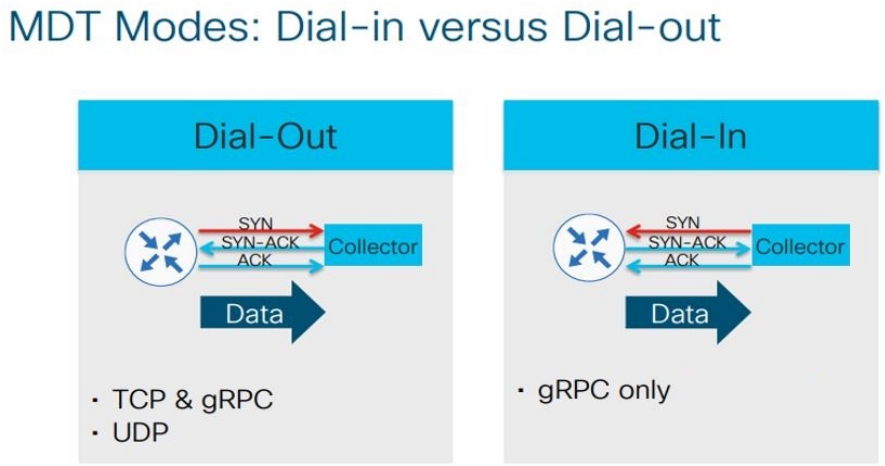


- ✓ GPB-Compact
- ✓ GPB-KV
- ✓ JSON

GPB is wire effective (low BW)
JSON is human readable but consumes high BW

Three options for Transport

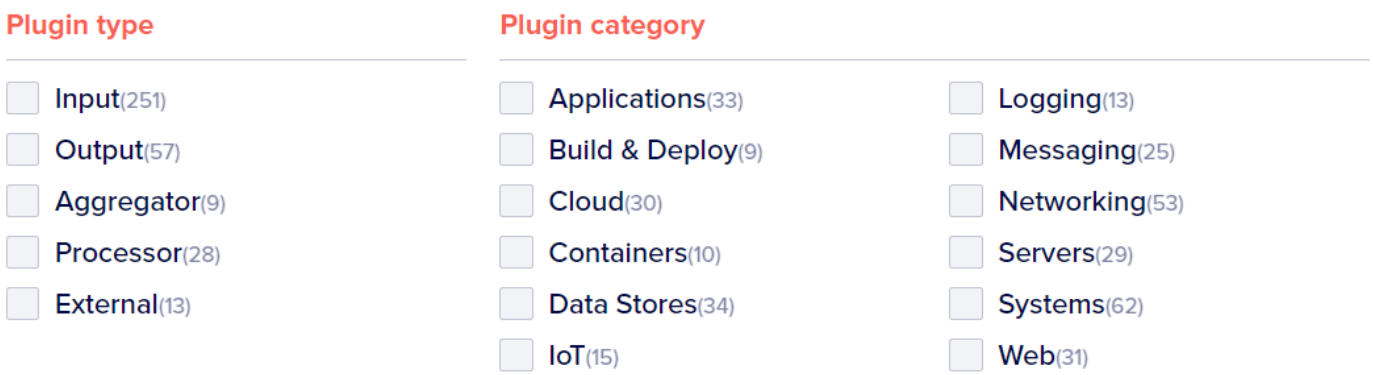
- TCP dial-out,
- gRPC dial-out, and
- gRPC dial-in.



- Dial out: Router “dials out” to the collector. TCP handshake is initiated by the router.
- Dial In: Router listens passively on a specified port until the collector “dials-in.”
- No telemetry configuration is required on the router for Dial-In since the connection is initiated by the server.
- Dial-Out is preferred because it offers greater flexibility.

Telegraf configuration

- Written in Go language
- Plugin-driven (input, output, aggregator, and processor)
- Configuration file is written in TOML format



Telegraf configuration(Contd.)

- Telegraf agent global configuration

```
# Configuration for telegraf agent
[agent]
  ## Default data collection interval for all inputs
  interval = "300s"
  round_interval = true
  metric_batch_size = 5000
  metric_buffer_limit = 50000
  collection_jitter = "0s"
  flush_interval = "600s"
  flush_jitter = "0s"
  precision = ""
  logfile = "/var/log/telegraf/snmp_influxv2_huawei_optics.log"
  logfile_rotation_interval = "168h"
  logfile_rotation_max_archives = 5

  ## Override default hostname, if empty use os.Hostname()
  hostname = ""
  ## If set to true, do no set the "host" tag in the telegraf agent.
  omit_hostname = false
```

- Output plugin

```
[[outputs.influxdb_v2]]
  urls = ["http://192.168.41.2:8086"]
  ## Token for authentication.
  token = "$INFLUX_TOKEN"
  ## Organization is the name of the organization
  organization = "Fiber@Home"
  ## Destination bucket to write into.
  bucket = "snmp"
```

Telegraf configuration(Contd.)

- Telemetry Input plugin configuration for Cisco-XR device

```
[[inputs.cisco_telemetry_mdt]]
 ## Telemetry transport can be "tcp" or "grpc".  TLS is only supported when
 ## using the grpc transport.
 transport = "grpc"

 ## Address and port to host telemetry listener
 service_address = ":57000"
 max_msg_size = 20000000

 ## Define aliases to map telemetry encoding paths to simple measurement names
 [inputs.cisco_telemetry_mdt.aliases]
   ifstats = "ietf-interfaces:interfaces-state/interface/statistics"
 ##Define Property Xformation, please refer README and https://pubhub.devnetcloud.com/media/dme-docs-9-3-3/docs/appendix/
 [inputs.cisco_telemetry_mdt.dmes]
   ModTs = "ignore"
   CreateTs = "ignore"
```

