OPSAWG
Internet-Draft
Intended status: Informational
Expires: April 23, 2020

H. Song, Ed.
Futurewei
Z. Li
T. Zhou
Huawei
F. Qin
China Mobile
H. Chen
China Telecom
J. Jin
LG U+
J. Shin
SK Telecom

October 21, 2019

# In-situ Flow Information Telemetry
## draft-song-opsawg-ifit-framework-06

Abstract

For efficient network operation, most operators rely on traditional Operation, Administration and Maintenance (OAM) methods, which include proactive and reactive techniques, running in active and passive modes.  As networks increase in scale they become more susceptible to measurement accuracy and misconfiguration errors.

With the advent of programmable data-plane emerging on-path OAM techniques, such as flow telemetry, provide unprecedented insight and fast notification of network issues (e.g., jitter, increased latency, packet loss, significant bit error variations, and unequal load-balancing).

In-situ Flow Information Telemetry (iFIT) provides a method for efficiently applying underlying on-path flow telemetry techniques, applicable across various network environments.

This document outlines an iFIT framework, which enumerates several critical functional components and describes how these components are assembled together to achieve a complete and closed-loop working solution for on-path flow telemetry.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP

14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

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 April 23, 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. Requirements and Challenges

The sheer complexity of today's networks requires radical rethinking of existing methods network monitoring and troubleshooting. Current dynamic networks require "on-path" fault monitoring and traffic measurement solutions for a wide range of use cases which include intelligent management of existing network traffic, and better traffic visibility of emerging applications such as large scale Virtual Server (VS) mobility, fluid content distribution, and elastic bandwidth allocation.

Furthermore, the ability to expedite failure detection, fault localization, and recovery mechanisms, particularly in the case of soft failures or path degradation are experienced, without causing extreme or obvious disruption. This is extremely important for since these types of network issues are often difficult to localize with existing Operation, Administration and Maintenance (OAM) methods and reduce overall network efficiency.

Future networks must also support application-aware networking. Application-aware networking is an emerging industry term and typically used to describe the capacity of an intelligent network to maintain current information about user and application connections that use network resources and, as a result, the operator is able to optimize the network resource usage and monitoring to ensure application and traffic optimality.

Application-aware network operation is important for user SLA compliance, service path enforcement, fault diagnosis, and network resource optimization. A family of on-path flow telemetry techniques, including In-situ OAM (IOAM) [I-D.brockners-inband-oam-data], Postcard Based Telemetry (PBT) [I-D.song-ippm-postcard-based-telemetry], In-band Flow Analyzer (IFA) [I-D.kumar-ippm-ifa], Enhanced Alternate Marking (EAM)

[I-D.zhou-ippm-enhanced-alternate-marking], and Hybrid Two Steps (HTS) [I-D.mirsky-ippm-hybrid-two-step], are emerging, which can provide flow information on the entire forwarding path on a per-packet basis in real time. These on-path flow telemetry techniques are very different from the previous active and passive OAM schemes in that they directly modify the user packets. Given the unique characteristics of the aforementioned techniques, we may categorize these on-path telemetry techniques as the hybrid OAM type III, supplementing the classification defined in [RFC7799].

These techniques are invaluable for application-aware network operations not only in data center and enterprise networks but also in carrier networks which may cross multiple domains. Carrier network operators have shown strong interests in utilizing such techniques for various purposes. For example, it is vital for the operators who offer the bandwidth intensive, latency and loss sensitive services such as video streaming and gaming to closely monitor the relevant flows in real time as the indispensable first step for any further measure.

However, successfully applying such techniques in carrier networks poses several practical challenges:

- C1: On-path flow telemetry incurs extra packet processing which may strain the network data plane. The potential impact on the forwarding performance creates an unfavorable "observer effect" which not only damages the fidelity of the measurement but also defies the purpose of the measurement.

- C2: On-path flow telemetry can generate a huge amount of OAM data which may claim too much transport bandwidth and inundate the servers for data collection, storage, and analysis. Increasing the data handling capacity is technically viable but expensive. For example, assume IOAM is applied to all the traffic. One node will collect a few tens of bytes as telemetry data for each packets. The whole forwarding path might accumulate a data trace with a size similar to the average size of the original packets. Exporting the telemetry data will consume almost half of the network bandwidth.

