Protocol Assisted Protocol (PAP)
draft-li-rtgwg-protocol-assisted-protocol-01

Abstract

For routing protocol troubleshooting, different approaches exhibit merits w.r.t. different situations. They can be generally divided into two categories, the distributive way and the centralized way. A very commonly used distributive approach is to log in possibly all related devices one by one to check massive data via CLI. Such approach provides very detailed device information, however it requires operators with high NOC (Network Operation Center) experience and suffers from low troubleshooting efficiency and high cost. The centralized approach is realized by collecting data from devices via approaches, like the streaming Telemetry or BMP (BGP Monitoring Protocol) RFC7854 [RFC7854], for the centralized server to analyze all gathered data. Such approach allows a comprehensive view to the whole network and facilitates automated troubleshooting, but is limited by the data collection boundary set by different management domains, as well as high network bandwidth and CPU computation costs.

This document proposes a semi-distributive and semi-centralized approach for fast routing protocol troubleshooting, localizing the target device and possibly the root cause, more precisely. It defines a new protocol, called the PAP (Protocol assisted Protocol), for devices to exchange protocol related information between each other in both active and on-demand manners. It allow devices to request specific information from other devices and receive replies to the requested data. It also allows actively transmission of information without request to inform other devices to better react w.r.t. network issues.

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 https://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 9, 2020.

Copyright Notice

Copyright (c) 2019 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 (https://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 ............................................. 3
  1.1. Motivation ........................................... 3
  1.2. PAP Usage Use cases ................................. 5
       1.2.1. Use Case 1: BGP Route Oscillation .......... 5
       1.2.2. Use Case 2: RSVP-TE Set Up Failure ......... 5
  2. Terminology ............................................ 6
  3. PAP Overview ........................................... 6
  4. PAP Message Format ...................................... 8
     4.1. Common Header ..................................... 8
     4.2. Negotiation Message ................................ 9
     4.3. Request Message ................................... 10
     4.4. Reply Message ..................................... 10
     4.5. Notification Message ............................... 11
     4.6. ACK Message ....................................... 12
1. Introduction

A healthy control plane, providing network connectivity, is the foundation of a well-functioning network. There have been rich routing and signaling protocols designed and used for IP networks, such as IGP (ISIS, OSPF), BGP, LDP, RSVP-TE and so on. The health issues of these protocols, such as neighbor/peer disconnect/set up failure, LSP set up failure, route flapping and so on, have been devoted with ongoing efforts for diagnosing and remediation.

1.1. Motivation

The distributive protocol troubleshooting approach is typically realized through manual per-device check. It’s both time- and labor-consuming, and requires NOC experience of the operators. Amongst all, localizing the target device is usually the most difficult and time-consuming part. For example, in the case of route loop, operators first log in a random device that reports TTL alarms, and then check the looped route in the Forwarding Information Base (FIB) and/or the Routing Information Base (RIB). It requires device by device check, as well as manual data correlation, to pin point to the exact responsible device, since the information retrieval and analysis of such distributive way is fragmented. In addition, the low efficiency and manual troubleshooting activities may further impact new network services and/or enlarge affected areas.

The centralized network OAM, by collecting network-wide data from devices, enables automatic routing protocol troubleshooting. Date collection protocols, such as SNMP (Simple Network Management Protocol) [RFC1157], NETCONF (Network Configuration Protocol) [RFC6241], and BMP [RFC7854], can provide various information retrieval, such as network states, routing data, configurations and so on. Such centralized way relies on the existence of a centralized server/controller, which is not supported by some legacy networks. What’s more, even with the existence of a centralized server/controller, it can only collect the data within its own management domain, while the cross-domain data are not available due to independent management of different ISPs. Thus, the lack of such...
information may lead to troubleshooting failure. In addition, centralized approaches may suffer from high network bandwidth and CPU computation consumptions.

Another way of protocol troubleshooting is utilizing the protocol itself to convey diagnosing information. For example, some reason codes are carried in the Path-Err/ResvErr messages of RSVP-TE, so that to other nodes may know the why the tunnel fails to be set up. Such approaches is semi-distributive and semi-centralized. It does not rely on the deployment of a centralized server, but still gets partial global view of the network. However, there still requires non-trivial augmentation works to existing routing protocols in order to support troubleshooting. This then raises the question that whether such non-routing data is suitable to be carried in these routing protocols. The extra encapsulation, parsing and analyzing work for the non-routing data would further slow down the network convergence. Thus, it’s better to separate the routing and non-routing data transmission as well as data parsing. In addition, coexisting with legacy devices may cause interop issues. Thus, relying on augmenting existing routing protocols without network-wide upgrading may not only fail to provide the troubleshooting benefit, but further affect the operation of the existing routing system. What’s more, the failure of routing protocol instance would lead to the failure of diagnosing itself. All in all, it’s reasonable to separate the protocol diagnosing data generation/encapsulation/transmission/parsing from the protocol itself.

