

IPPM  
Internet-Draft  
Intended status: Standards Track  
Expires: October 24, 2019

H. Song, Ed.  
T. Zhou  
Z. Li  
Huawei  
J. Shin  
SK Telecom  
April 22, 2019

**Postcard-based On-Path Flow Data Telemetry**  
**draft-song-ippm-postcard-based-telemetry-03**

Abstract

The Postcard-Based Telemetry (PBT) allows network OAM applications to collect telemetry data about any user packet. Unlike the E2E and trace modes in in-situ OAM (IOAM), PBT does not require user packets to carry any telemetry data, but directly exports the telemetry data from network nodes to a collector through separated OAM packets called postcards. Two variations of PBT, PBT-I and PBT-M, are described. PBT-I requires inserting an instruction header to user packets to guide the data collection. PBT-I is designed as another mode of IOAM, Per-Hop Postcard (PHP), to complement the existing operational modes of IOAM. PBT-M only marks the user packets or configure the flow filter to invoke the data collection. PBT-M also provides a complement to IOAM and address several implementation and deployment challenges of it.

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 October 24, 2019.

## 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

|                        |   |                    |
|------------------------|---|--------------------|
| <a href="#">1.</a>     | Motivation . . . . .  | <a href="#">2</a>  |
| <a href="#">2.</a>     | PBT-M: Postcard-based Telemetry with Packet Marking . . . . .     | <a href="#">5</a>  |
| <a href="#">2.1.</a>   | New Requirements . . . . .  | <a href="#">5</a>  |
| <a href="#">2.2.</a>   | Solution Description . . . . .                                    | <a href="#">6</a>  |
| <a href="#">2.3.</a>   | New Challenges . . . . .  | <a href="#">7</a>  |
| <a href="#">2.4.</a>   | Considerations on PBT-M Design . . . . .                          | <a href="#">8</a>  |
| <a href="#">2.4.1.</a> | Packet Marking . . . . .  | <a href="#">8</a>  |
| <a href="#">2.4.2.</a> | Flow Path Discovery . . . . .                                     | <a href="#">8</a>  |
| <a href="#">2.4.3.</a> | Packet Identity for Export Data Correlation . . . . .             | <a href="#">9</a>  |
| <a href="#">2.5.</a>   | Avoid Packet Marking through Node Configuration . . . . .         | <a href="#">9</a>  |
| <a href="#">3.</a>     | PBT-I: Postcard-based Telemetry with Instruction Header . . . . . | <a href="#">10</a> |
| <a href="#">3.1.</a>   | Solution Description . . . . .                                    | <a href="#">11</a> |
| <a href="#">3.2.</a>   | PHP Header . . . . .  | <a href="#">12</a> |
| <a href="#">4.</a>     | Postcard Format . . . . .   | <a href="#">13</a> |
| <a href="#">5.</a>     | Security Considerations . . . . .                                 | <a href="#">13</a> |
| <a href="#">6.</a>     | IANA Considerations . . . . .                                     | <a href="#">13</a> |
| <a href="#">7.</a>     | Contributors . . . . .  | <a href="#">13</a> |
| <a href="#">8.</a>     | Acknowledgments . . . . .   | <a href="#">13</a> |
| <a href="#">9.</a>     | Informative References . . . . .                                  | <a href="#">14</a> |
|                        | Authors' Addresses . . . . .                                      | <a href="#">17</a> |

## [1.](#) Motivation

In order to gain detailed data plane visibility to support effective network OAM, it is important to be able to examine the trace of user packets along their forwarding paths. Such on-path flow data reflect the state and status of each user packet's real-time experience and provide valuable information for network monitoring, measurement, and diagnosis.



The telemetry data include but not limited to the detailed forwarding path, the timestamp/latency at each network node, and, in case of packet drop, the drop location and reason. The emerging programmable data plane devices allow user-defined data collection[I-D.song-opsawg-dnp4iq] or conditional data collection based on trigger events. Such on-path flow data are from and about the live user traffic, which complement with the data acquired through other passive and active OAM mechanisms such as IPFIX [RFC7011] and ICMP [RFC2925].

