Next Generation Network Telemetry

Exploring Technologies and Advancements in Standardization

Bologna, ITNOG 9-10 May 2023

Paolo Lucente
Salvatore Cuzzilla

■ whoami

- Salvatore Cuzzilla
- System Engineer @Swisscom (AS3303)

in https://www.linkedin.com/in/salvatore-cuzzilla/
https://github.com/scuzzilla
✉ salvatore@cuzzilla.org

■ whoami

- Paolo Lucente
- Network Tools Engineer @NTT (AS2914)

https://www.linkedin.com/in/plucente
https://github.com/paololucente
✉ paolo@pmacct.net

■ Streaming Telemetry

is the commercial term used to describe the idea of 「continuously streaming structured data」

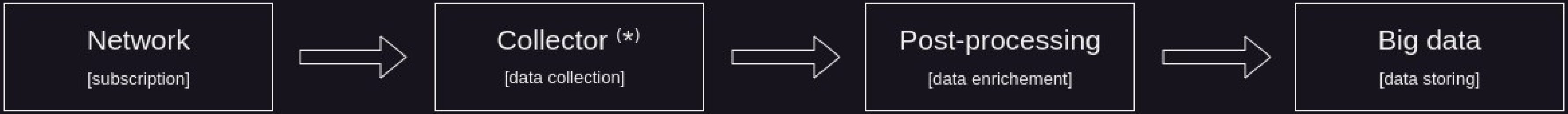
- Continuous, ***real-time transmission*** of data from the network devices to the data collector
- The data is organized according to a ***predefined schema*** 「*YANG models*」 , allowing for easy processing

■ Modeled Data

- Configuration Data settings & parameters configured on network devices to help network administrators to ensure ***consistent configurations*** across the network

- Operational *(or State)* Data real-time performance and state information of network devices to help network administrators to monitor the ***health and performance*** of the network

〈Modeled〉 Data collection



Network

[subscription]

Collector (*)

[data collection]

Post-processing

[data enrichement]

Big data

[data storing]

(*) pmacct 〈multi-protocol data collection & delivery〉

- Libpcap
- Netflow/IPFIX/sFlow
- BGP/BMP
- **YANG-modeled data streaming** 「gRPC dial-out」

■ YANG-modeled data streaming: desirable features

- Efficiency & performance ***low latency and high throughput*** to transmit telemetry data in real-time and minimize the overhead on network devices and monitoring systems

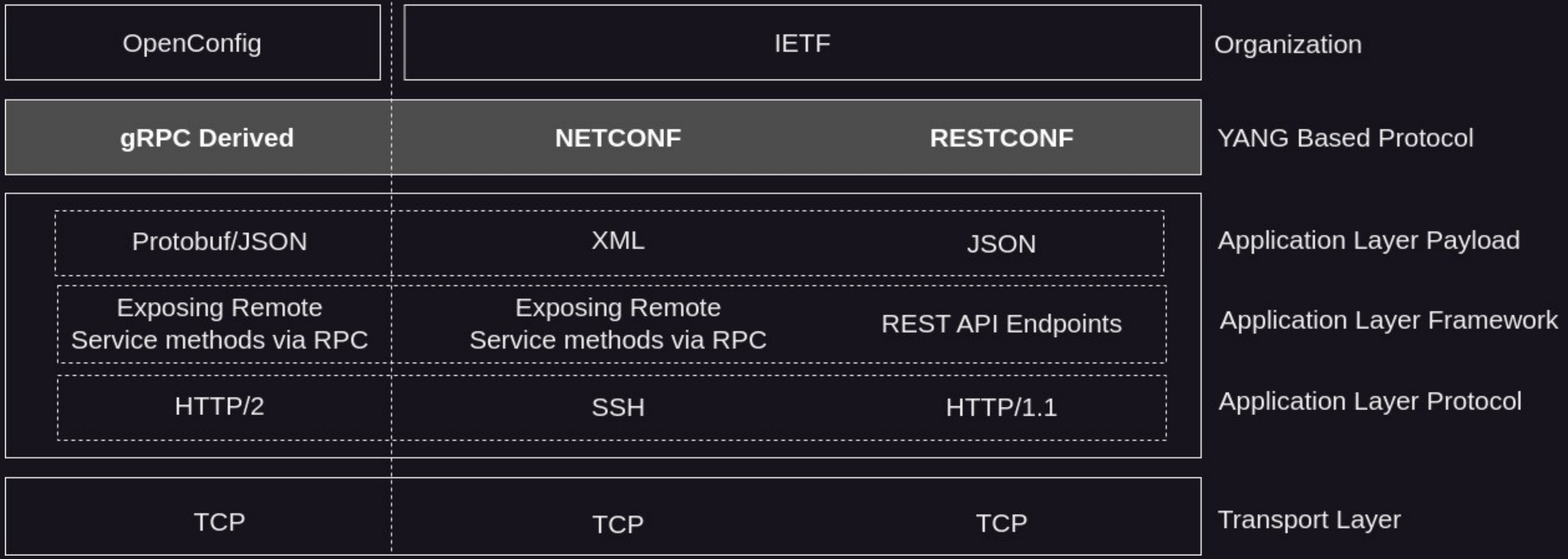
- Scalability handling large-scale network environments, supporting ***numerous devices and data points simultaneously***, without affecting performance or data transmission quality