This document proposes a new protocol, called the PAP (Protocol assisted Protocol), for devices to exchange protocol related information between each other. It allows both active and on-demand data exchange. Considering that massiveness of protocol/routing related data, the intuitive of designing PAP is not to exchange the comprehensive protocol/routing status between devices, but to provide very specific information required for the contemporary troubleshooting. The benefits of such a semi-distributive and semi-centralized approach are summarized as follows:

1. It facilitates automatic troubleshooting without requiring manual device by device check.
2. It allows individual device to have a more global view by requesting data from other devices.
3. It does not rely on the deployment of a centralized server/controller.
4. It passes the data collection boundary set by different management domains by cross-domain data exchange between devices.

5. It relieves the bandwidth pressure of network-wide data collection, and the processing pressure of the centralized server.

6. It does not affect the running of existing routing protocols.

1.2. PAP Usage Use cases

PAP allows both data request/reply and data notification between devices. PAP speakers use the exchanged PAP data to help fast localize the network issues.

1.2.1. Use Case 1: BGP Route Oscillation

A BGP route oscillation can be caused by various reasons, and usually leaves network-wide impact. In order to find the root cause and take remediation actions, the first step is to localize the oscillation source. In this case, a BGP speaker can send a PAP Request Message to the next hop device of the oscillating route asking "Are you the oscillation source?". If the BGP speaker is the oscillation source, possibly knows by running a device diagnosing system, replies with a PAP Reply Message saying that "I’m the oscillation source!" to the device who sends the PAP Request Message. If the BGP speaker is not the oscillation source, it further asks the same question with a PAP Request Message to its next hop device of the oscillating route. This request and reply process continues until the request has reached the oscillation source. The source device then sends a PAP Reply Message to tell its upstream device along the PAP request path that "I am the oscillation source!", and then "xx is the oscillation source!" information is further sent back hop by hop to the device who originates the request.

1.2.2. Use Case 2: RSVP-TE Set Up Failure

The MPLS label switch path set up, either using RSVP-TE or LDP, may fail due to various reasons. Typical troubleshooting procedures are to log in the device, and then check if the failure lies on the configuration, or path computation error, or link failure. Sometimes, it requires the check of multiple devices along the tunnel. Certain reason codes can be carried in the Path-Err/ResvErr messages of RSVP-TE, while other data are currently not supported to be transmitted to the path ingress/egress node, such as the authentication failure. Using PAP, the device, which is responsible for the tunnel set up failure, can send the PAP Notification Message to the Ingress device, and possibly with some reason codes so that
the ingress device can not only localize the target device but also the root cause.

2. Terminology

IGP: Interior Gateway Protocol
IS-IS: Intermediate System to Intermediate System
OSPF: Open Shortest Path First
BGP: Boarder Gateway Protocol
BGP-LS: Boarder Gateway Protocol-Link State
MPLS: Multi-Protocol Label Switching
RSVP-TE: Resource Reservation Protocol-Traffic Engineering
LDP: Label Distribution Protocol
BMP: BGP Monitoring Protocol
LSP: Link State Packet
IPFIX: Internet Protocol Flow Information Export
PAP: Protocol assisted Protocol
UDP: User Datagram Protocol

3. PAP Overview

PAP is a non-routing protocol used for PAP speakers to exchange routing protocol related information. The primary function of PAP is to provide a unified tunnel for protocol diagnosing information exchange without augmenting each specific protocol.

PAP uses UDP as its transport protocol, which is connectionless. The reason that UDP is selected over TCP is because PAP is intended for on-demand communications. The PAP packet is defined as follows. This document requires the assignment of a User Port registry for the UDP Destination Port.
This document uses PAP speakers to refer to two routing devices that support PAP and/or communicate with each other using PAP throughout. The communications between two PAP speakers should follow three major processes, i.e., the Capability Negotiation Process, the Request and Reply Process, and the Notification Process. This document defines 5 PAP Message types, i.e., Negotiation Message, Request Message, Reply Message, Notification Message, and ACK Message, which are used in the above PAP processes.