In-band Network Telemetry (INT) was designed to cater this need (note that although INT has been widely used, the term "in-band" here does not comply with IETF's definition. "on-path" or "in-situ" may be more accurate terms). in-situ OAM (IOAM) [I-D.brockners-inband-oam-requirements] represents the related standardization efforts. In essence, INT augments user packets with instructions to tell each network node on their forwarding paths what data to collect. The requested data are inserted into and travel along with the user packets. Some end nodes are responsible to strip off the data trace and export it to a data collector for processing.

While the concept is simple and straightforward, INT faces several technical challenges:

- o Issue 1: INT header and data processing needs to be done in data plane fast path. It may interfere with the normal traffic forwarding (e.g., leading to forwarding performance degradation) and lead to inaccurate measurements (e.g., resulting in longer latency measurements than usual). This undesirable "observer effect" is problematic to carrier networks where stringent SLA must be observed.
- o Issue 2: INT may significantly increase the user packet's original size by adding the instruction header and data at each traversed node. The longer the forwarding path and the more the data collected, the larger the packet will become. The size may exceed the path MTU so either INT cannot apply or the packet needs to be fragmented. Limiting the data size or path length reduces the effectiveness of INT. On the other hand, the INT header and data can be deeply embedded in a packet due to various transport protocol and tunnel configurations. The required deep packet header inspection and processing may be infeasible to some data plane fast path where only a limited number of header bytes are accessible.
- o Issue 3: INT requires attaching an instruction header to user packets to inform network nodes what types of data to collect. Due to the header overhead constraint and hardware-friendly



consideration, TLV is undesirable for data type encoding. Instead, IOAM use a bitmap where each bit indicates one pre-defined data type [[I-D.ietf-ippm-ioam-data](#)]. However, new use cases may require new data types. The current allocated 16-bit bitmap limits the data type scalability. The proposed bitmap extension in [[I-D.song-ippm-ioam-data-extension](#)] provides a method to support more data types but it also increases the IOAM header size.

- o Issue 4: INT header needs to be encapsulated into user packets for transport. [[I-D.brockners-inband-oam-transport](#)] has discussed several encapsulation approaches for different transport protocols. However, it is difficult to encapsulate extra header in MPLS and IPv4 networks which happens to be the most widely deployed and where the path-associated telemetry data is most wanted by operators. The proposed NVGRE encapsulation for IPv4 in [[I-D.brockners-inband-oam-transport](#)] requires a tunnel to be built between each pair of nodes which may be unrealistic for plain IP networks.
- o Issue 5: The INT header and data are vulnerable to eavesdropping and tampering as well as DoS attack. Extra protective measurement is difficult on the fast data path.
- o Issue 6: Since INT only exports the telemetry data at the designated end node, if the packet is dropped in the network, the data will be lost as well. It cannot pinpoint the packet drop location which is required for fault diagnosis. Even worse, the end node may not be aware of the lost of packet at all.

The above issues are inherent to the INT-based solutions. Nevertheless, the on-path data acquired by INT are valuable for network operators. Therefore, alternative approaches which can collect the same data but avoid or mitigate the above issues are desired. This document provides a new approach named Postcard-Based Telemetry (PBT) with two different implementation variations, each having its own trade-off and addressing some or all of the above issues. The basic idea of PBT is simple: at each node, instead of inserting the collected data into the user packets, the data are directly exported through dedicated OAM packets. Such "postcard" approach is in contrast to the "passport stamps" approach adopted by INT [[DOI 10.1145 2342441.2342453](#)]. The OAM packets or postcards can be generated by the node's slow path and transported in band or out of band, independent of the original user packets.



## **2. PBT-M: Postcard-based Telemetry with Packet Marking**

This section describes the first variation of PBT. PBT-M aims to address all the challenges of INT listed above and introduce some new benefits. We first list all the design requirements of PBT-M.