- C3: The collectible data defined currently are essential but limited. As the network operation evolves to be declarative (intent-based) and automated, and the trends of network virtualization, network convergence, and packet-optical integration continue, more data will be needed in an on-demand and interactive fashion. Flexibility and extensibility on data defining, acquisition, and filtering, must be considered.

- C4: If we were to apply some on-path telemetry technique in today's carrier networks, we must provide solutions to tailor the provider's network deployment base and support an incremental deployment strategy. That is, we need to support established encapsulation schemes for various predominant protocols such as Ethernet, IPv4, and MPLS with backward compatibility and properly handle various transport tunnels.

- C5: Applying only a single underlying telemetry technique may lead to defective result. For example, packet drop can cause the lost of the flow telemetry data and the packet drop location and reason remains unknown if only In-situ OAM trace option is used. A comprehensive solution needs the flexibility to switch between different underlying techniques and adjust the configurations and parameters at runtime.

- C6: Development of simplified on-path telemetry primitives and models, including: telemetry data (e.g., nodes, links, ports, paths, flows, timestamps) query primitives. These may be used by an API-based telemetry service for external applications, for monitoring end-to-end latency measurement of network paths and application latency calculation.

## 2. iFIT Framework Overview

To address these challenges, we propose a framework based on multiple network operators' requirements and the common industry practice, which can help to build a workable on-path flow telemetry solution. We name the framework "In-situ Flow Information Telemetry" (iFIT) to reflect the fact that this framework is dedicated to the on-path telemetry data about user/application flow experience. As a solution framework, iFIT works a level higher than any specific OAM techniques, be it active, passive, or hybrid. The framework is built up on a few architectural components. By assembling these components together, a closed-loop is formed to provide a complete solution for a particular static, dynamic, and interactive telemetry applications.

iFIT is an open framework. It does not enforce any implementation details for each component. Users are free to pick one or more underlying techniques and design their own algorithms and architectures to fit in each component and make all the components work in concert.

The network architecture that applies iFIT is shown in Figure 1. The iFIT domain is confined between the iFIT head nodes and the iFIT end nodes. An iFIT domain may cross multiple network domains. iFIT support two basic on-path telemetry data collection modes: passport mode (e.g., IOAM trace option and IFA), in which telemetry data are

carried in user packets and exported at the iFIT end nodes, and postcard mode (e.g., PBT), in which each node in the iFIT domain may export telemetry data through independent OAM packets. Note that the boundary between the two modes can be blurry. An application only need to mix the two modes.

```
                          +---------------------------------+
                          |                                 |
                          |        iFIT Applications        |
                          |                                 |
                          +---------------------------------+
                                 ^                    ^
                                 |                    |
                                 V                    |
                          +------------+        +-----+-----+
                          |            |        |           |
                          | Controller |        | Collector |
                          |            |        |           |
                          +-----:------+        +-----------+
                                :                     ^
                                :configuration        |telemetry data
                                :                     |
                 ...............:.....................|..........
                 :             :                 :    |         :
                 :   +---------:---+-------------:---++---------:---+
                 :   |         :   |             :   |          :   |
                 V   |         V   |             V   |          V   |
              +------+-+     +-----+--+       +------+-+     +------+-+
    usr pkts  | iFIT   |     | Path   |       | Path   |     | iFIT   |
         ====>| Head   |====>| Node   |==//==>| Node   |====>| End    |====>
              | Node   |     | A      |       | B      |     | Node   |
              +--------+     +--------+       +--------+     +--------+
```

Figure 1: iFIT Network Architecture

## 2.1. Passport vs. Postcard

[passport-postcard] first uses the analogy of passport and postcard to describe how the packet trace data can be collected and exported. In the passport mode, each node on the path adds the telemetry data to the user packets. The accumulated data trace is exported at a configured end node. In the postcard mode, each node directly exports the telemetry data using an independent packet while the user packets are intact.

