NETCONF
Internet-Draft
Intended status: Standards Track
Expires: January 11, 2021

G. Zheng
T. Zhou
Huawei
T. Graf
Swisscom
P. Francois
INSA-Lyon
P. Lucente
NTT

July 10, 2020

UDP-based Transport for Configured Subscriptions
draft-unyte-netconf-udp-notif-00

Abstract

This document describes an UDP-based notification mechanism to collect data from networking devices. A shim header is proposed to facilitate the streaming of data directly from line cards to a collector. The objective is to rely on a lightweight approach to allow for higher frequency and better transit performance compared to already established notification mechanisms.

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

Copyright Notice

Table of Contents

1. Introduction

Sub-Notif [RFC8639] defines a mechanism that lets a collector subscribe to the publication of YANG-defined data maintained in a YANG [RFC7950] datastore. The mechanism separates the management and control of subscriptions from the transport used to deliver the data. Three transport mechanisms, namely NETCONF transport [RFC8640], RESTCONF transport [RFC8650], and HTTPS transport [I-D.ietf-netconf-https-notif] have been defined so far for such notification messages.

While powerful in its features and general in their architecture, the currently available transport mechanisms need to be complemented to support data publications at high velocity from devices that feature a distributed architecture. The currently available transports are based on TCP and lack the efficiency needed to continuously send notifications at high velocity.

This document specifies a transport option for Sub-Notif that leverages UDP. Specifically, it facilitates the distributed data collection mechanism described in [I-D.unyte-netconf-distributed-notif]. In the case of data originating from multiple line cards, centralized designs require data to be internally forwarded from those line cards to the push server, presumably on a route processor, which then combines the individual data items into a single consolidated stream. The centralized data collection mechanism can result in a performance bottleneck, especially when large amounts of data are involved.

What is needed is the support for a mechanism that allows for directly pushing multiple substreams, e.g. one from each line card, without passing them through an additional processing stage for internal consolidation. The proposed UDP-based transport allows for such a distributed data collection approach.

- Firstly, a UDP approach reduces the burden of maintaining a large amount of active TCP connections at the collector, notably in cases where it collects data from the line cards of a large amount of networking devices.

- Secondly, as no connection state needs to be maintained, UDP encapsulation can be easily implemented by the hardware of the publication streamer, which will further improve performance.

- Ultimately, such advantages allow for a larger data analysis feature set, as more voluminous, finer grained data sets can be streamed to the collector.

The transport described in this document can be used for transmitting notification messages over both IPv4 and IPv6.

This document describes the notification mechanism. It is intended to be used in conjunction with [RFC8639], extended by [I-D.unyte-netconf-distributed-notif].

Section 2 describes the control of the proposed transport mechanism. Section 3 details the notification mechanism and message format. Section 4 discusses congestion control. Section 5 covers the applicability of the proposed mechanism.

2. Configured Subscription to UDP-Notif

This section describes how the proposed mechanism can be controlled using subscription channels based on NETCONF or RESTCONF.

Following the usual approach of Sub-Notif, configured subscriptions contain the location information of all the receivers, including the IP address and the port number, so that the publisher can actively send UDP-Notif messages to the corresponding receivers.

Note that receivers MAY NOT be already up and running when the configuration of the subscription takes effect on the monitored device. The first message MUST be a separate subscription-started notification to indicate the Receiver that the stream has started flowing. Then, the notifications can be sent immediately without delay. All the subscription state notifications, as defined in [RFC8639], MUST be encapsulated in separate notification messages.

3. UDP-Based Transport

In this section, we specify the UDP-Notif Transport behaviour. Section 3.1 describes the general design of the solution. Section 3.2 specifies the UDP-Notif message format. Section 3.3 describes a generic optional sub TLV format. Section 3.3.1 uses such options to provide a fragmentation solution for large UDP-Notif message payloads. Section 3.4 describes the encoding of the message payload.

3.1. Design Overview

As specified in Sub-Notif, the telemetry data is encapsulated in the NETCONF/RESTCONF notification message, which is then encapsulated and carried using transport protocols such as TLS or HTTP2. Figure 1 illustrates the the structure of an UDP-Notif message.

- The Message Header contains information that facilitate the message transmission before deserializing the notification message.