### **2.1. New Requirements**

- o Req. 1: We should avoid augmenting user packets with new headers or introducing new data plane protocols. This helps to alleviate or eliminate the issue 1, 2, 4, and 5. We expect the OAM data collecting signaling remains in data plane. Simple packet marking techniques suffice to serve this purpose. It is also possible to configure the OAM data collecting from the control plane.
- o Req. 2: We should make the scheme extensible for collecting arbitrary new data to support possible future use cases. The data set to be collected is preferred to be configured through management plane or control plane. Since there is no limitation on the types of data, any custom data including those generated by DNPs [[I-D.song-opsawg-dnp4iq](#)] can be collected. Since there is no size constraints any more, it is free to use the more flexible data set template for data type definition. This addresses the issue 2 and 3.
- o Req. 3: We should avoid interfering the normal forwarding and affecting the forwarding performance when conducting data plane OAM tasks. Hence, the collected data are better to be transported independently by dedicated OAM packets through in-band or out-of-band channels. The data collecting, processing, assembly, encapsulation, and transport are therefore decoupled from the forwarding of the corresponding user packets and can be performed in data plane slow path if necessary. This addresses the issue 1, 4, and 5.
- o Req. 4: The data collected from each node is not necessarily identical, depending on application requirements and node capability. Data for different operation modes can be collected at the same time. These requirements are either impossible or very difficult to be supported by INT in which data types collected per node are supposed to be identical and for a single mode.
- o Req. 5: The flow's path-associated data can be sensitive and the security concerns need to be carefully addressed. Sending OAM data with independent packets also makes it easy to secure the collected data without exposing it to unnecessary entities. For example, the data can be encrypted before being sent to the





collector so passive eavesdropping and man-in-the-middle attack can both be deterred. This addresses the issue 5.

- o Req. 6: Even if a user packet under inspection is dropped in network, the OAM data that have been collected should still be exported and help to diagnose the packet drop location and reason. This addresses the issue 6.

## **2.2. Solution Description**

In light of the above discussion, the sketch of the proposed solution, PBT-M, is as follows. The user packet, if its path-associated data need to be collected, is marked at the path head node. At each PBT-aware node, if the mark is detected, a postcard (i.e., the dedicated OAM packet triggered by a marked user packet) is generated and sent to a collector. The postcard contains the data requested by the management plane. The requested data are configured by the management plane through data set templates (as in IPFIX [[RFC7011](#)]). Once the collector receives all the postcards for a single user packet, it can infer the packet's forwarding path and analyze the data set. The path end node is configured to unmark the packets to its original format if necessary.

The overall architecture of PBT-M is depict in Figure 1.