Although PAP is connectionless, there still requires a successful Capability Negotiation between two PAP speakers before the exchange of any PAP messages (except the Negotiation Message). The purpose of the Capability Negotiation process is to inform both PAP speakers of each other's PAP capabilities. The capabilities of a PAP speaker refer to the support of diagnosing information exchange of specific protocols, while the generation of such data are possibly enabled by a local protocol diagnosing system. The negotiation can be initiated by either the local or remote PAP speaker through sending out a PAP Negotiation Message. The Negotiation Message may or may not require an ACK Message, as indicated in the Negotiation Message. A successful negotiation is achieved if both PAP speakers have correctly received the other speaker's Negotiation Message regardless of the message content. A more detailed definition of a successful negotiation is defined in Section 5.1. After a successful negotiation, two PAP speakers can exchange any PAP Message on-demand.

The purpose of the Request and Reply process is to acquire information needed by a PAP speaker from other PAP speakers for diagnosing a specific protocol. The Request and Reply Messages can be customized for different protocols. The process starts by any PAP speaker sending a Request Message to another PAP speaker. The PAP receiver, after receiving the Request message, sends out a Reply Message to the request sender. The generation of both Request and Reply Messages is protocol-specific, and depends on possibly a local protocol diagnosing system. One Request Message received may further results in a new Request Message generated and sent out to a third PAP speaker, if the receiver does not have the answer to the sender. Both Request and Reply Messages may or may not require an ACK Message, as indicated in the Request/Reply Messages.

The Notification Process is used to serve active data notification. Any PAP speaker, having information to share with other PAP speakers,
may start to send Notification Messages to any target PAP speaker. This information sharing is also protocol-specific, and can be possibly generated by a local protocol diagnosing system. The Notification Message may or may not require an ACK Message, as indicated in the Notification Message.

4. PAP Message Format

4.1. Common Header

The common header is encapsulated in all PAP messages. It includes the following fields.

```
   0                  15                 31
+---------------------+-----------------------------+
|       Sequence Number       |        Length        |
+---------------------+-----------------------------+
|     Msg. Type     |                                 |
+---------------------+                                 
```

Figure 2. PAP Common Header

- Sequence Number (2 byte): It is used to indicate the sequence of the PAP Message. It uses unsigned integers to distinguish the PAP Messages sent to the same remote device. The integer is set incrementally in order time.

- Length (2 bytes): Length of the message in bytes, including the Common Header and the following Message.

- Message Type (1 byte): This indicates the PAP message type. The following types are defined, and listed as follows.
  * Type = TBD1: Negotiation Message. It is used for two devices to inform each other of the capabilities they support and no longer support.
  * Type = TBD2: Request Message.
  * Type = TBD3: Reply Message.
  * Type = TBD4: Notification Message.
  * Type = TBD5: ACK Message. It is used to confirm to the local device that the remote device has received a previous sent PAP message, which can be either a Negotiation Message, a Request Message, a Reply Message or a Notification Message.
4.2. Negotiation Message

The Negotiation Message is used for both capability negotiation between the local PAP speaker and the remote PAP speaker. It is also used to indicate the enabling and disabling of any supported capability. The Negotiation Message is required to be exchanged before two PAP speakers can exchange any other PAP Message types.

```
+--------------------------------------------+
<p>|       Version        |A|C|   Reserved      |
|--------------------------------------------|</p>
<table>
<thead>
<tr>
<th>Protocol Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Capability</td>
</tr>
<tr>
<td>--------------------------------------------</td>
</tr>
</tbody>
</table>
```

Figure 3. PAP Negotiation Message

- **Version (2 bytes):** It indicates the PAP version. The current version is 0.

- **Flags (2 bytes):** Two flag bits are currently defined.

  * The A bit is used to indicate if an ACK Message from the remote PAP speaker is required for each Negotiation Message sent. If an ACK is required, then the A bit SHOULD be set to "1", and "0" otherwise.

  * The C bit is used to indicate the enabling/disabling of the capabilities that carried in the Protocol Capability field. If the local device wants to inform the remote device of enabling one or more capabilities, the C bit SHOULD be set to "1". If the local device wants to inform the remote device of disabling one or more capabilities, the C bit SHOULD be set to "0".

