

NETCONF
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2018

G. Zheng
T. Zhou
A. Clemm
Huawei
July 2, 2017

UDP based Publication Channel for Streaming Telemetry
draft-zheng-netconf-udp-pub-channel-00

Abstract

This document describes a UDP-based publication channel for streaming telemetry use to collect data from devices. A new shim header is proposed to facilitate the distributed data collection mechanism which directly pushes data from line cards to the collector. Because of the lightweight UDP encapsulation, higher frequency and better transit performance can be achieved.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 3, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	4
3.	Solution Overview	4
4.	UDP Transport for Publication Channel	6
4.1.	Data Format	6
4.2.	Options	8
4.2.1.	Reliability Option	8
4.2.2.	Authentication Option	9
4.2.3.	Encryption Option	9
4.3.	Data encoding	9
5.	YANG Data Model for Subscription Management	9
6.	Retransmission Request	10
7.	IANA Considerations	10
8.	Security Considerations	10
9.	Acknowledgements	10
10.	References	10
10.1.	Normative References	10
10.2.	Informative References	11
Appendix A.	An Appendix	11
Authors' Addresses	11

[1.](#) Introduction

Streaming telemetry refers to sending a continuous stream of operational data from a device to a remote receiver. This provides an ability to monitor a network from remote and to provide network analytics. Devices generate telemetry data and push that data to a collector for further analysis. By streaming the data, much better performance, finer-grained sampling, monitoring accuracy, and bandwidth utilization can be achieved than with polling-based alternatives.

Sub-Notif [[I-D.ietf-netconf-subscribed-notifications](#)] and YANG-Push [[I-D.ietf-netconf-yang-push](#)] defines a mechanism that allows a collector to subscribe to updates of YANG-defined data that is maintained in a YANG [[RFC7950](#)] datastore. The mechanism separates

the management and control of subscriptions from the transport that is used to actually stream and deliver the data. Two transports have been defined so far, Netconf and Restconf/HTTP2.

While powerful in its features and general in its architecture, in its current form the mechanism needs to be extended to stream telemetry data at high velocity from devices that feature a distributed architecture. Specifically, there are two aspects that need to be addressed:

1. The transports that have been defined so far, Netconf and HTTP2, are ultimately based on TCP and lack the efficiency needed to stream data continuously at high velocity. A lightweight, more efficient transport, e.g. a transport based on UDP is needed.
 - * Firstly, data collector will suffer a lot of TCP connections from many line cards equipped on different devices.
 - * Secondly, as no connection state needs to be maintained, UDP encapsulation can be easily implemented by hardware which will further improve the performance.
 - * Thirdly, because of the lightweight UDP encapsulation, higher frequency and better transit performance can be achieved, which is important for streaming telemetry.
2. The current design involves a single push server. In the case of data originating from multiple line cards, the design requires data to be internally forwarded from those line cards to the push server, presumably on a main board, which then combines the individual data items into a single consolidated stream. This centralized data collection mechanism can result in a performance bottleneck, especially when large amounts of data are involved. What is needed instead is support for a distributed mechanism that allows to directly push multiple individual substreams, e.g. one from each line card, without needing to first pass them through an additional processing stage for internal consolidation, but still allowing those substreams to be managed and controlled via a single subscription.

This document specifies a distributed data collection mechanism which can directly push data from line cards to a collector by using a UDP based publication channel. Specifically, the following are specified:

- o A higher-performance transport option for YANG-Push that leverages UDP.

- o Extensions to YANG-Push's subscription model that allow a single subscription to control multiple internal data originators that each generate their own independent telemetry streams. Note: Because the ability to support multiple streams via a single subscription might be applicable to other transports as well, this aspect might be split into a separate specification in future revisions of this draft.

While this document will focus on the data publication channel, the subscription can be used in conjunction with the mechanism proposed in [[I-D.ietf-netconf-yang-push](#)] with necessary extensions.

2. Terminology

Streaming telemetry: refers to sending a continuous stream of operational data from a device to a remote receiver. This provides an ability to monitor a network from remote and to provide network analytics.

Component subscription: A subscription that defines the data from each individual entity which is managed and controlled by a single subscription server.

Subscription agent: An agent that streams telemetry data per the terms of a component subscription.

3. Solution Overview

The typical distributed data collection solution is shown in figure 1. The subscription server located in the main board receives the subscription requests or configurations. It may be colocated, not necessary, with a Netconf server which interacts with outside clients. When receiving a subscription request, the subscription server decomposes the subscription into multiple component subscriptions, each involving data from a separate internal telemetry source, for example a line card. The component subscriptions are distributed within the device to the subscription agents located in line cards. Subsequently, each line card generates its own stream of telemetry data, collecting and encapsulating the packets per the component subscription and streaming it to the designated data collector.