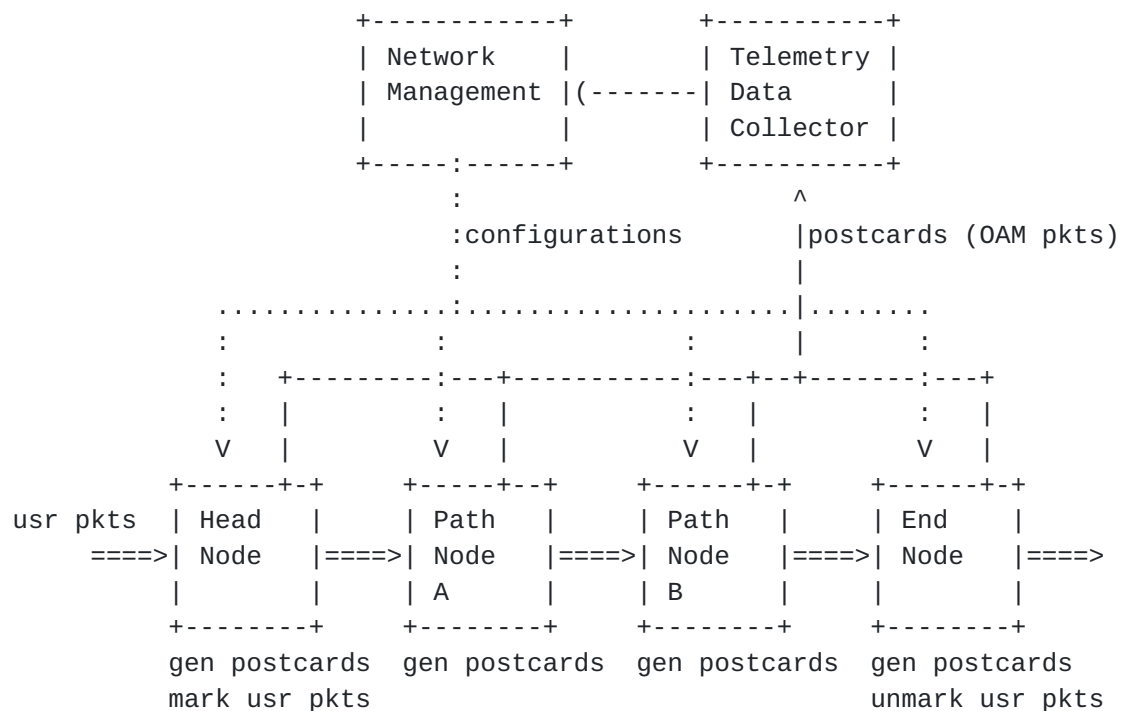


Figure 1: Architecture of PBT-M

### 2.3. New Challenges

Although PBT-M solves the issues of INT, it introduces a few new challenges.

- o Challenge 1: A user packet needs to be marked in order to trigger the path-associated data collection. Since we do not want to augment user packets with any new header fields (i.e., Req. 1), we must reuse some bit from existing header fields.
- o Challenge 2: Since the packet header will not carry OAM instructions any more, the data plane devices need to be configured to know what data to collect. However, in general, the forwarding path of a flow packet (due to ECMP or dynamic routing) is unknown beforehand. Configuring the data set for each flow at all data plane devices is expensive in terms of configuration load and data plane resources.
- o Challenge 3: Due to the variable transport latency, the dedicated OAM packets for a single packet may arrive at the collector out of order or be dropped in networks for some reason. In order to infer the packet forwarding path, the collector needs some information from the OAM packets to identify the user packet affiliation and the order of path node traversal.



## **2.4. Considerations on PBT-M Design**

To address the above challenges, we propose several design details of PBT-M.

### **2.4.1. Packet Marking**

To trigger the path-associated data collection, usually a single bit from some header field is sufficient. While no such bit is available, other packet marking techniques are needed. we discuss three possible application scenarios.

- o IPv4. IPFPM [[I-D.ietf-ippm-alt-mark](#)] is an IP flow performance measurement framework which also requires a single bit for packet coloring. The difference is that IPFPM does in-network measurement while PBT only collects and exports data at network nodes (i.e., the data analysis is done at the collector rather than in the network nodes). IPFPM suggests to use some reserved bit of the Flag field or some unused bit of the TOS field. Actually, IPFPM can be considered a subcase of PBT so the same bit can be used for PBT. The management plane is responsible to configure the actual operation mode.
- o SFC NSH. The OAM bit in NSH header can be used to trigger the path-associated data collection [[I-D.ietf-sfc-nsh](#)]. PBT does not add any other metadata to NSH.
- o MPLS. Instead of choosing a header bit, we take advantage of the synonymous flow label [[I-D.bryant-mpls-synonymous-flow-labels](#)] approach to mark the packets. A synonymous flow label indicates the path-associated data should be collected and forwarded through a postcard.

### **2.4.2. Flow Path Discovery**

By default, all PBT-aware nodes are configured to react to the marked packets by exporting some basic data such as node ID and TTL before a data set template for that flow is configured. This way, the management plane can learn the flow path dynamically.

If the management plane wants to collect the path-associated data for some flow, it configures the head node(s) with a probability or time interval for the flow packet marking. When the first marked packet is forwarded in the network, the PBT-aware nodes will export the basic data to the collector. Hence, the flow path is identified. If other types of data need to be collected, the management plane can further configure the data set template to the target nodes on the flow's path. The PBT-aware nodes would collect and export data



accordingly if the packet is marked and a data set template is present.

If for any reason, the flow path is changed. The new path nodes can be learned immediately by the collector, so the management plane controller can be informed to configure the new path nodes. The outdated configuration can be automatically timed out or explicitly revoked by the management plane controller.

#### **2.4.3. Packet Identity for Export Data Correlation**

The collector needs to correlate all the OAM packets for a single user packet. Once this is done, the TTL (or the timestamp, if the network time is synchronized) can be used to infer the flow forwarding path. The key issue here is to correlate all the postcards for a same user packet.

