Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

R. Shakir
A. Shaikh
P. Borman
M. Hines
C. Lebsack
C. Morrow
Google

March 5, 2018

gRPC Network Management Interface (gNMI)
draft-openconfig-rtgwg-gnmi-spec-01

Abstract

This document describes the gRPC Network Management Interface (gNMI), a network management protocol based on the gRPC framework. gNMI supports retrieval and manipulation of state from network elements where the data is represented by a tree structure, and addressable by paths. The gNMI service defines operations for configuration management, operational state retrieval, and bulk data collection via streaming telemetry. The authoritative gNMI specification is maintained at [GNMI-SPEC].

Status of This Memo

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Table of Contents

1. Introduction

This document provides an overview of gNMI, a gRPC-based protocol for state management on network elements [GRPC].The gRPC Network Management Interface (gNMI) supports modification and retrieval of configuration, as well as control and transmission telemetry streams from a network element to a data collection system.This allows a single implementation on the network element, as well as a single NMS element to interact with the device via telemetry and configuration RPCs.

All messages within the gRPC service definition are defined as protocol buffers (specifically proto3) [PROTO].gRPC service definitions are described using the relevant features of the protobuf IDL.The authoritative protobuf definition is maintained in [GNMI-PROTO].The current, authoritative version of the gNMI specification is available at [GNMI-SPEC].

gNMI offers an alternative to management protocols such as NETCONF [RFC6241] and RESTCONF [RFC8040] with implementations on devices from multiple vendors.gNMI derives a number of benefits from being built on gRPC and HTTP/2, including modern security mechanisms, bidirectional streaming, binary framing, and a wide variety of language bindings to simplify integration with management applications.With protobuf encoding, it also provides significant efficiency advantages over XML serialization with a 3 to 10 times reduction in data volume (see the "Developer Guide" in [PROTO] for examples).A number of open source tools and reference implementations are available from the OpenConfig working group [GNMI-TOOLS].

2. Terminology

Throughout this document the following terminology is used:

- o Telemetry - refers to streaming data relating to underlying characteristics of the device - either operational state or configuration.

- o Configuration - elements within the data schema which are read/ write and can be manipulated by the client.

- o Target - the device within gNMI which acts as the owner of the data that is being manipulated or reported on. Typically this will be a network device.

- o Client - the device or system using gNMI to query/modify data on the target, or act as a collector for telemetry data. Typically this will be a network management application.

3. Protocol overview

The sections below provide an overview of the gNMI protocol operations, leaving a detailed discussion to the full specification in [GNMI-SPEC].

3.1. Data payloads and paths

gNMI is often used to carry payloads that contain data instances of YANG schemas (for example based on OpenConfig models [OPENCONFIG]), but can be used for any data with the following characteristics:

1. structure that can be represented by a tree, where nodes can be uniquely identified by a path consisting of node names, or node names coupled with attributes;

2. values can be serialised into a scalar object.

Values may be serialised to native protobuf scalar types, structured data types (e.g. as JSON or protobuf object), or a schema language-specific type (e.g., YANG Decimal64).

Data in gNMI is addressed by a path, which is represented as a structured list of elements, each with associated attributes if present. For example, the human-readable path "/interfaces/interface[name=eth0]/config/description" is represented as a text-encoded protobuf as:

```
  path: <
  elem: <
    name: "interfaces"
  >
  elem: <
    name: "interface"
    key: <
      key: "name"
      value: "eth0"
    >
  >
  elem: <
    name: "config"
  >
  elem: <
    name: "description"
  >
>
```

For more efficient handling of paths, gNMI supports a path prefix that is applied to every path in a message. Paths in gNMI are always absolute, constructed by concatenating the prefix with the path.

3.2. gNMI RPCs

gNMI is designed with a small number of base remote procedure calls (RPCs) to simplify client and target implementations. This section provides a high-level overview of each RPC. Full details are available in [GNMI-SPEC].

3.2.1. Capabilities

The Capabilities RPC allows a client to interrogate a gNMI target to learn about supported features. The primary information returned by a target include the set of models it supports, the data encodings supported, and the version of the gNMI protocol it is using. The model information is expected to be based on a published model catalog [I-D.openconfig-netmod-model-catalog].

3.2.2. Subscribe

Subscribe is a bidirectional streaming RPC that allows clients and targets to send independent sequences of telemetry messages. Clients may subscribe to notifications for telemetry data by specifying a path to the desired data and a mode (in addition to other parameters). gNMI supports several modes, but the two common use cases are for periodically sampled data, such as counters, and event-driven data in which a notification is sent only when the corresponding data value changes. In response to a subscription, the target sends a stream of telemetry notifications that contain a timestamp, path, and updated value. Multiple updates may be included in a single RPC message.