Sample Configuration in Cisco-XR

```
telemetry model-driven
 destination-group DGroup1
  address-family ipv4 server_ip port 57000
   encoding self-describing-gpb
   protocol grpc no-tls
  !
 !
 sensor-group optics
  sensor-path Cisco-IOS-XR-controller-optics-oper:optics-oper/optics-ports
  sensor-path Cisco-IOS-XR-controller-optics-oper:optics-oper/optics-ports/optics-port/optics-info
  sensor-path Cisco-IOS-XR-controller-optics-oper:optics-oper/optics-ports/optics-port/optics-lanes/optics-lane
 !
 sensor-group interfaces
  sensor-path Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-summary
  sensor-path Cisco-IOS-XR-bundlemgr-oper:bundles-adjacency/nodes/node/brief
  sensor-path Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters
  sensor-path Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/member-interfaces/member-interface/output
 !
 subscription optics_sub
  sensor-group-id optics sample-interval 900000
  destination-id DGroup1
 !
 subscription interfaces_sub
  sensor-group-id interfaces sample-interval 300000
  destination-id DGroup1
 !
!
```

Subscription status

```
RP/0/RP0/CPU0:CT-CT-GP-CGORN2#sh telemetry model-driven subscription interfaces_sub
Wed Apr 19 03:08:35.684 +06
Subscription:  interfaces_sub
------------
  State:       ACTIVE
  Sensor groups:
  Id: interfaces
    Sample Interval:      300000 ms
    Heartbeat Interval:   NA
    Sensor Path:          Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-summary
    Sensor Path State:    Resolved
    Sensor Path:          Cisco-IOS-XR-bundlemgr-oper:bundles-adjacency/nodes/node/brief
    Sensor Path State:    Resolved
    Sensor Path:          Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters
    Sensor Path State:    Resolved
    Sensor Path:          Cisco-IOS-XR-qos-ma-oper:qos/interface-table/interface/member-interfaces/member-interface/output
    Sensor Path State:    Resolved

  Destination Groups:
  Group Id: DGroup1
    Destination IP:       10.249.0.6
    Destination Port:     57000
    Encoding:             self-describing-gpb
    Transport:            grpc
    State:                Active
    TLS :                 False
    Total bytes sent:     112533773
    Total packets sent:   5718
    Last Sent time:       2023-04-19 03:06:21.2918417573 +0600
```

What kind of data does the sensor-path transmit?

Command: run mdt_exec -s Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters


```
{
   "node_id_str":"CT-CT-GP-CGORN2",
   "subscription_id_str":"app_TEST_200000001",
   "encoding_path":"Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters",
   "collection_id":"2161540",
   "collection_start_time":"1681852367480",
   "msg_timestamp":"1681852367571",
   "data_json":[
      {
         "timestamp":"1681852367504",
         "keys":[
            {
               "interface-name":"GigabitEthernet0/0/0/0"
            }
         ],
         "content":{
            "packets-received":"16818121741",
            "bytes-received":"3822393205444",
            "packets-sent":"34553437950",
            "bytes-sent":"26786744275107",
            "multicast-packets-received":"9008",
            "broadcast-packets-received":"153095",
            "multicast-packets-sent":"0",
            "broadcast-packets-sent":"2280",
            "output-drops":0,
            "output-queue-drops":0,
            "input-drops":0,
            "input-queue-drops":0,
            "runt-packets-received":0,
            "giant-packets-received":0,
            "throttled-packets-received":0,
            "parity-packets-received":0,
            "unknown-protocol-packets-received":0,
            "input-errors":0,
            "crc-errors":0,
            "input-overruns":0,
            "framing-errors-received":0,
            "input-ignored-packets":0,
            "input-aborts":0,
            "output-errors":0,
```