The first possible approach is to include the flow ID plus the user packet ID in the OAM packets. The flow ID can be the 5-tuple IP header of the user traffic. The user packet ID can be some unique information pertaining to a user packet (e.g., the sequence number of a TCP packet).

If the packet marking interval is large enough, then the flow ID itself is enough to identify the user packet. That is, we can assume all the exported OAM packets for the same flow during a short period of time belong to the same user packet.

Alternatively, if the network is synchronized, then the flow ID plus the timestamp at each node can also infer the postcard affiliation. However, some errors may occur under some circumstances. For example, if two consecutive user packets from the same flows are both marked but one exported postcard from a node is lost, then it is difficult for the collector to decide which user packet the remaining postcard belongs to. In many cases, such rare error has no catastrophic consequence therefore is tolerable.

#### **2.5. Avoid Packet Marking through Node Configuration**

It is possible to avoid needing to mark user packets yet still allowing in-band flow data collection. We could simply configure the Access Control List (ACL) to filter out the set of target flows. This approach has two potential issues: (1) Since the packet forwarding path is unknown in advance, one needs to configure all the nodes in a network to filter the flows and capture the complete data set. This wastes the precious ACL resource and is not scalable. (2) If a node cannot collect data for all the filtered packets of a flow, it needs to determine which packets to sample independently, so the





collector may not be able to receive the full set of postcards for a same user packet.

Nevertheless, since this approach does not require to touch the user packets at all, it has its unique merits: (1) User can freely choose any nodes as vantage points for data collection; (2) No need to worry that any "modified" user packets to leak out of the PBT domain; (3) It has the minimum impact to the forwarding of the user traffic.

No data plane standard is required to support this mode, except the postcard format.

### **3. PBT-I: Postcard-based Telemetry with Instruction Header**

Since PBT-M has some challenges as listed in [Section 2.3](#), this section describes another variation of PBT, which essentially compromises some of the design requirements listed in [Section 2.1](#), yet retains most of the benefits of PBT.

PBT-I can be seen as a trade-off between INT and PBT-M. PBT-I needs to add a fixed length instruction header to user packets for OAM data collection. However, the collected data will be exported through dedicated postcards. On the one hand, PBT-I violates the Req. 1 in [Section 2.1](#). It also makes it harder to meet the Req. 2. On the other hand, the overhead of the instruction header is fixed and user packets will not inflate with path length or telemetry data quantity. We also introduce an optimization to mitigate the impact on Req. 2. In return, PBT-I addresses all the challenges of PBT-M:

- o There is no need to find an existing header field to mark a user packet. PBT-I is purposely made another mode of IOAM, Per-Hop Postcard (PHP). The PHP header use the same encapsulation method for IOAM. So far, the IOAM header encapsulation methods have been defined for several protocols, including IPv6, VXLAN-GPE, NSH, SRV6 [[I-D.brockners-inband-oam-transport](#)], [[I-D.ietf-sfc-ioam-nsh](#)], GENEVE [[I-D.brockners-ippm-ioam-geneve](#)], and GRE [[I-D.weis-ippm-ioam-gre](#)]. [[I-D.song-mpls-extension-header](#)] describes the approach to encapsulate the instruction header into MPLS packets.
- o There is no need to configure the nodes about the data to be collected since the data set information is carried in the instruction header. The data set representation is identical to that of the trace modes of IOAM.
- o The PHP header contains enough information to help correlate the OAM packets belonging to a user packets. Even better, new fields are added to track the flow and the packet, so any packet under



inspection can be easily identified even in tunnels and the collector can easily check if any user packet under inspection or its OAM data packet is missing.

### 3.1. Solution Description

The sketch of the proposed solution, PBT-I, is as follows. If the path-associated data need to be collected for a user packet, a PHP header is inserted into the packet at the path head node. At each PBT-aware node, if PHP header is detected, a postcard is generated and sent to a collector. Once the collector receives all the postcards for a single user packet, it can combine and analyze the data set. The path end node is configured to remove the PHP header.