- Security secure communication between network devices and monitoring systems to protect sensitive data and ***maintain network integrity***

- Streaming capabilities continuous streaming of data, as well as event-driven notifications and on-demand data retrieval, to provide a ***comprehensive and real-time view*** of the network

- Interoperability & standardization ***based on open standards and easy to integrate*** with various network devices, monitoring tools, and management systems

YANG-modeled data streaming: Protocols Comparison



OpenConfig	IETF		Organization
gRPC Derived	**NETCONF**	**RESTCONF**	YANG Based Protocol
Protobuf/JSON	XML	JSON	Application Layer Payload
Exposing Remote Service methods via RPC	Exposing Remote Service methods via RPC	REST API Endpoints	Application Layer Framework
HTTP/2	SSH	HTTP/1.1	Application Layer Protocol
TCP	TCP	TCP	Transport Layer

▸ Interoperability & standardization	IETF > OpenConfig
▸ Efficiency & performance	Protobuf > JSON > XML
▸ Scalability	HTTP/2.0 > HTTP/1.0 > SSH
▸ Security	TLS =
▸ Streaming capabilities	gRPC derived > RESTCONF > NETCONF

■ gRPC ⟨SW Development perspective⟩ features

- Cross-platform & *language support* ⟨C++, Java, Python, Ruby, ...⟩
- API contract definition ⟨GPB schema file defines the ***API contract*** between client and server⟩
- Code generation ⟨***gRPC compiler*** is generating a "skeleton" from a GPB schema⟩

■ gRPC/MDT(*) dial-out collector

https://github.com/network-analytics/mdt-dialout-collector

- Developed in C/C++ ⟨ leveraging grpc.io Asynchronous-API ⟩
- Dial-out ⟨ the subscription to a specific data stream is a prerequistite ⟩
- Continuous streaming ⟨ support only periodic subscription ⟩

(*)Model-Driven Telemetry

■ gRPC/MDT dial-out collector

https://github.com/network-analytics/mdt-dialout-collector

- Multi-vendor gRPC dial-out support
  - Cisco-XR ⟨7.4.1@NSC-54⟩
  - Cisco-XE ⟨17.06.01prd7@C8000V⟩
  - Junos ⟨20.4R3-S2.6@mx10003⟩
  - Huawei VRP ⟨V800R021C10SPC300T@NE40E⟩
- Encodings support
  - JSON ⟨Cisco-XR | Junos | Huawei VRP⟩
  - GPB-KV ⟨Cisco-XR | Cisco-XE⟩
  - GPB ⟨Huawei VRP "OC interface"⟩