InfluxDB terminologies

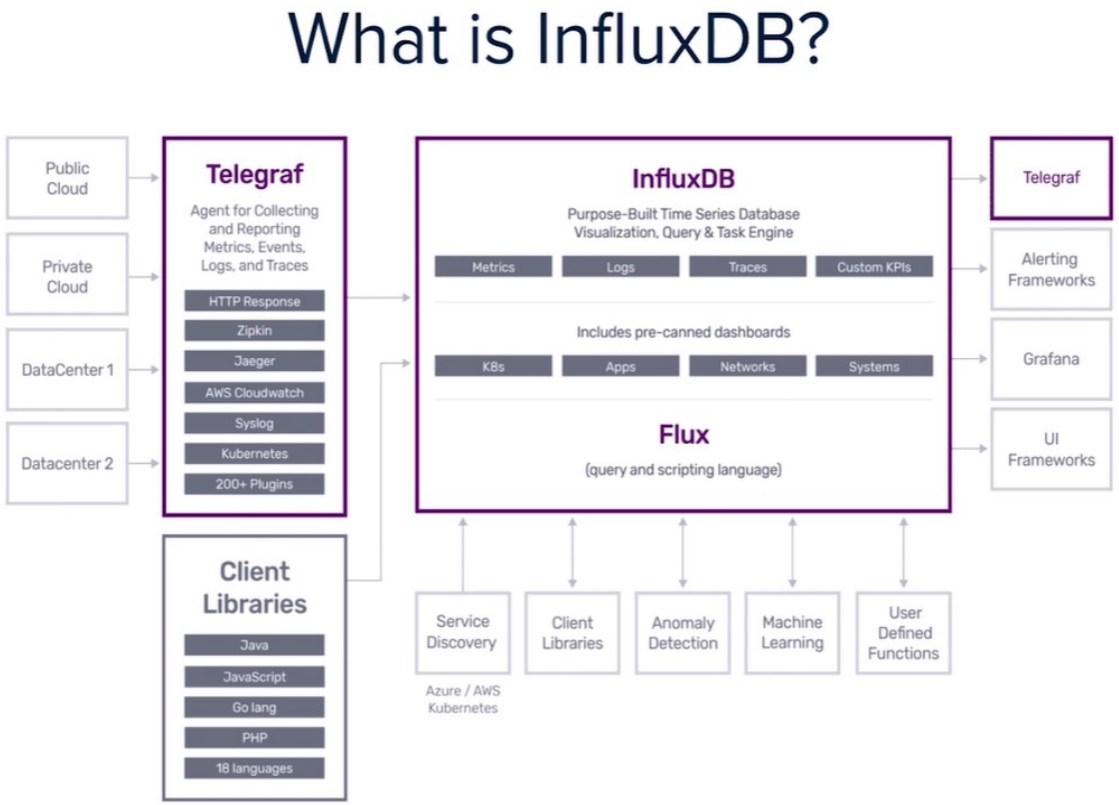
Not a Relational Database

RDBMS

- Databases
- Tables
- Columns
- Rows
- Row-oriented

InfluxDB

- Buckets
- Measurements
- Tags & Fields
- Points
- Column-oriented



InfluxDB line protocol

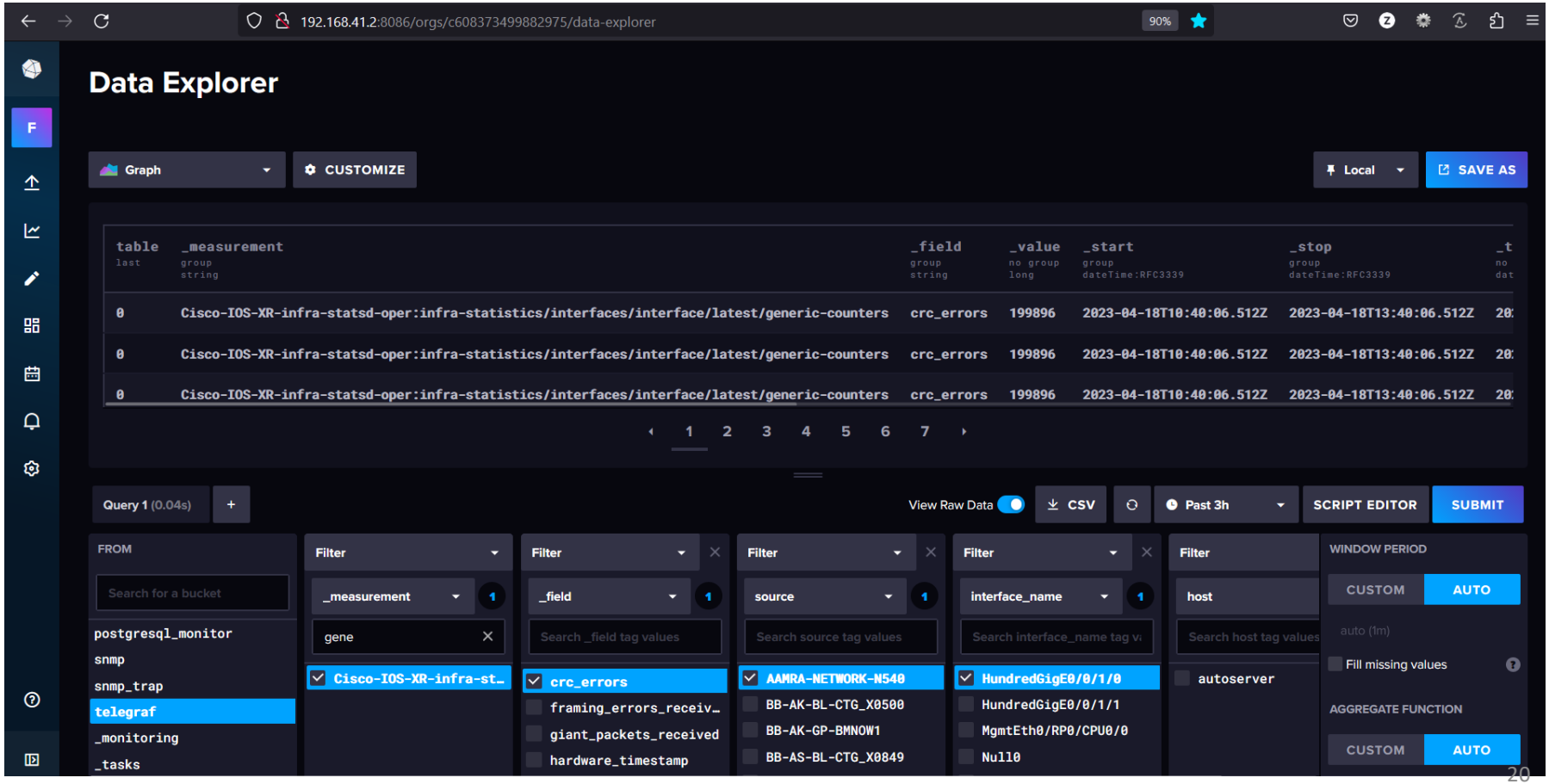
InfluxDB line protocol is a text-based format used for writing data into InfluxDB. It is a compact and efficient way of representing data points in a single line, making it ideal for high-volume data ingestion.

Telegraf sends data to InfluxDB in line protocol format after applying configured preprocessing of raw data.

Format: measurement[,tag_name=tag_value] field1=value1,field2] [timestamp]