The overall architecture of PBT-I is depict in Figure 2. Note that in the figure we omit the controller which configures the nodes for necessary functions (e.g., head node encapsulation) and information (e.g., IP address of the data collector).

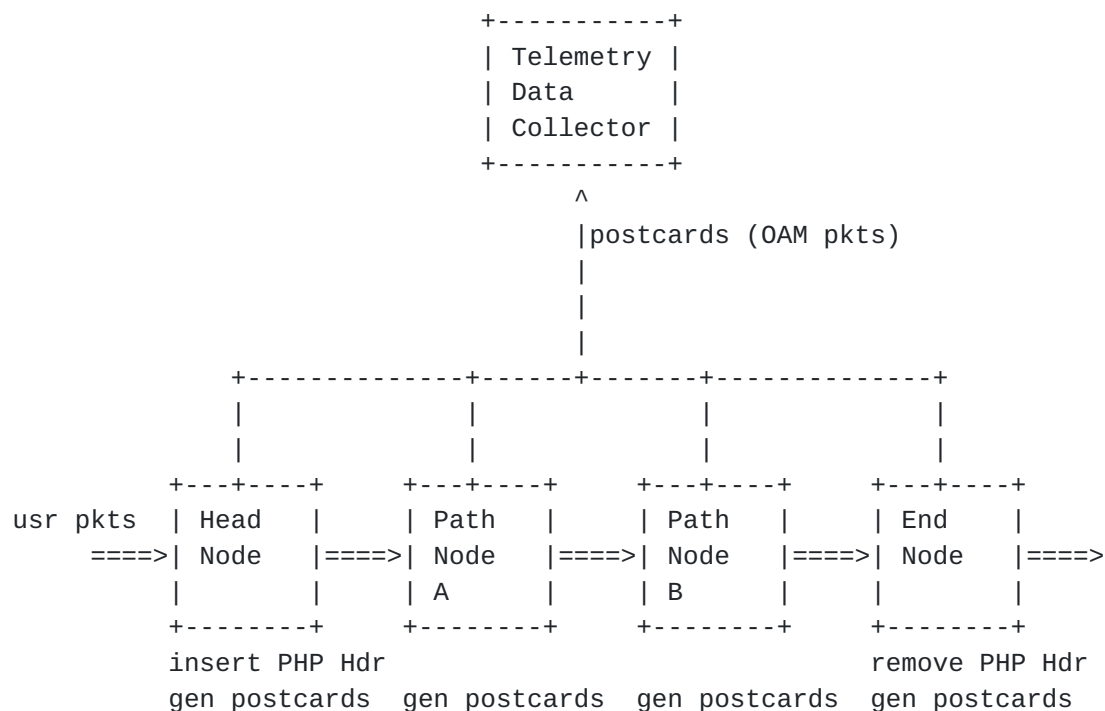


Figure 2: Architecture of PBT-I



### 3.2. PHP Header

The proposed format of PHP Header is shown in Figure 3.