A streaming RPC for telemetry has the benefit of not requiring the target to collect, stage in memory, and serialize all of the requested data at once. The target is able to send data as soon as it is available, and can manage its resources to avoid becoming overloaded when sending a large volume of data.

3.2.3. Set

Set is a unary RPC (i.e., single request and single response) that is sent by a client to update the state of the target. Set includes several operation types whereby data may be updated, deleted, or replaced. A Set RPC may include multiple operations -- the target is expected to treat each RPC as a transaction such that if all included operations cannot be completed successfully, the target’s state is unchanged. Clients using the gNMI Set RPC pre-stage a set of update operations into a single Set RPC call, which must be either completely applied, or rolled back - eliminating the complex, long-lived candidate changes used in other protocols.

3.2.4. Get

Get is also a unary RPC that allows clients to request an immediate snapshot of the current state from the target, specified by a path. The target is expected to collect the data when the request is received, and serialize it for immediate transmission to the client. Where supported, gNMI allows the client to specify the type of data that should be returned (e.g., configuration state, operational state, etc.).

4. Additional operations

The sections below describe additional features and operations in gNMI.

4.1. Error handling

Rather than defining application-level error messates, gNMI leverages native error handling mechanisms in gRPC in which canonical error codes and context information are part of the Status message in every RPC. The gNMI specification provides guidance on how gNMI error conditions should be mapped to canonical error codes.

4.2. gNMI Extensions

While the base gNMI protocol is deliberately limited to a set of simple operations, some use cases require additional parameters that may be only applicable in those scenarios. gNMI extensions [GNMI-EXT] define a common way to add new payload to gNMI RPCs for these use cases without requiring changes in the core protocol specification.

5. Security Considerations

gNMI allows access and manipulation of state on network devices, hence it requires careful consideration of security implications including authentication and authorization of RPCs. The gNMI specification [GNMI-SPEC] and companion document [GNMI-SECURITY] discuss the considerations in more detail.

6. IANA Considerations

No IANA considerations at this time. In the future there may be a request for a protocol registry entry and well-known port allocation.

7. References

7.1. Normative references

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

7.2. Informative references

[GRPC] The gRPC authors, "gRPC", March 2018, <https://grpc.io/>.

[PROTO] Google, "Protocol buffers", March 2018, <https://developers.google.com/protocol-buffers/>.

[GNMI-PROTO] OpenConfig operator working group, "gnmi.proto", February 2018, <https://github.com/openconfig/gnmi/tree/master/proto/gnmi>.

[GNMI-SPEC] OpenConfig operator working group, "gRPC Network Management Interface (gNMI) v0.6.0", January 2018, <https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-specification.md>.

[GNMI-TOOLS] OpenConfig operator working group, "gNMI Github repository", March 2018, <https://github.com/openconfig/gnmi>.

[GNMI-EXT] OpenConfig operator working group, "Extensions to gNMI", January 2018, <https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-extensions.md>.

[GNMI-SECURITY] OpenConfig operator working group, "gNMI Authentication and Encryption", October 2016, <https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-authentication.md>.

[OPENCONFIG] OpenConfig operator working group, "OpenConfig", March 2018, <http://www.openconfig.net/>.

[I-D.openconfig-netmod-model-catalog] Shaikh, A., Shakir, R., and K. D’Souza, "Catalog and registry for YANG models", draft-openconfig-netmod-model-catalog-02 (work in progress), March 2017.

Appendix A. Change summary

A.1. Changes between revisions -00 and -01

- Replaced specification content with overview material and reference to normative reference to the gNMI specification document

- Updated to reflect changes in the gNMI specification and introduction of gNMI extensions.

Authors' Addresses

Rob Shakir
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043

Email: robjs@google.com

Anees Shaikh
Google
1600 Amphitheatre Pkwy
Mountain View, CA 94043
US

Email: aashaikh@google.com

Paul Borman
Google
1600 Amphitheatre Pkwy
Mountain View, CA 94043
US

Email: borman@google.com

Marcus Hines
Google
1600 Amphitheatre Pkwy
Mountain View, CA 94043
US

Email: hines@google.com

Carl Lebsack
Google
1600 Amphitheatre Pkwy
Mountain View, CA 94043
US

Email: csl@google.com

Chris Morrow
Google

Email: christopher.morrow@gmail.com