A prominent advantage of the passport mode is that it naturally retains the telemetry data correlation along the entire path. The passport mode also reduces the number of data export packets and the bandwidth consumed by the data export packets. These can help to make the data collector and analyzer's work easier. On the other hand, the passport mode requires more processing on the user packets and increases the size of user packets, which can cause various problems. Some other issues are documented in [I-D.song-ippm-postcard-based-telemetry].

The postcard mode provides a perfect complement to the passport mode. It addresses most of the issues faced by the passport mode, at a cost of needing extra efforts to correlate the postcard packets.

## 3. Architectural Components of iFIT

The key components of iFIT are listed as follows:

o Smart flow and data selection policy to address C1.

o Smart data export to address C2.

o Dynamic network probe to address C3.

o Encapsulation and tunneling to address C4.

o On-demand technique selection and integration to address C5.

Next we provide the detailed description of each component.

### 3.1. Smart Flow and Data Selection

In most cases, it is impractical to enable the data collection for all the flows and for all the packets in a flow due to the potential performance and bandwidth impacts. Therefore, a workable solution must select only a subset of flows and flow packets to enable the data collection, even though this means the loss of some information.

In data plane, the Access Control List (ACL) provides an ideal means to determine the subset of flow(s). [I-D.song-ippm-ioam-data-validation-option] describes how one can set a sample rate or probability to a flow to allow only a subset of flow packets to be monitored, how one can collect different set of data for different packets, and how one can disable or enable data collection on any specific network node. The document further introduces enhancement to IOAM to allow any node to accept or deny the data collection in full or partially.

Based on these flexible mechanisms, iFIT allows applications to apply smart flow and data selection policies to suit the requirements. The applications can dynamically change the policies at any time based on the network load, processing capability, focus of interest, and any other criteria. We have developed some adaptive algorithm which can limit the performance impact and yet achieve the satisfactory telemetry data density.

#### 3.1.1. Use Case: Sketch-guided Elephant Flow Selection

Network operators are usually more interested in elephant flows which consume more resource and are sensitive to network condition changes. We implement a CountMin Sketch [CMSketch] on the data path of the head nodes, which identifies and reports the elephant flows periodically. The controller maintains a current set of elephant flows and dynamically enables the on-path telemetry for only these flows.

#### 3.1.2. Use Case: Adaptive Packet Sampling

Applying on-path telemetry on all packets of selected flows can still be out of reach. We should set a sample rate for these flows and only enable telemetry on the sampled packets. However, the head nodes have no clue on the proper sampling rate. An overly high rate would exhaust the network resource and even cause packet drops; An overly low rate, on the contrary, would result in the loss of information and inaccuracy of measurements.

We can use an adaptive approach based on the network conditions to dynamically adjust the sampling rate. Every node gives user traffic forwarding higher priority than telemetry data export. In case of network congestion, the telemetry can sense some signals from the data collected (e.g., deep buffer size, long delay, packet drop, and data loss). The controller uses these signals to adjust the packet sampling rate. In each adjustment period (i.e., RTT of the feedback loop), the sampling rate is either decreased or increased in response of the signals. We can use the AIMD policy similar to the TCP flow control mechanism for the rate adjustment.

### 3.2. Smart Data Export

The flow telemetry data can catch the dynamics of the network and the interactions between user traffic and network. Nevertheless, the data inevitably contain redundancy. It is advisable to remove the redundancy from the data in order to reduce the data transport bandwidth and server processing load.

In addition to efficiently encode the export data (e.g., IPFIX [RFC7011] or protobuf [1]), iFIT can also cache the data and send the accumulated data in batch if the data is not time sensitive. Various deduplication and compression techniques can be applied on the batch data.

From the application perspective, an application may only be interested in some special events which can be derived from the telemetry data. For example, in case that the forwarding delay of a packet exceeds a threshold or a flow changes its forwarding path is of interest, it is unnecessary to send the original raw data to the data collecting and processing servers. Rather, iFIT takes advantage of the in-network computing capability of network devices to process the raw data and only push the event notifications to the subscribing applications.