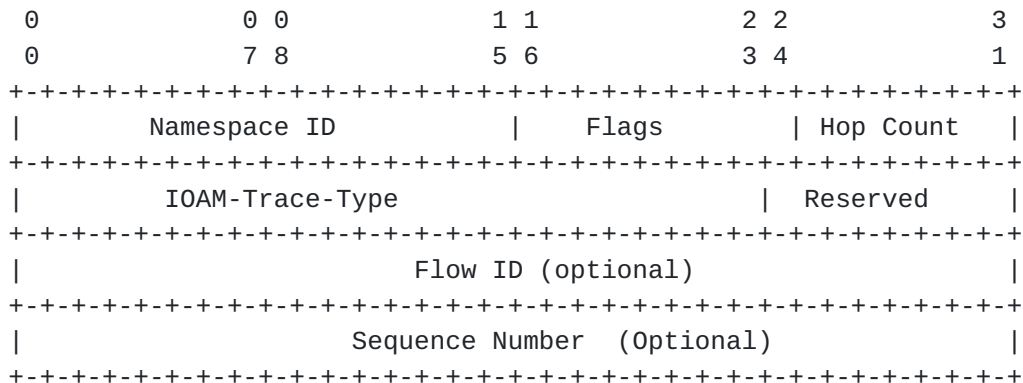


Figure 3: PHP Header Format

- o Namespace ID: the same as that defined in [\[I-D.ietf-ippm-ioam-data\]](#).
- o Flags: Extended to 9 bits from 4 bits defined in [\[I-D.ietf-ippm-ioam-data\]](#). The defined flags keeps the same definition as in [\[I-D.ietf-ippm-ioam-data\]](#).
- o Hop Count: the 8-bit Hop Count field. If it is initialized to 0, it means the field is unused and its value remains to 0 at each hop. If it is initialized to 1, it is used to count the hops of the PBT-aware nodes and its value is incremented by 1 at each PBT-aware node. The value is saturated at 127. Hop Count provides a convenient approach to order the postcards from the same user packets and help identify if any postcard is missing, in case the TTL data is absent from collected data set or nonconsecutive. Note that the data collector cannot tell whether or not some postcards are missing with nonconsecutive TTL values.
- o IOAM-Trace-Type: the same as that defined in [\[I-D.ietf-ippm-ioam-data\]](#).
- o Flow ID: the 32-bit flow ID field. If the actual flow ID is shorter than 32 bits, it is right aligned with the leading bits being filled with 0. The field is set at the head node. The flow ID can be uniformly assigned by the central controller or algorithmically generated by the head node. The latter approach cannot guarantee the uniqueness of flow ID, yet the conflict probability is small due to the large flow ID space. Flow ID



provides a convenient approach to correlate the postcards from the same flow.

- o Sequence Number: the 32-bit sequence number starting from 0 and increasing by 1 for each following monitored packet from the same flow at the head node. Sequence Number, when combining with Flow ID, provides a convenient approach to correlate the postcards from the same user packet.

Both Flow ID and Sequence Number fields are optional. These two fields are either present or absent simultaneously. Therefore, the PHP header length can be either 8 bytes or 16 bytes, which is indicated by the upper layer encapsulation header. Making these two fields optional is to cater the need of minimizing the header overhead in case the postcard correlation can still be achieved without the help of flow ID and Sequence Number.

#### **4. Postcard Format**

Postcard can use the same data export format as that used by IOAM. [[I-D.spiegel-ippm-ioam-rawexport](#)] proposes a raw format that can be interpreted by IPFIX.

#### **5. Security Considerations**

Several security issues need to be considered.

- o Eavesdrop and tamper: the postcards can be encrypted and authenticated to avoid such security threats.
- o DoS attack: PBT can be limited to a single administration domain, or the enforce mark or instruction header are checked at the domain edge. The node can rate limit the extra traffic incurred by postcards.

#### **6. IANA Considerations**

An IOAM Mode ID for PHP needs to be assigned.

#### **7. Contributors**

TBD.

#### **8. Acknowledgments**

We thank Barak Gafni, Mickey Spiegel, Frank Brockners, Zhenqiang Li, and others for reviewing and commenting the earlier version of the draft.





## 9. Informative References

[DOI\_10.1145\_2342441.2342453]

Handigol, N., Heller, B., Jeyakumar, V., MaziA(C)res, D., and N. McKeown, "Where is the debugger for my software-defined network?", Proceedings of the first workshop on Hot topics in software defined networks - HotSDN '12, DOI 10.1145/2342441.2342453, 2012.

[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. Chang, "Requirements for In-situ OAM", [draft-brockners-inband-oam-requirements-03](#) (work in progress), March 2017.

[I-D.brockners-inband-oam-transport]

Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "Encapsulations for In-situ OAM Data", [draft-brockners-inband-oam-transport-05](#) (work in progress), July 2017.

[I-D.brockners-ippm-ioam-geneve]

Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "Geneve encapsulation for In-situ OAM Data", [draft-brockners-ippm-ioam-geneve-01](#) (work in progress), June 2018.

[I-D.bryant-mpls-synonymous-flow-labels]

Bryant, S., Swallow, G., Sivabalan, S., Mirsky, G., Chen, M., and Z. Li, "[RFC6374](#) Synonymous Flow Labels", [draft-bryant-mpls-synonymous-flow-labels-01](#) (work in progress), July 2015.