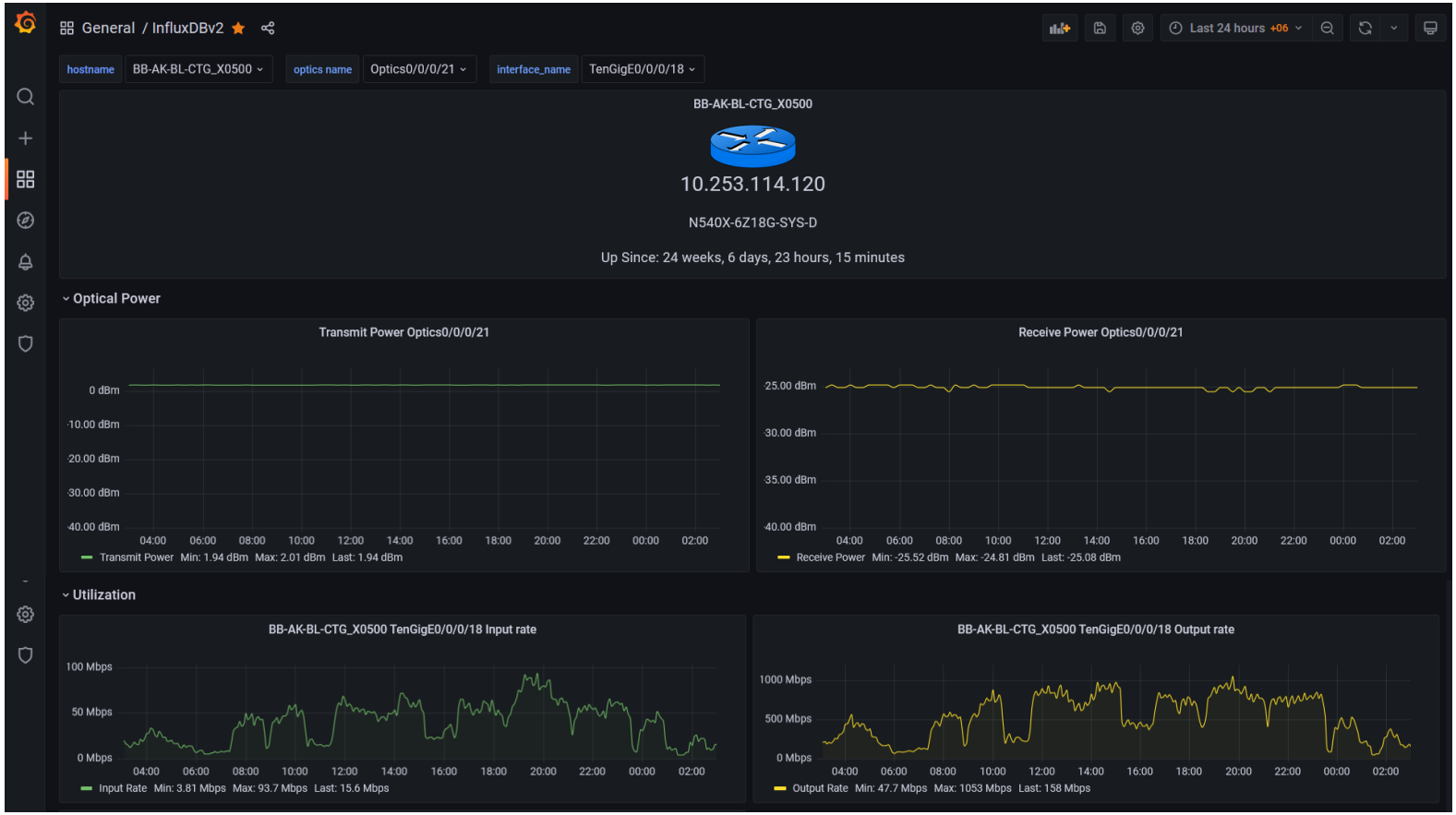
Example: telegraf,location=Dhaka temp=45,humidity=60 1641417317000000

InfluxDB matrix view



Grafana Dashboard of Cisco XR Router

Grafana Dashboard of Cisco XR Router(Contd.)



Flux Query to build dashboards

A InfluxDBv2_40.35

```
from(bucket: "telegraf")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters")
  |> filter(fn: (r) => r["_field"] == "bytes_sent")
  |> filter(fn: (r) => r["source"] == "$hostname")
  |> filter(fn: (r) => r["interface_name"] == "$interface_name")
  |> aggregateWindow(every: 5s, fn: last, createEmpty: false)
  |> difference(nonNegative: true)
  |> map(fn: (r) => ({r with _value: float(v: r._value*8) / (5000000.00)}))
  |> yield(name: "Input_rate")
```

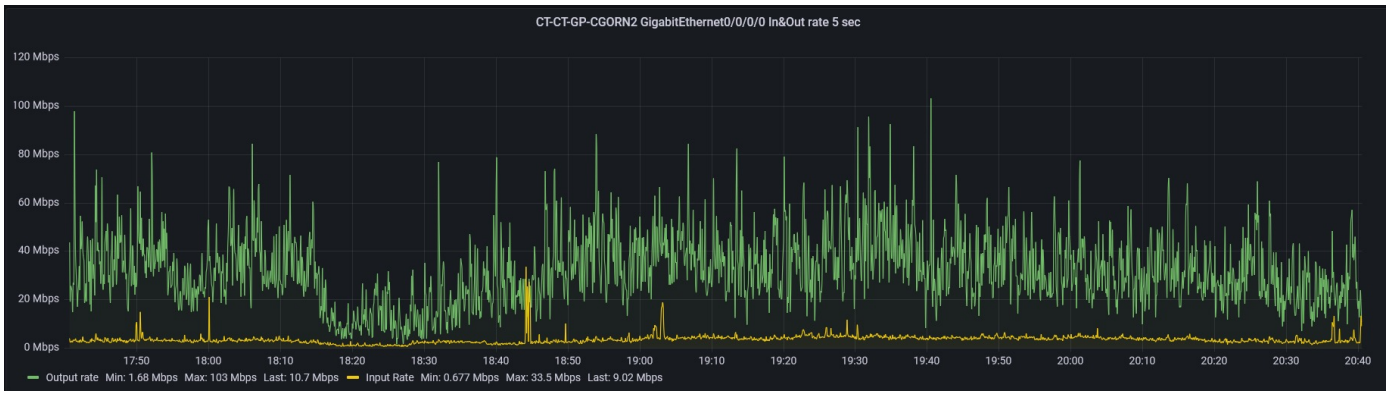
This query will

- Get data from ..*generic-counters* measurement
- Filter selected interface out direction traffic
- Calculate non-negative difference between counter values
- Convert to Mbps unit at 5 sec interval