■ Main coordinates for YANG Push Notifications

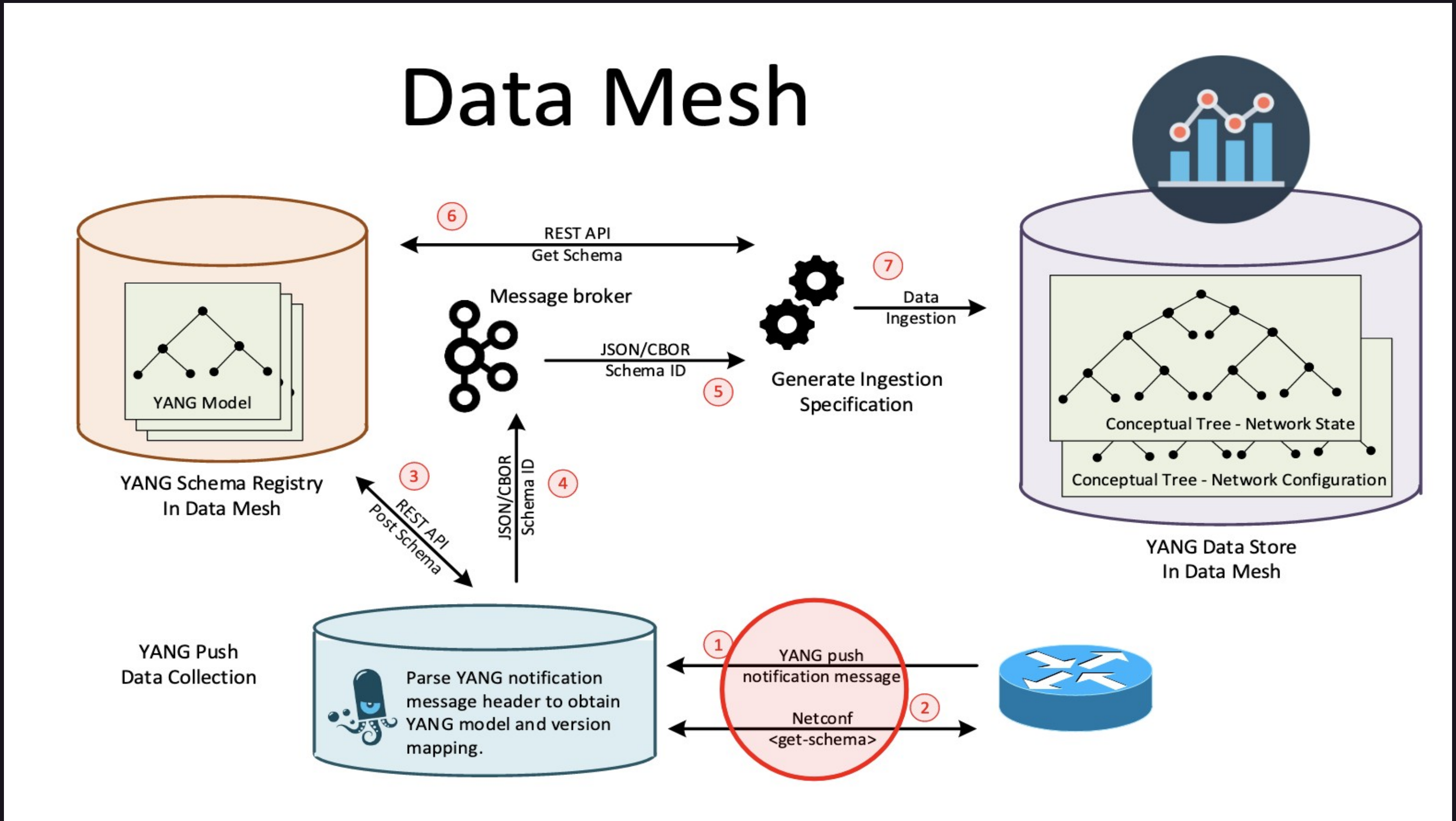
- Massive data scale out
- Model distribution
- Report model versions
- Accurate time-stamping
- Documented cross en/de-coding
- Pipeline protocol (vs router-collector protocol)
- Standardization