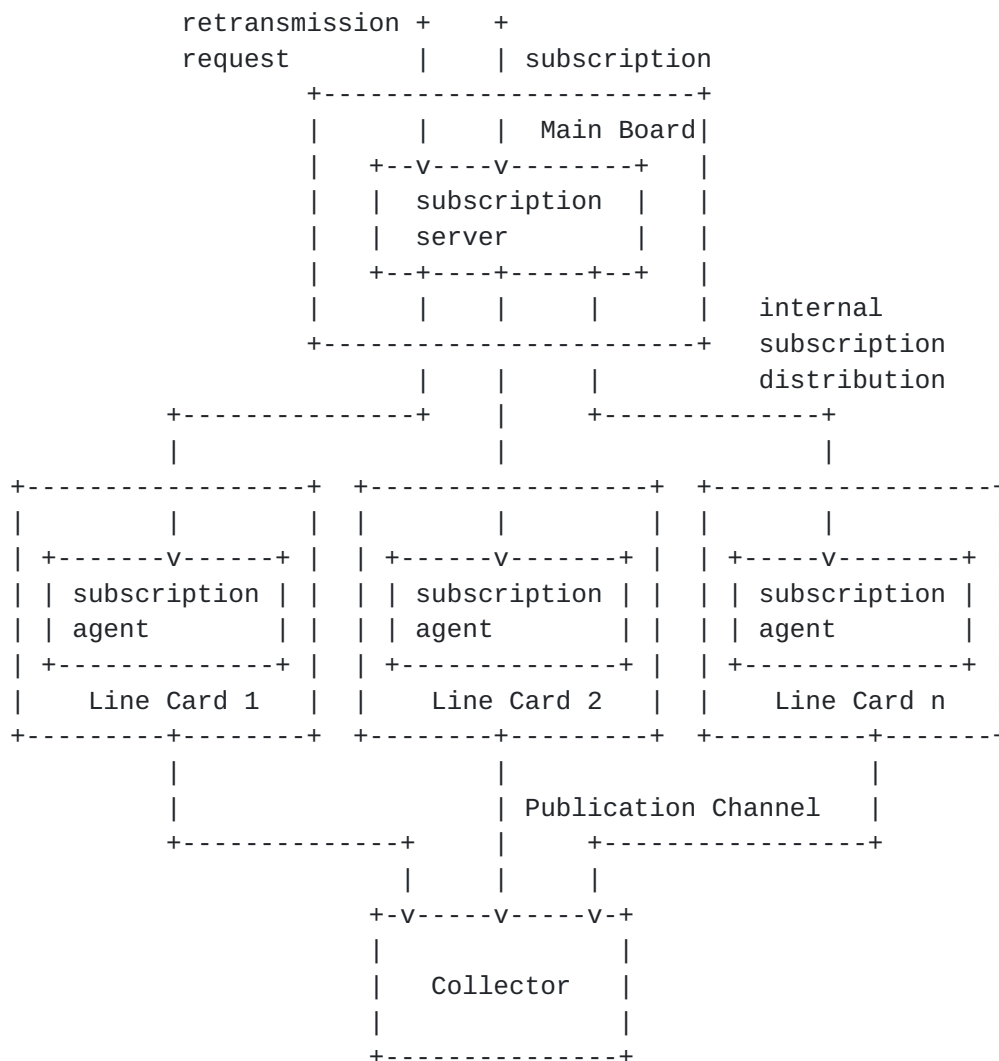
The publication channel supports the reliable data streaming, for example for some alarm events. The subscriber has the option of deducing the packet loss and the disorder based on the information carried by the notification data. And the subscriber will decide the behavior to request retransmission. The subscriber can send the

retransmission request to the subscriber server for further processing.

Subscription server and subscription agents interact with each other in several ways:

- o Subscription agents need to have a registration or announcement handshake with the subscription server, so the subscription server is aware of them and of lifecycle events (such as subscription agents appearing and disappearing).
- o The subscription server relays the component subscriptions to the subscription agents.
- o The subscription agents indicate status of component subscriptions to the subscription server. The status of the overall "master" subscription is maintained by the subscription server. The subscription server is also responsible for notifying the subscriber in case of any problems of component subscriptions.

The rest of the draft describes the UDP based publication channel.



4. UDP Transport for Publication Channel

In [\[I-D.void-netconf-notification-messages\]](#), the transport independent message header is proposed for the notification use. The following shim header refers to and implements that message header definition.

4.1. Data Format

The data format of the UDP based based publication transport is shown as follows.

- o The Timestamp, including the second part and the microsecond part, indicate the time the message was packaged and sent to the receiver. The Timestamp is defined per [RFC 3339](#).
- o The details of the Options will be described in the respective sections.

After the inform header is the real content which is encoded. The actual encoding is based on the subscription, e.g., in binary with GPB or CBOR [[RFC7049](#)].

More details of the content encoding is TBD.