- Notification Message is the encoded content that the publication stream transports. The common encoding methods include GPB [1], CBOR [RFC7049], JSON, and XML. [I-D.ietf-netconf-notification-messages] describes the structure of the Notification Message for single notifications and bundled notifications.

```
               +-------+  +--------------+  +--------------+
               |  UDP  |  |   Message    |  | Notification |
               |       |  |   Header     |  | Message      |
               +-------+  +--------------+  +--------------+
```

Figure 1: UDP-Notif Message Overview

3.2. Format of the UDP-Notif Message Header

The UDP-Notif Message Header contains information that facilitate the message transmission before deserializing the notification message. The data format is shown in Figure 2.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-------+-------+---------------+-------------------------------+
     | Vers. |  ET   |  Header Len   |      Message Length           |
     +-------+-------+---------------+-------------------------------+
     |                       Message-Generator-ID                    |
     +---------------------------------------------------------------+
     |                       Message ID                              |
     +---------------------------------------------------------------+
     ~                       Options                                 ~
     +---------------------------------------------------------------+
```

Figure 2: UDP-Notif Message Header Format

The Message Header contains the following field:

- Version represents the PDU (Protocol Data Unit) encoding version. The initial version value is 0.

- ET is a 4 bit identifier to indicate the encoding type used for the Notification Message. 16 types of encoding can be expressed:

  * 0: GPB;

  * 1: CBOR;

  * 2: JSON;

  * 3: XML;

  * others are reserved.

- Header Length is the length of the message header in octets, including both the fixed header and the options.

- Message Length is the total length of the message within one UDP datagram, measured in octets, including the message header.

- Message-Generator-ID is a 32-bit identifier of the process which created the notification message. This allows disambiguation of an information source, such as the identification of different line cards sending the notification messages. The source IP address of the UDP datagrams SHOULD NOT be interpreted as the identifier for the host that originated the UDP-Notif message. Indeed, the streamer sending the UDP-Notif message could be a relay for the actual source of data carried within UDP-Notif messages.

- The Message ID is generated continuously by the sender of UDP-Notif messages. Different subscribers share the same Message ID sequence.

- Options is a variable-length field in the TLV format. When the Header Length is larger than 12 octets, which is the length of the fixed header, Options TLVs follow directly after the fixed message header (i.e., Message ID). The details of the options are described in the following section.

3.3. Options

All the options are defined with the following format, illustrated in Figure 3.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +---------------+---------------+--------------------------------
     |     Type      |    Length     |    Variable-length data
     +---------------+---------------+--------------------------------
```

Figure 3: Generic Option Format

- Type: 1 octet describing the option type;

- Length: 1 octet of the TLV Length, including the Type and Length fields;

- Variable-length data: 0 or more octets of TLV Value.

3.3.1. Fragmentation Option

The UDP payload length is limited to 65535. Application level headers will make the actual payload shorter. Even though binary encodings such as GPB and CBOR may not require more space than what

is left, more voluminous encodings such as JSON and XML may suffer from this size limitation. Although IPv4 and IPv6 senders can fragment outgoing packets exceeding their Maximum Transmission Unit(MTU), fragmented IP packets may not be desired for operational and performance reasons.