Evolving YANG Push ⟨missing puzzle pieces⟩

YANG Push	Today at Network Operators	Today at IETF
Transport protocol	Many and non-standard	netconf-http-notif and netconf-udp-notif
Encoding	JSON and Protobuf widely adopted. CBOR not yet implemented	XML in RFC7950, JSON in RFC7951, CBOR RFC9254
Subscription	Non-standard. Periodical widely adopted. On-change, sparse.	RFC8639 and RFC8641
Metadata	Non-standard. Partially among message content.	draft-tgraf-netconf-notif-sequencing draft-tgraf-yang-push-observation-time draft-claise-opsawg-collected-data-manifest draft-claise-netconf-metadata-for-collection
Versioning	Neither covered in subscription nor in publishing.	netmod-yang-module-versioning netconf-yang-notification-versioning
YANG Module	Non-standard widely adopted. IETF coverage non-existent.	Many RFCs defined

Credits to: T. Graf (Swisscom) @ IETF 116

■ YANG Push: Envisioned pipeline



Credits to: T. Graf (Swisscom) @ IETF 116

■ YANG Push: Tooling status quo

- Two hot metal vendors involved
- Transport libraries:
  - https://github.com/insa-unyte/udp-notif-c-collector
  - https://github.com/insa-unyte/https-notif-c-collector
- (Basic) YANG Push collector:
  - https://github.com/pmacct/pmacct (lots of WiP)
- YANG Schema Registry:
  - Confluent / Kafka (WiP)
- Three non-relational database vendors involved

■ YANG Push: behind the scenes

- Two hot metal vendors involved
- Four network operators
- One Streaming Pipeline vendor
- Three non-relational database vendors involved
- Check the latest here

■ YANG Push: Work to do

❬ developers! developers! developers! ❭

- Build ecosystem
  - Tools
  - Libraries
  - APIs

- Keep pushing
  - Standardization
  - Traction among operators
  - Adoption among vendors

Thanks

Backup

■ Comparison 〈YANG Based〉

	gRPC derived	NETCONF	RESTCONF
Efficiency & Performance	★★★★★	★★★☆☆	★★★★☆
Scalability	★★★★★	★★★☆☆	★★★☆☆
Security	★★★★☆	★★★★☆	★★★★☆
Streaming Capabilities	★★★★★	★★☆☆☆	★★☆☆☆
Interoperability	★★★★☆	★★★★★	★★★★★

■ Data (En|De)coding

- Information is Encoded according to a predefined schema

- Serialization process is converting the data into a stream of bytes, ready to be transmitted over the network

- Binary stream is delimited to ensure that the serialized data can be properly parsed on the receiving end

- Encoded data can be transmitted using any networking protocol that supports binary data

■ Protocol Buffers (GPB)

```
//▸ 「Compact」 & 「efficient」 binary representation of structured data
syntax = "proto3";

package huawei_dialout;

//▸ RPC service definition
service gRPCDataservice {
  rpc dataPublish(stream serviceArgs) returns(stream serviceArgs) {};
}

//▸ RPC message. Encoded/serialized using GBP
message serviceArgs {
  int64 ReqId = 1;
  oneof MessageData {
    //▸ Binary payload. Encoded/serialized using either GBP or GPB-KV
    bytes data = 2;
    //▸ Text payload, String representation of a JSON object. Encoded/serialized using UTF-8
    string data_json = 4;
  }
  string errors = 3;
}
```