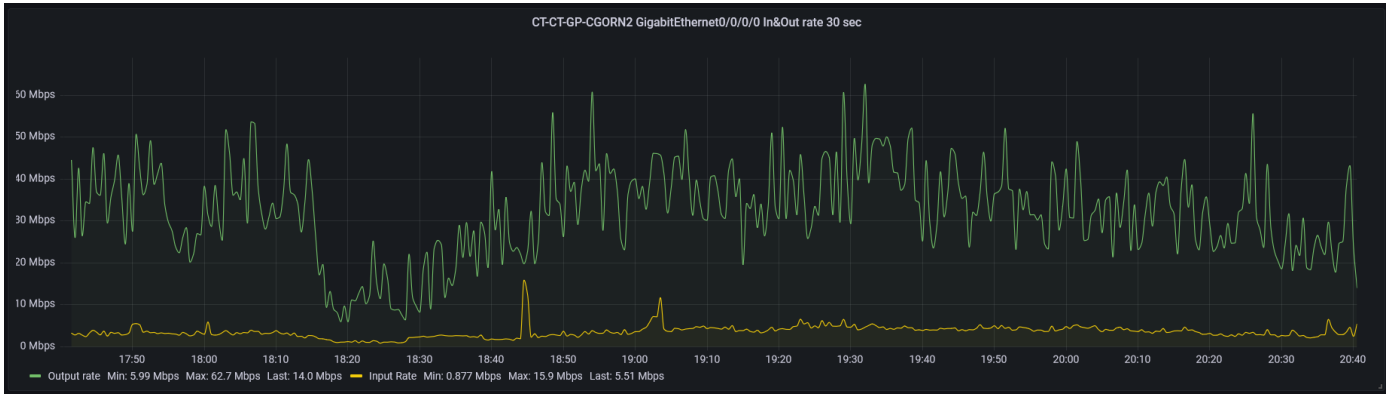
Applications of TIG stack and Telemetry

- Large scale high resolution real-time traffic monitoring.
- Microburst traffic detection.
- Customized alarm generation.
- SNMP & Telemetry monitoring from single platform.
- Secured application development using InfluxDB real-time data.

Monitoring with high resolution

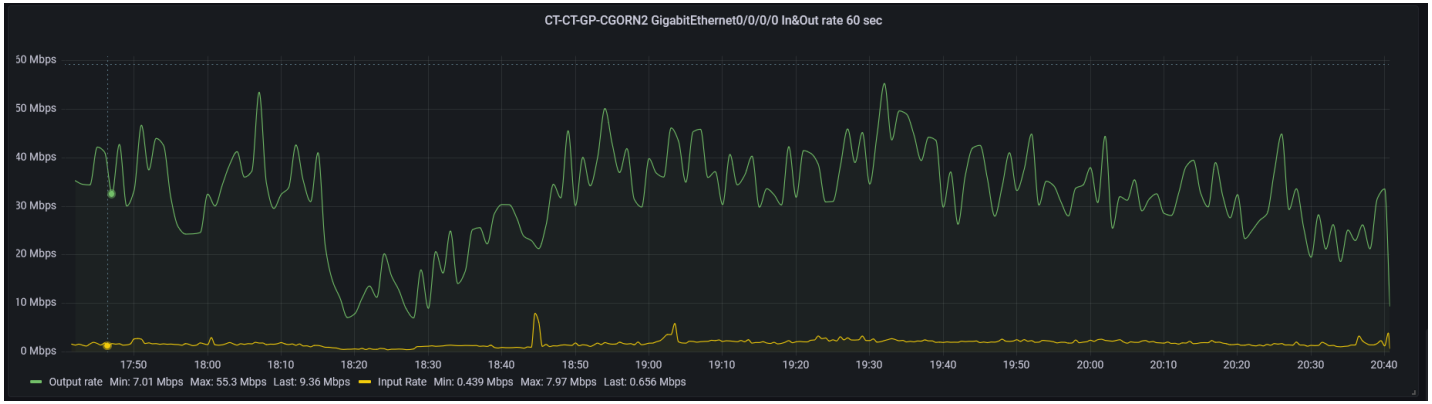


5 sec

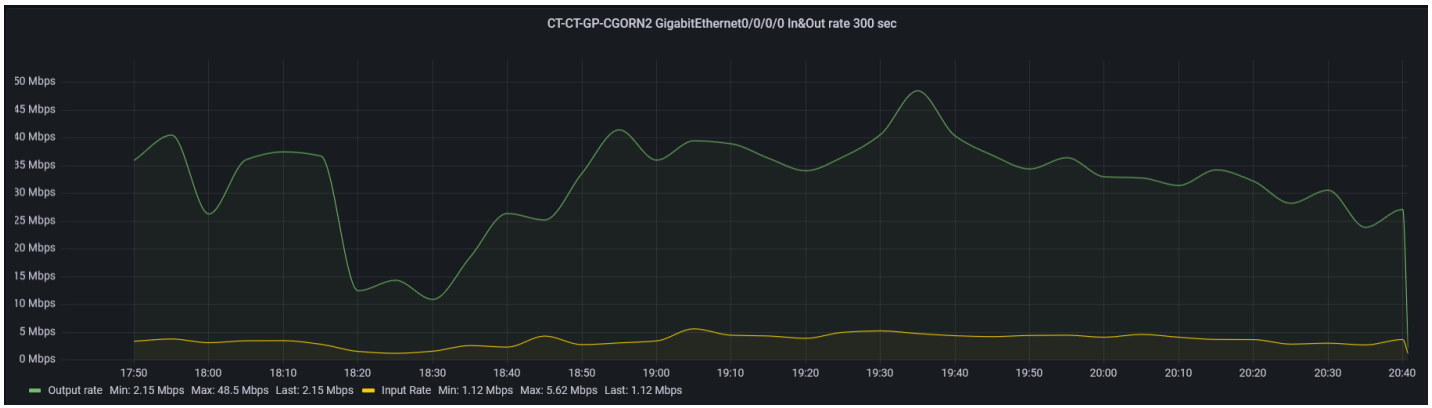


30 sec

Monitoring with high resolution(Contd.)



60 sec



300 sec

Monitoring with high resolution(Contd.)

- Identify microburst traffic, potential network congestion and bottlenecks.
- As the number of microbursts increases, the service retransmission rate goes up, leading to a decline in the quality of network communication.
- Capacity planning and ensure that network has enough resources to meet the demands of users and applications.

Customized alarm generation

- A Python script is created to retrieve real-time optics data from InfluxDB. The script then generates an organized alarm by verifying the threshold value of each link in a separate Django web application, which is intended for the NOC team to take action for optical power degradation incidents.

