Streaming Network Telemetry for fast and real-time data

Using Openconfig and gNMI

Sowmya Balasubramanian (Speaker)
Bruce Mah
2018 Technology Exchange, Orlando, FL
Oct 18, 2018
WHY?
Measurement Collection Today

- Current standard - SNMP based
- Data is sent every polling interval (~30s)
- Entire data table is sent
- Data is indexed using an integer (OID)
- Timestamping is done by the collector
- UDP based
Why Streaming Telemetry

- Subscription based
- Data is sent as soon as change occurs
- Data indexing is use-case friendly
- Data models
  - Standard (like Openconfig)
  - Vendor specific
Telemetry and Analytics Architecture

Network device → Collector agent → Data Collection → Other applications
Network device → Collector agent → Data Collection → Analytics
Network device → Collector agent → Data Collection → Visualization

Streaming data
Collector

- Simple lightweight process
- One collector for one network device (may extend to 2-3 network devices)
- Transforms the data into a normalized model
Openconfig Data Models

- Standard Yang based models
- Models for interfaces, telemetry, vlan, etc
- SNMP to openconfig mapping:

<table>
<thead>
<tr>
<th>SNMP</th>
<th>Openconfig</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifInDiscards</td>
<td>in-discards</td>
</tr>
<tr>
<td>ifInErrors</td>
<td>in-errors</td>
</tr>
<tr>
<td>ifInUcastPkts</td>
<td>in-unicast-pkts (from ifHCInUcastPkts)</td>
</tr>
<tr>
<td>ifOutDiscards</td>
<td>out-discards</td>
</tr>
<tr>
<td>ifOutErrors</td>
<td>out-errors</td>
</tr>
<tr>
<td>ifOutUcastPkts</td>
<td>out-unicast-pkts (from ifHCOOutUcastPkts)</td>
</tr>
<tr>
<td>ifHCInOctets</td>
<td>in-octets</td>
</tr>
<tr>
<td>ifHCOOutOctets</td>
<td>out-octets</td>
</tr>
<tr>
<td>ifName</td>
<td>name</td>
</tr>
<tr>
<td>ifAlias</td>
<td>description</td>
</tr>
</tbody>
</table>
gNMI (gRPC Network Management Interface)

- gRPC - A high performance, open-source universal RPC framework
- Uses protocol buffers for serialization
- Works across different languages and platforms
- Client calls procedures in server
  - Procedures are defined using service definitions
  - “Subscribe” call for streaming telemetry
- Different subscription modes
  - STREAM, POLL, ONCE
gNMI Client

```python
import grpc
from gnmi import gnmi_pb2
import util
import json

def listen(server, subscribePath, callback):
    # We get the gRPC channel differently depending on whether we need
    # TLS or not
    if server.tls:
        creds = grpc.ssl_channel_credentials(root_certificates=open(server.cert).read().encode("utf-8"))
        channel = grpc.secure_channel(target=server.access_point(), credentials=creds)
    else:
        channel = grpc.insecure_channel(server.access_point())
    grpc.channel_ready_future(channel).result(None)
    gnmi_stub = gnmi_pb2.gNMIStub(channel)
    gnmi_path = util:convertToGnmiPathElement(subscribePath)

    subscription_list = create_subscriptions(gnmiPath=gnmi_path)
    subscription_request = iter([gnmi_pb2.SubscribeRequest(subscribe=subscription_list)])
    responses = gnmi_stub.Subscribe(subscription_request, server.timeout, metadata=server.access_credentials())
    for response in responses:
        callback(response)

def create_subscriptions(gnmiPath):
    subscriptions = [gnmi_pb2.Subscription(path=gnmiPath, mode=0, suppress_redundant=1, sample_interval=10 * 1000000000, heartbeat_interval=10 * 1000000000)]

    return gnmi_pb2.SubscriptionList(prefix=None, mode=0, allow_aggregation=False, encoding=None, subscription=subscriptions, use_aliases=None, qos=None)
```
Experiments

• Explore / investigate
  – Protocol versions, functionality, and features
  – Availability of models
  – Use available open-source tooling

• Stream data
  – Python-based collector subscribes to telemetry, pushes updates to ESnet analytics back-end
  – Work with our next-gen data collection and analytics platform team
    • Generalize data models
    • Remove SNMP dependencies
  – Visualize data on experimental version of ESnet portal
Lessons Learned

- gNMI is still evolving, protocol version compatibility can be an issue
- Protocol functionality is vendor-specific
- Model support is vendor-specific
- Streaming telemetry differences with SNMP
  - Data model vs. MIBs
  - Subscription and push updates paradigm
  - Require different tooling
- Claimed improvements in accuracy and responsiveness needs more validation (future work?)
Thank you!
Questions?
sowmya@es.net