[I-D.clemm-netconf-push-smart-filters-ps]

Clemm, A., Voit, E., Liu, X., Bryskin, I., Zhou, T., Zheng, G., and H. Birkholz, "Smart filters for Push Updates - Problem Statement", [draft-clemm-netconf-push-smart-filters-ps-00](#) (work in progress), October 2017.

[I-D.ietf-ippm-alt-mark]

Fioccola, G., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate Marking method for passive and hybrid performance monitoring", [draft-ietf-ippm-alt-mark-14](#) (work in progress), December 2017.



`[I-D.ietf-ippm-ioam-data]`

Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and d. daniel.bernier@bell.ca, "Data Fields for In-situ OAM", [draft-ietf-ippm-ioam-data-00](#) (work in progress), September 2017.

`[I-D.ietf-netconf-udp-pub-channel]`

Zheng, G., Zhou, T., and A. Clemm, "UDP based Publication Channel for Streaming Telemetry", [draft-ietf-netconf-udp-pub-channel-01](#) (work in progress), November 2017.

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

Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", [draft-ietf-netconf-yang-push-12](#) (work in progress), December 2017.

`[I-D.ietf-sfc-ioam-nsh]`

Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "NSH Encapsulation for In-situ OAM Data", [draft-ietf-sfc-ioam-nsh-00](#) (work in progress), May 2018.

`[I-D.ietf-sfc-nsh]`

Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", [draft-ietf-sfc-nsh-28](#) (work in progress), November 2017.

`[I-D.sambo-netmod-yang-fsm]`

Sambo, N., Castoldi, P., Fioccola, G., Cugini, F., Song, H., and T. Zhou, "YANG model for finite state machine", [draft-sambo-netmod-yang-fsm-00](#) (work in progress), October 2017.

`[I-D.song-ippm-ioam-data-extension]`

Song, H. and T. Zhou, "In-situ OAM Data Type Extension", [draft-song-ippm-ioam-data-extension-00](#) (work in progress), October 2017.

`[I-D.song-ippm-ioam-tunnel-mode]`

Song, H., Li, Z., Zhou, T., and Z. Wang, "In-situ OAM Processing in Tunnels", [draft-song-ippm-ioam-tunnel-mode-00](#) (work in progress), June 2018.



[I-D.song-mpls-extension-header]

Song, H., Li, Z., Zhou, T., and L. Andersson, "MPLS Extension Header", [draft-song-mpls-extension-header-01](#) (work in progress), August 2018.

[I-D.song-opsawg-dnp4iq]

Song, H. and J. Gong, "Requirements for Interactive Query with Dynamic Network Probes", [draft-song-opsawg-dnp4iq-01](#) (work in progress), June 2017.

[I-D.spiegel-ippm-ioam-rawexport]

Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", [draft-spiegel-ippm-ioam-rawexport-01](#) (work in progress), October 2018.

[I-D.talwar-rtgwg-grpc-use-cases]

Specification, g., Kolhe, J., Shaikh, A., and J. George, "Use cases for gRPC in network management", [draft-talwar-rtgwg-grpc-use-cases-01](#) (work in progress), January 2017.

[I-D.weis-ippm-ioam-gre]

Weis, B., Brockners, F., crhill@cisco.com, c., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "GRE Encapsulation for In-situ OAM Data", [draft-weis-ippm-ioam-gre-00](#) (work in progress), March 2018.

[RFC2925] White, K., "Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations", [RFC 2925](#), DOI 10.17487/RFC2925, September 2000, <<https://www.rfc-editor.org/info/rfc2925>>.

[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>>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, [RFC 7011](#), DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.



Authors' Addresses

Haoyu Song (editor)  
Huawei  
2330 Central Expressway  
Santa Clara, 95050  
USA

Email: haoyu.song@huawei.com

Tianran Zhou  
Huawei  
156 Beiqing Road  
Beijing, 100095  
P.R. China

Email: zhoutianran@huawei.com

Zhenbin Li  
Huawei  
156 Beiqing Road  
Beijing, 100095  
P.R. China

Email: lizhenbin@huawei.com

Jongyoon Shin  
SK Telecom  
South Korea

Email: jongyoon.shin@sk.com