Consequently, implementations of the mechanism SHOULD provide a configurable max-fragment-size option to control the maximum size of a payload.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                  +---------------+---------------+
                                  |     Type      |     Length    |
  +-------------------------------+---------------+-------------+-+
  |            Fragment Number                                  |L|
  +-------------------------------------------------------------+-+
```

Figure 4: Fragmentation Option Format

The Fragmentation Option is to be included when the message content is fragmented into multiple pieces. Different fragments of one message share the same Message ID. An illustration is provided in Figure 4. The fields of this TLV are:

- Type: indicates Fragmentation Option. The Type value is to be asigned.

- Length: is a fixed value of 6 octets.

- Fragment Number: indicates the sequence number of the current fragment.

- L: is a flag to indicate whether the current fragment is the last one. When 0 is set, the current fragment is not the last one, hence more fragments are expected. When 1 is set, the current fragment is the last one.

3.4. Data Encoding

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

Implementation MAY support multiple encoding methods per subscription. When bundled notifications are supported between the

publisher and the receiver, only subscribed notifications with the same encoding can be bundled in a given message.

4. Congestion Control

Congestion control mechanisms that respond to congestion by reducing traffic rates and establish a degree of fairness between flows that share the same path are vital to the stable operation of the Internet [RFC2914]. While efficient, UDP has no built-in congestion control mechanism. Because streaming telemetry can generate unlimited amounts of data, transferring this data over UDP may be considered problematic. It is not recommended to use the proposed mechanism over congestion-sensitive network paths. The only environments where UDP-Notif is expected to be used are managed networks. The deployments require that the network path has been explicitly provisioned to handle the traffic through traffic engineering mechanisms, such as rate limiting or capacity reservations. The UDP-Notif Message ID can be used to deduce congestion based on packet loss detection. Hence the collector can notify the device to use a lower streaming rate. The interaction to control the streaming rate on the device is out of the scope of this document.

5. Applicability

The target application for UDP-Notif is the collection of data-plane information. The lack of reliability of the data streaming mechanism is thus considered acceptable as the mechanism is to be used in controlled environments, mitigating the risk of information loss, while allowing for publication of very large amounts of data. Moreover, in this context, sporadic events when incomplete data collection is provided is not critical for the proper management of the network.

6. A YANG Data Model for Management of UDP-Notif

The YANG model defined in Section 9 has two leafs augmented into one place of Sub-Notif [RFC8639], plus one identity.

```
    module: ietf-udp-subscribed-notifications
     augment /sn:subscriptions/sn:subscription/sn:receivers/sn:receiver:
       +--rw address   inet:ip-address
       +--rw port      inet:port-number
       +--rw enable-fragment?  boolean
       +--rw max-fragment-size?  uint32
```

7. YANG Module

```
<CODE BEGINS> file "ietf-udp-notif@2020-04-27.yang"
module ietf-udp-notif {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-udp-notif";
  prefix un;
  import ietf-subscribed-notifications {
    prefix sn;
    reference
      "RFC 8639: Subscription to YANG Notifications";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }

  organization "IETF NETCONF (Network Configuration) Working Group";
  contact
    "WG Web:   <http:/tools.ietf.org/wg/netconf/>
     WG List:  <mailto:netconf@ietf.org>

     Authors:  Guangying Zheng
               <mailto:zhengguangying@huawei.com>
               Tianran Zhou
               <mailto:zhoutianran@huawei.com>
               Thomas Graf
               <mailto:thomas.graf@swisscom.com>
               Pierre Francois
               <mailto:pierre.francois@insa-lyon.fr>
               Paolo Lucente
               <mailto:paolo@ntt.net>";

  description
    "Defines UDP-Notif as a supported transport for subscribed
    event notifications.

    Copyright (c) 2018 IETF Trust and the persons identified as authors
    of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or without
    modification, is permitted pursuant to, and subject to the license
    terms contained in, the Simplified BSD License set forth in Section
    4.c of the IETF Trust's Legal Provisions Relating to IETF Documents
    (https://trustee.ietf.org/license-info).
```

```
    This version of this YANG module is part of RFC XXXX; see the RFC

    itself for full legal notices.";

  revision 2020-04-27 {
    description
      "Initial version";
    reference
      "RFC XXXX: UDP-based Notifications for Streaming Telemetry";
  }

  identity udp-notif {
    base sn:transport;
    description
      "UDP-Notif is used as transport for notification messages
and state change notifications.";
  }

  identity encode-cbor {
    base sn:encoding;
    description
      "Encode data using CBOR as described in RFC 7049.";
    reference
      "RFC 7049: Concise Binary Object Representation";
  }

  identity encode-gpb {
    base sn:encoding;
    description
      "Encode data using GPB.";
  }

  grouping target-receiver {
    description
      "Provides a reusable description of a UDP-Notif target receiver.";
    leaf address {
      type inet:ip-address;
      mandatory true;
      description
        "IP address of target UDP-Notif receiver, which can be an
        IPv4 address or an IPV6 address.";
    }
    leaf port {
      type inet:port-number;
      mandatory true;
      description
        "Port number of target UDP-Notif receiver, if not specified,
        the system should use default port number.";
```