- **Protocol Capability (4 bytes):** It is 4-byte bit map that indicates the capability of information exchange regarding various protocols. Each bit represents one protocol. For example, the right most bit of the Protocol Capability represents the capability of exchanging IS-IS protocol related information. The format and definition of information exchanged, regarding each protocol, MUST follow the PAP Message format and definition.

- **Optional Capability (4 bytes):** It is 4-byte bit map that indicates the capability of other information. It can be used to carry other application-specific information, which allows future extension. Its format and definition remains to be determined.
4.3. Request Message

The Request Message is used for the local device to request specific data regarding one specific protocol or application from the remote device. It MUST be sent after a successful Capability Negotiation Process (described in Section 5.1), and the requested protocol/application MUST be supported by both the local and remote devices, as indicated in the Negotiation Messages exchanged between the local and remote devices.

The Statistic TLV is defined as follows.

```
+----------------------+
|A|     Reserved       |
+----------------------+
|                  Capability                |
+--------------------------------------------+
|                 Data Request               |
+--------------------------------------------+
```

Figure 4. PAP Request Message

- Flags (2 bytes): The A bit is used to indicate if an ACK Message from the remote PAP speaker is required for each Request Message sent. If an ACK is required, then the A bit SHOULD be set to "1", and "0" otherwise.

- Capability (4 bytes): It is 4-byte bit map that indicates the specific protocol/application the Request Message is requesting data for. The respective bit of this specific protocol/application MUST be set to "1", and the rest of the bits MUST be set to "0".

- Data Request (Variable): Specifies what data the local device is requesting. The specific format remains to be determined.

4.4. Reply Message

The Reply Message is used to carry the information that the local device requests from the remote device through the Request Message.
4.5. Notification Message

The Notification Message is used to carry the information that the local device sends to the remote device.

- Flags (2 bytes): The A bit is used to indicate if an ACK Message from the remote PAP speaker is required for each Request Message sent. If an ACK is required, then the A bit SHOULD be set to "1", and "0" otherwise.

- Capability (4 bytes): It is 4-byte bit map that indicates the specific protocol/application the Notification Message is replying for. The respective bit of this specific protocol/application MUST be set to "1", and the rest of the bits MUST be set to "0".

- Data Nitification (Variable): It contains the data that the local device requests from the remote device. The specific format remains to be determined.
notifying for. The respective bit of this specific protocol/application MUST be set to "1", and the rest of the bits MUST be set to "0".

- Data Notification (Variable): It contains the data that the local device actively sends to the remote device. The specific format remains to be determined.

4.6. ACK Message

The ACK Message is used to confirm that the remote device has received a PAP Message that set the A bit to "1". The ACK Message includes only the ACK sequence number field, which MUST be set to the sequence number carried in the PAP Common Header of the received PAP message, which requires this ACK Message.

```
+--------------------------------------------+
|            ACK Sequence Number             |
+--------------------------------------------+
```

Figure 7. PAP ACK Message

5. PAP Operations

The PAP operations include the following 3 processes, the Capability Negotiation Process, the Data Request and Reply Process, and the Data Notification Process.

5.1. Capability Negotiation Process

The Capability Negotiation process refers to the process that two devices inform each other of the capabilities they support and no longer support. A successful negotiation that inform each other of the supported capabilities between two devices MUST be achieved before the exchange of any other PAP Message.

The first Negotiation Message can be sent at any time per the local device’s configuration. One or more Negotiation Messages can be sent at any time after the first Negotiation Message. Once a Negotiation Message is sent from a local device, an ACK Message from the remote device is required/or not per the indication of the "A" bit in the Negotiation Message. If the A bit is set to "1" (i.e., ACK is required), the local device SHOULD wait for the ACK Message from the remote device for a certain time period before taking further actions, and if no ACK Message is received within this time frame, the local device SHOULD resend the Negotiation Message to the remote device. The waiting period can be configured locally. This send and
wait process CAN be repeated for at most 3 times before receiving a
ACK Message from the remote device. If after 3 times of resending
the Negotiation Message, still no ACK received, then this negotiation
is treated as unsuccessful. If the A bit is set to "0", then the
local device does not wait for any ACK Message. If no Negotiation
Message is received from the remote device within a time frame, the
local device can resend the Negotiation Message. This send and wait
process CAN be repeated for at most 3 times before receiving a
Negotiation Message from the remote device. If after 3 times of
resending the Negotiation Message, still no Negotiation Message
received, then this negotiation is treated as Unsuccessful. The
waiting period can be configured locally. Once a Negotiation Message
is received from the remote device, the negotiation between the two
devices is treated as successful regardless from the local device’s
perspective.