[4.2.](#) Options

[4.2.1.](#) Reliability Option

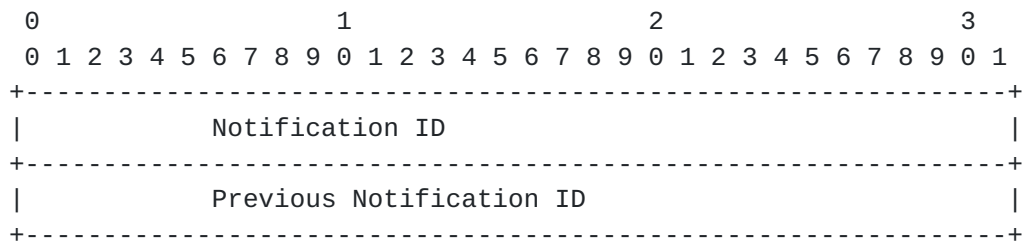
The UDP based publication transport described in this document provides two streaming modes, the reliable mode and the unreliable mode, for different SLA (Service Level Agreement) and telemetry requirements.

In the unreliable streaming mode, the line card pushes the encapsulated data to the data collector without any sequence information. So the subscriber does not know whether the data is correctly received or not. Hence no retransmission happens.

The reliable streaming mode provides sequence information in the UDP packet, based on which the subscriber can deduce the packet loss and disorder. Then the subscriber can decide whether to request the retransmission of the lost packets.

In most case, the unreliable streaming mode is preferred. Because the reliable streaming mode will cost more network bandwidth and precious device resource. Different from the unreliable streaming mode, the line card cannot remove the sent reliable notifications immediately, but to keep them in the memory for a while. Reliable notifications may be pushed multiple times, which will increase the traffic. When choosing the reliable streaming mode or the unreliable streaming mode, the operator needs to consider the reliable requirement together with the resource usage.

When the reliability flag is set to 1. The following option will be attached



The notification ID is generated continuously by the message generator. Different subscribers share the same notification ID sequence. Current ID and previous ID will be added in the packets. For example, there are two subscriber A and B,

- o Notification IDs for the generator are : 1,2,3,[4,5],6,7,8,9, in which Subscriber A subscribes [1,2,3,6,7], Subscriber B subscribes [1,2,4,5,7,8,9].
- o A will receive : [0,1][1,2][2,3][3,6][6,7]
- o B will receive : [0,1][1,2][2,4][4,5][5,7][7,8]

4.2.2. Authentication Option

TBD

4.2.3. Encryption Option

TBD

4.3. Data encoding

Subscribed data can be encoded in GPB, CBOR, XML or JSON format. It is conceivable that additional encodings may be supported as options in the future. This can be accomplished by augmenting the subscription data model with additional identity statements used to refer to requested encodings.

5. YANG Data Model for Subscription Management

To enable the UDP based publication transport, the subscription configuration need to provide necessary information. The subscription management YANG Model extends ietf-subscribed-notifications described [[I-D.ietf-netconf-yang-push](#)]


```
+--rw subscription-config {configured-subscriptions}?  
| + ...  
| +--rw receivers  
| | + ...  
| | +--rw protocol? transport-protocol  
| | | +--rw udp-transport-type? udp-transport-type  
| | |   +--rw reliable?  
| | |   +--rw authentication?  
| | |   +--rw encryption?
```

As in the above YANG tree, when the transport protocol is set to UDP, `retries` indicates the maximum retry times of the reliable streaming mode, and the `timeout` indicates the time out for retry in reliable streaming mode.

TBD. Note this YANG tree just to show we need to extend subscription mode, including the configurations and the RPCs. More details will be added later.

6. Retransmission Request

TBD

7. IANA Considerations

TBD

8. Security Considerations

9. Acknowledgements

The authors of this documents would like to thank Eric Voit, Tim Jenkins, and Huiyang Yang for the initial comments.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", [RFC 7049](#), DOI 10.17487/RFC7049, October 2013, <<http://www.rfc-editor.org/info/rfc7049>>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<http://www.rfc-editor.org/info/rfc7950>>.

10.2. Informative References

[I-D.ietf-netconf-subscribed-notifications]

Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Custom Subscription to Event Notifications", [draft-ietf-netconf-subscribed-notifications-02](#) (work in progress), April 2017.

[I-D.ietf-netconf-yang-push]

Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscribing to YANG datastore push updates", [draft-ietf-netconf-yang-push-07](#) (work in progress), June 2017.

[I-D.voit-netconf-notification-messages]

Voit, E., Bierman, A., Clemm, A., and T. Jenkins, "Notification Message Headers and Bundles", [draft-ietf-netconf-notification-messages-00](#) (work in progress), April 2017.

Appendix A. An Appendix

Authors' Addresses

Guangying Zheng
Huawei
Nanjing, Jiangsu
China

Fax: +
Email: zhengguangying@huawei.com

Tianran Zhou
Huawei
156 Beiqing Rd., Haidian District
Beijing
China

Email: zhoutianran@huawei.com

Alexander Clemm
Huawei
2330 Central Expressway
Santa Clara, California
USA

Email: alexander.clemm@huawei.com