```
    }

    leaf enable-fragment {
      type boolean;
      default false;
      description
        "The switch for the fragment feature. When disabled, the
         publisher will not allow fragment for a very large data";
    }

    leaf max-fragment-size {
      when "../enable-fragment = true";
      type uint32;
      description "UDP-Notif provides a configurable max-fragment-size
      to control the size of each message.";
    }
  }

  augment "/sn:subscriptions/sn:subscription/sn:receivers/sn:receiver" {
    description
      "This augmentation allows UDP-Notif specific parameters to be
       exposed for a subscription.";
    uses target-receiver;
  }
}
<CODE ENDS>
```

8. IANA Considerations

This RFC requests that IANA assigns one UDP port number in the "Registered Port Numbers" range with the service name "udp-notif". This port will be the default port for the UDP-based notification Streaming Telemetry (UDP-Notif) for NETCONF and RESTCONF. Below is the registration template following the rules of [RFC6335].

Service Name: udp-notif

Transport Protocol(s): UDP

Assignee: IESG <iesg@ietf.org>

Contact: IETF Chair <chair@ietf.org>

Description: UDP-based Publication Streaming Telemetry

Reference: RFC XXXX

Port Number: PORT-X

IANA is requested to assign a new URI from the IETF XML Registry [RFC3688]. The following URI is suggested:

URI: urn:ietf:params:xml:ns:yang:ietf-udp-notif
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

This document also requests a new YANG module name in the YANG Module Names registry [RFC7950] with the following suggestion:

name: ietf-udp-notif
namespace: urn:ietf:params:xml:ns:yang:ietf-udp-notif
prefix: un
reference: RFC XXXX

9. Security Considerations

TBD

10. Acknowledgements

The authors of this documents would like to thank Alexander Clemm, Eric Voit, Huiyang Yang, Kent Watsen, Mahesh Jethanandani, Stephane Frenot, Timothy Carey, Tim Jenkins, and Yunan Gu for their constructive suggestions for improving this document.

11. References

11.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, DOI 10.17487/RFC2914, September 2000, <https://www.rfc-editor.org/info/rfc2914>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, DOI 10.17487/RFC4347, April 2006, <https://www.rfc-editor.org/info/rfc4347>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[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>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://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, <https://www.rfc-editor.org/info/rfc7950>.

[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>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", RFC 8640, DOI 10.17487/RFC8640, September 2019, <https://www.rfc-editor.org/info/rfc8640>.

[RFC8650] Voit, E., Rahman, R., Nilsen-Nygaard, E., Clemm, A., and A. Bierman, "Dynamic Subscription to YANG Events and Datastores over RESTCONF", RFC 8650, DOI 10.17487/RFC8650, November 2019, <https://www.rfc-editor.org/info/rfc8650>.

11.2. Informative References

[I-D.ietf-netconf-https-notif]
Jethanandani, M. and K. Watsen, "An HTTPS-based Transport for Configured Subscriptions", draft-ietf-netconf-https-notif-02 (work in progress), March 2020.

[I-D.ietf-netconf-notification-messages]
Voit, E., Jenkins, T., Birkholz, H., Bierman, A., and A. Clemm, "Notification Message Headers and Bundles", draft-ietf-netconf-notification-messages-08 (work in progress), November 2019.

[I-D.unyte-netconf-distributed-notif]
Zhou, T., Zheng, G., Voit, E., Graf, T., and P. Francois, "Subscription to Distributed Notifications", draft-unyte-netconf-distributed-notif-00 (work in progress), June 2020.

11.3. URIs

[1] https://developers.google.com/protocol-buffers/

Authors' Addresses

Guangying Zheng
Huawei
101 Yu-Hua-Tai Software Road
Nanjing, Jiangsu
China

Email: zhengguangying@huawei.com

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

Email: zhoutianran@huawei.com

Thomas Graf
Swisscom
Binzring 17
Zuerich 8045
Switzerland

Email: thomas.graf@swisscom.com

Pierre Francois
INSA-Lyon
Lyon
France

Email: pierre.francois@insa-lyon.fr

Paolo Lucente
NTT
Siriusdreef 70-72
Hoofddorp, WT 2132
NL

Email: paolo@ntt.net