Optical Power Alarm Management(High Loss)

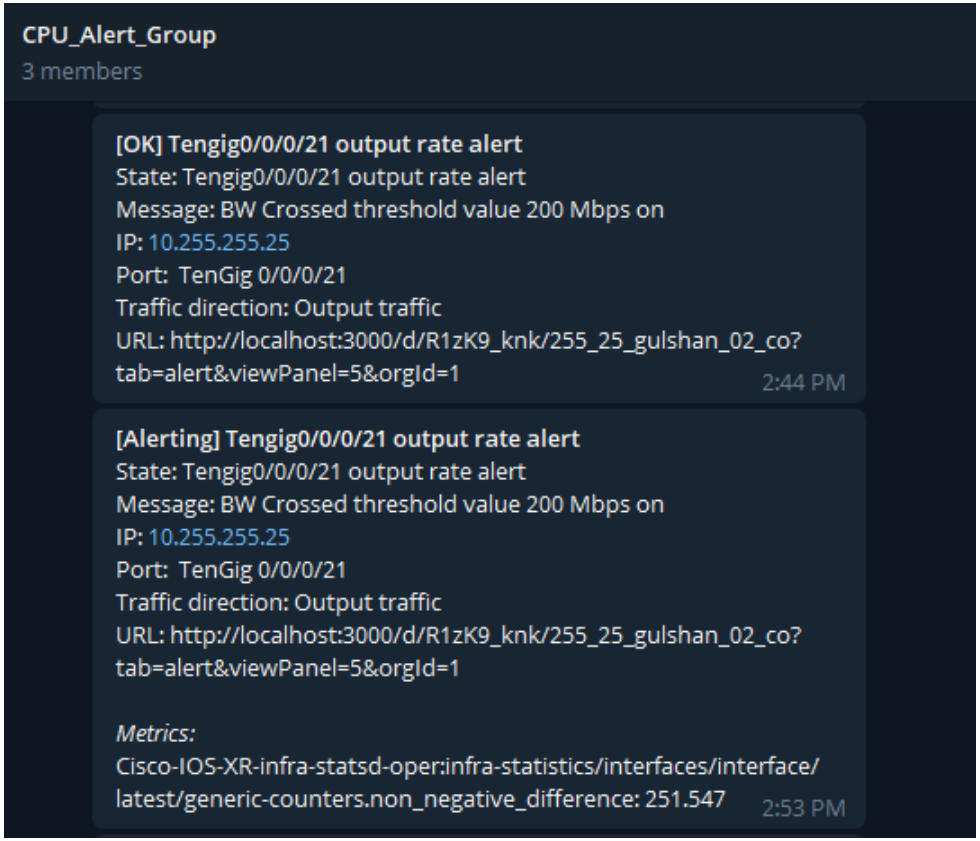
Total links in monitoring:	Total device in monitoring:	Total Alarm(s):	Cleared Alarm(s):
5758	1112	60	1991

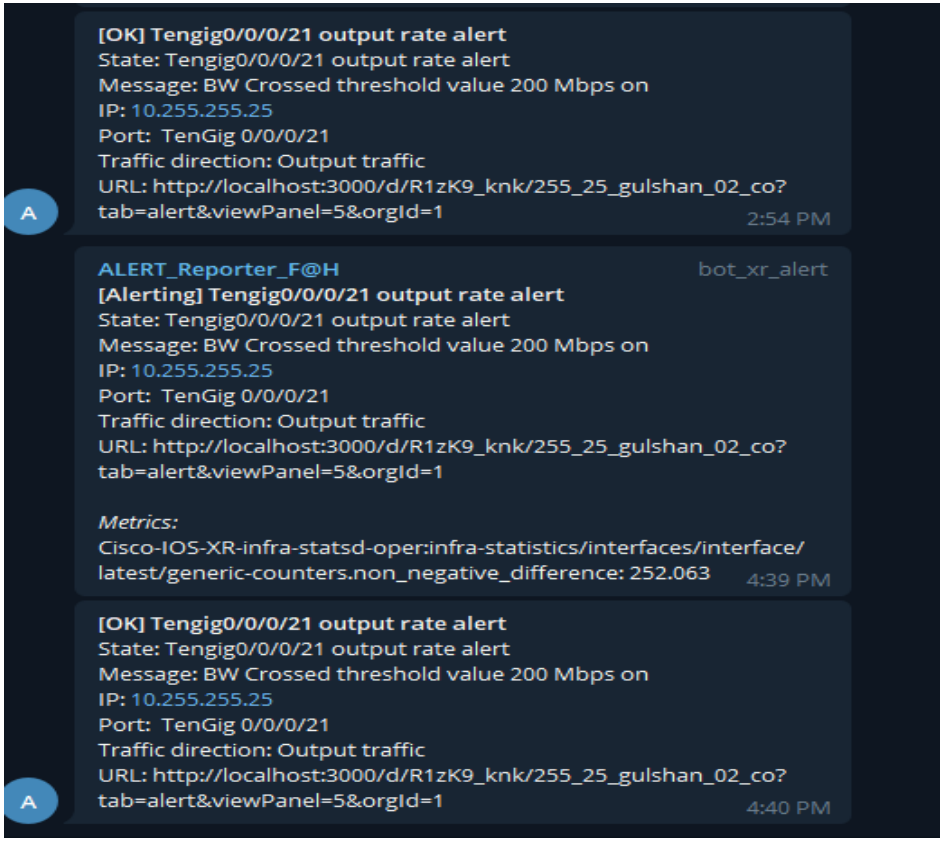
Choose link type: All Choose Operator: All

Excel Search:

Serial	Alarm ID	Priority	Alarm Status	ACK Status	Alarm Time	IP	Interface	Hostname	TX	RX	THG
60	202304180097	4	RUNNING		April 18, 2023, 8:43 p.m.	10.253.231.231	GigabitEthernet0/0/0/5	DH-GU-RB-DHGULP6	-30.45	-32.21	-19.0
59	202304180093	4	RUNNING		April 18, 2023, 7:32 p.m.	10.253.105.90	GigabitEthernet0/0/0/17	RB-CTG-CHOWKBAZAR_WIC	-6.66	-28.23	-26.0
58	202304180074	3	RUNNING		April 18, 2023, 4:19 p.m.	10.253.199.35	TenGigE0/0/0/22	SA-SHARIATPUR-CL-01-N540X2C-PE-01	1.49	-23.56	-23.0
57	202304180069	4	RUNNING		April 18, 2023, 3:58 p.m.	10.253.148.39	GigabitEthernet0/0/0/6	NW-SH-RB-NWSBG04	-6.44	-33.01	-11.0
56	202304180046	2	RUNNING		April 18, 2023, 11:57 a.m.	10.255.255.189	FortyGigE0/0/1/0	MU-SREENAGAR-CL-03-N5402C-PE-01	3.11	-18.79	-18.0
55	202304180026	4	RUNNING		April 18, 2023, 7:13 a.m.	10.255.255.111	GigabitEthernet0/0/0/7	RS-RAJ-CL-02-N5402C-PE-01	-7.44	-18.82	-18.0
54	202304180023	3	RUNNING		April 18, 2023, 6:56 a.m.	10.253.165.236	TenGigE0/0/0/11	ST-SATKHIRA-CL-01-N540X2C-PE-02	2.65	-18.26	-18.0