On the other hand, the remote device, after receiving the Negotiation
Message from the local device, SHOULD send out its own Negotiation
Message that indicates the capabilities that it supports. If the A
bit of the received Negotiation Message from the local device is set
to "1", the remote device SHOULD sent an ACK Message before sending
the new Negotiation Message. After the remote device sends out the
Negotiation Message back to the local device, it waits/or not for the
ACK message, depending on if the A bit of the Negotiation Message
sent to the local device is set to "1". If it’s set to "1", the
remote device waits for the ACK message for a certain time period
before taking further actions. If no ACK Message is received within
this time frame, the remote device SHOULD resend the Negotiation
Message to the local device. The waiting period can be configured
locally at the remote device. This send and wait process CAN be
repeated for at most 3 times before receiving a ACK Message from the
local device. If after 3 times of resending the Negotiation Message,
still no ACK received, then the remote device treats this negotiation
as unsuccessful, and successful otherwise. If the A bit of the
Negotiation Message sent from the remote device is set to "0", then
the remote device treats the negotiation as successful after sending
out the Negotiation Message.

The C bit in the Negotiation Message MUST always be set to "1" before
the negotiation is successful. A successful Capability Negotiation
means that both the local and remote devices have the information of
what capabilities the other side support so that when exchanging any
other messages, only the capabilities that are supported by both ends
SHOULD be carried in the respective capability fields.

Another process of the Capability Negotiation Process is to inform
the remote device of the capability/capabilities that the local
device no longer supports with the indication of C bit set as "0".
To inform the other device of the disabled capability/capabilities, the local device MUST have sent one or more Negotiation Messages that indicate the support of such capability/capabilities previously.

5.2. Data Request and Reply Process

After a successful Capability Negotiation, the local device CAN send the Data Request Message to the remote device per local configuration. An Reply Message SHOULD only be sent after receiving a Request Message. If the A bit of the Request Message is set to "1" (i.e., ACK is required), the local device SHOULD wait for the ACK Message from the remote device for a certain time period before taking further actions, and if no ACK Message is received within this time frame, the local device SHOULD resend the Request Message to the remote device. The waiting period can be configured locally. This send and wait process CAN be repeated for at most 3 times before receiving a ACK Message from the remote device. If the A bit is set to "0", then the local device does not wait for any ACK Message. If no Reply Message is received from the remote device within a time frame, the local device can resend the Request Message. This send and wait process CAN be repeated for at most 3 times before receiving a Reply Message from the remote device. If after 3 times of resending the Request Message, still no Reply Message received, then no further action is taken. The waiting period can be configured locally.

On the other hand, the remote device, after receiving the Request Message from the local device, SHOULD send out the Reply Message to reply the request. If the A bit of the received Request Message from the local device is set to "1", the remote device SHOULD sent an ACK Message before sending the Reply Message.

To request data for multiple protocols/applications, separate Request Messages SHOULD be sent, with each Request Message requesting one specific protocol/application data. Accordingly, the Reply Message is also sent separately for each Request Message.

5.3. Data Notification Process

The Notification Message CAN be sent by the local or the remote device at any time once the Capability Negotiation is successful. Each Notification Message SHOULD only indicate one specific protocol/application. The A bit can be set to "1" or "0" to allow the local/remote device to know if the other device has received the Notification Message through the ACK Message.
6. IANA

TBD

7. Contributors

Jiaqing Zhang, Huawei, zhangjiaqing@huawei.com

8. Acknowledgments

TBD

9. References

[I-D.brockners-inband-oam-requirements]
Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mozes, D., Mizrahi, T., Lapukhov, P., and r. remy@barefootnetworks.com, "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements-03 (work in progress), March 2017.

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

[I-D.song-ntf]


Authors’ Addresses

Zhenbin Li
Huawei
156 Beiqing Rd
Beijing
China

Email: lizhenbin@huawei.com
Lei Li
Huawei
156 Beiqing Rd
Beijing
China

Email: lily.lilei@huawei.com

Yunan Gu
Huawei
156 Beiqing Rd
Beijing
China

Email: guyunan@huawei.com