Alarm notify in Telegram group from Grafana

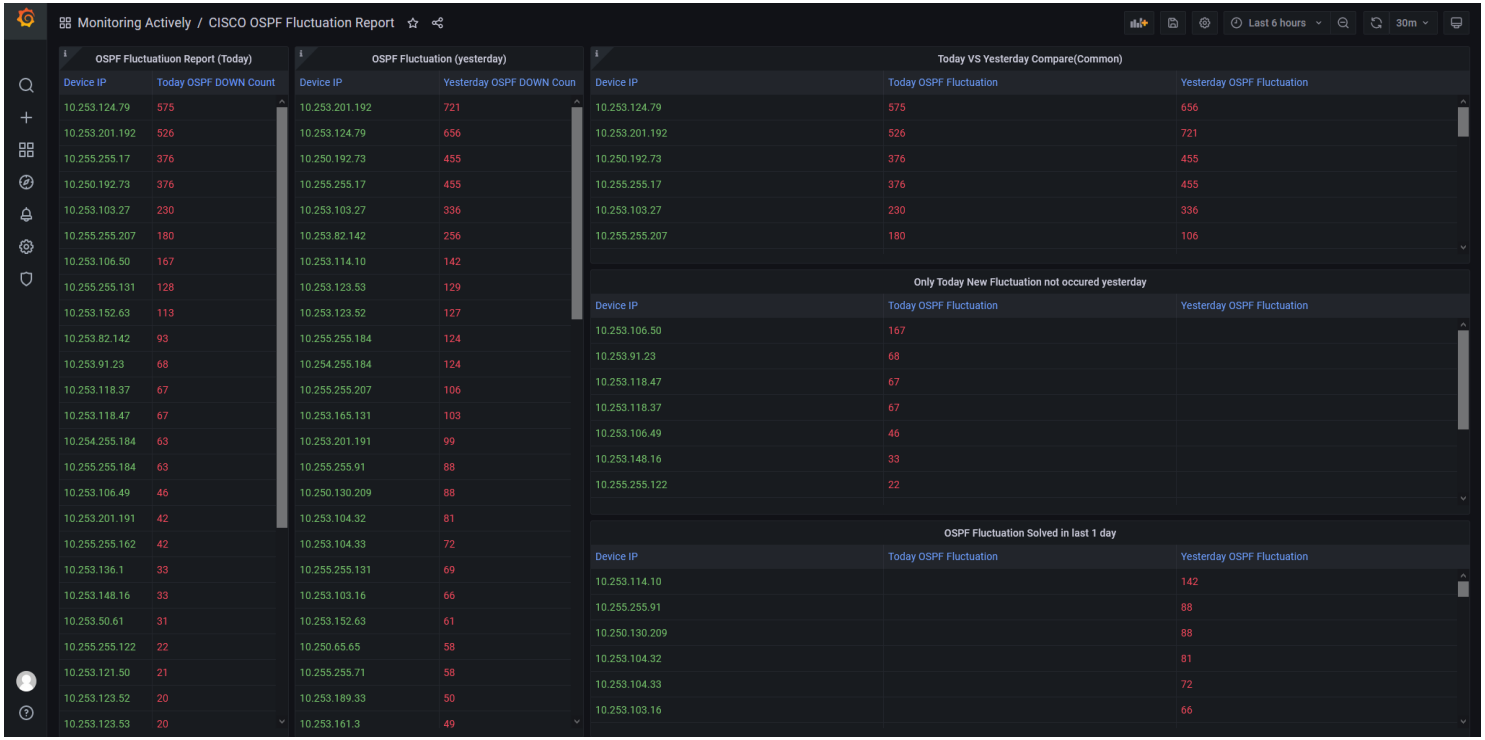




Real-time report from multiple databases

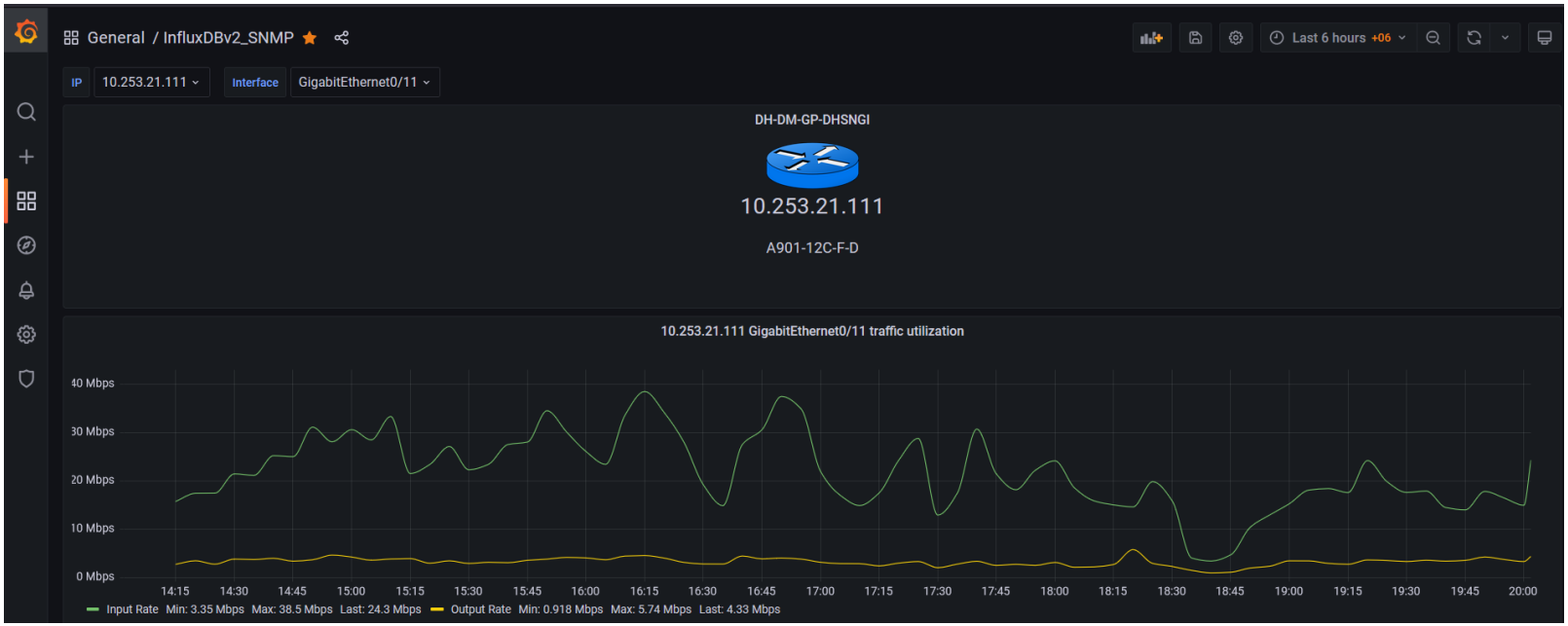
Grafana can be used for preparing reports from multiple data sources and combining results.

Example: SYSLOG database to Grafana OSPF flap reporting dashboard



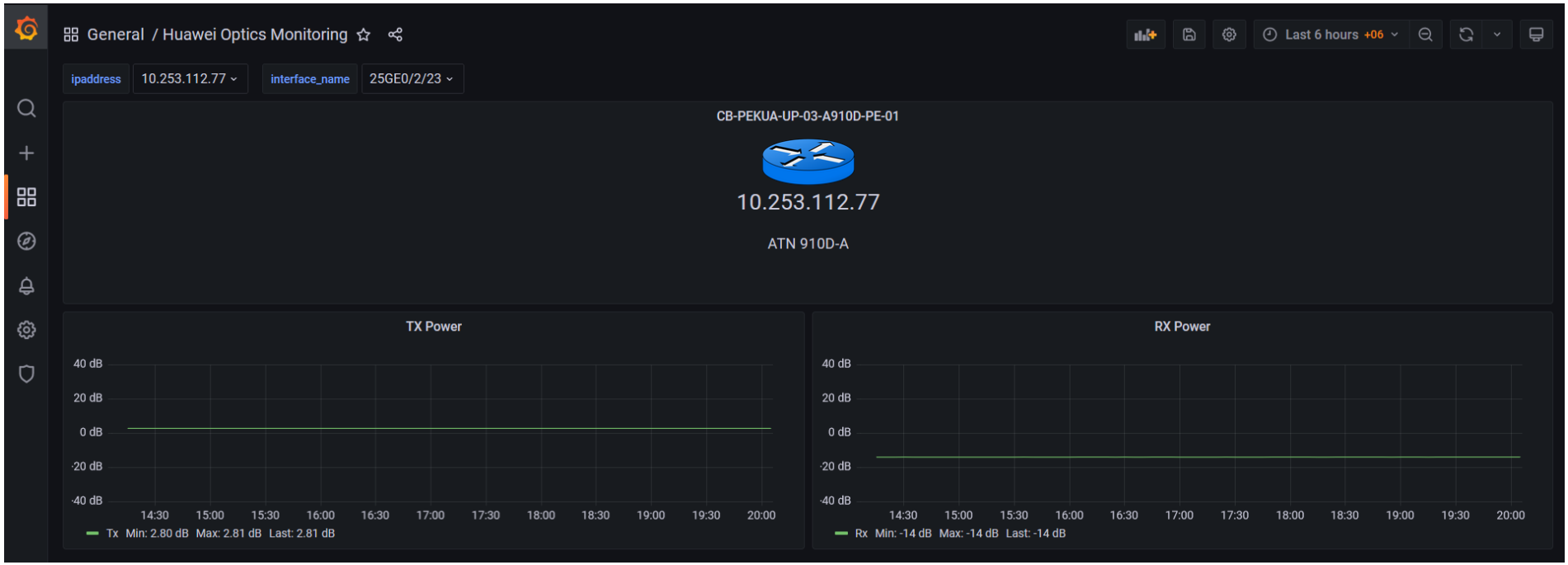
Legacy device monitoring with SNMP in TIG

- Devices that do not support Telemetry can also be integrated within TIG stack using SNMP.



Cisco A901-12C-F-D interface utilization monitoring using SNMP in Grafana

Legacy device monitoring with SNMP(Contd.)



Huawei ATN 910D-A model router optics monitoring in Grafana using SNMP

Things to consider before deployment

- InfluxDB version 1.x or 2.x, scalability.
- Series cardinality optimization.
- Higher CPU, RAM, IOPS for OSS/Enterprise solutions.

References

- Telemetry Configuration Guide for Cisco *https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/telemetry/b-telemetry-cg-ncs5500-62x/b-telemetry-cg-ncs5500-62x_chapter_01.html*
- Telegraf plugins *https://docs.influxdata.com/telegraf/v1.26/plugins/*
- InfluxDB *https://www.influxdata.com/*
- Grafana *https://grafana.com/oss/grafana/*
- OpenConfig *https://openconfig.net/projects/models/schemadocs/yangdoc/openconfig-telemetry.html*

Any