#### 3.2.1. Use Case: On-demand Anomaly Monitor

Network operators are interested in the anomalies such as path change, network congestion, and packet drop. Such anomalies are hidden in raw telemetry data (e.g., path trace, timestamp). We can describe such anomalies as events and program them into the device data plane. Only the triggered events are exported. For example, if a new flow appears at any node, a path change event is triggered; if the packet delay exceeds a predefined threshold in a node, the congestion event is triggered; if a packet is dropped due to buffer overflow, a packet drop event is triggered.

### 3.3. Dynamic Network Probe

Due to the limited data plane resource, it is unlikely one can provide all the data all the time. On the other hand, the data needed by applications may be arbitrary but ephemeral. It is critical to meet the dynamic data requirements with limited resource.

Fortunately, data plane programmability allows iFIT to dynamically load new data probes. These on-demand probes are called Dynamic Network Probes (DNP) [I-D.song-opsawg-dnp4iq]. DNP is the technique to enable probes for customized data collection in different network planes. When working with IOAM or PBT, DNP is loaded to the data plane through incremental programming or configuration. The DNP can effectively conduct data generation, processing, and aggregation.

DNP introduces enough flexibility and extensibility to iFIT. It can implement the optimizations for export data reduction motioned in the previous section. It can also generate custom data as required by today and tomorrow's applications.

The aforementioned sketch module and anomaly triggers can all be implemented as DNPs so they can be loaded to or unloaded from the data plane dynamically based on application requirments.

## 3.4. Encapsulation and Tunneling

Since MPLS and IPv4 network are still prevalent in carrier networks. iFIT provides solutions to apply the on-path flow telemetry techniques in such networks. PBT-M [I-D.song-ippm-postcard-based-telemetry] does not introduce new headers to the packets so the trouble of encapsulation for a new header is avoided. In case a technique that requires a new header is preferred, [I-D.song-mpls-extension-header] provides a means to encapsulate the extra header using an MPLS extension header. As for IPv4, it is possible to encapsulate the new header in an IP option. For example, RAO [RFC2113] can be used to indicate the presence of the new header. A recent proposal [I-D.herbert-ipv4-eh] that introduces the IPv4 extension header may lead to a long term solution.

In carrier networks, it is common for user traffic to traverse various tunnels for QoS, traffic engineering, or security. iFIT supports both the uniform mode and the pipe mode for tunnel support as described in [I-D.song-ippm-ioam-tunnel-mode]. With such flexibility, the operator can either gain a true end-to-end visibility or apply a hierarchical approach which isolates the monitoring domain between customer and provider.

## 3.5. On-demand Technique Selection and Integration

With multiple underlying data collection and export techniques at its disposal, iFIT can flexibly adapt to different network conditions and different application requirements.

For example, depending on the types of data that are of interest, iFIT may choose either IOAM or PBT to collect the data; if an application needs to track down where the packets are lost, it may switch from IOAM to PBT.

iFIT can further integrate multiple data plane monitoring and measurement techniques together and present a comprehensive data plane telemetry solution to network operating applications.

## 3.6. iFIT Closed-Loop Architecture

Working together, the aforementioned components form a closed-loop for complete iFIT applications, as shown in Figure 2.

```
                           +---------------------+
                           |                     |
                    +------+  iFIT Applications  |<------+
                    |      |                     |       |
                    |      +---------------------+       |
                    |         Technique Selection        |
                    |         and Integration            |
                    |                                    |
                    |Smart Flow                    Smart |
                    |and Data     closed-loop      Data  |
                    |Selection                     Export|
                    |                                    |
                    |                               +----+----+
                    V                              +---------+|
              +----------+ Encapsulation          +---------+||
              |  iFIT    | and Tunneling          |  iFIT   |||
              |  Head    |----------------------->|         ||+
              |  Node    |                        |  Nodes  |+
              +----------+                        +---------+
                  DNP                                DNP
```

Figure 2: iFIT Closed-Loop Architecture

An iFIT application would pick a suite of telemetry techniques based on its requirements and apply an initial technique to the data plane. It then configures the iFIT head nodes to decide the initial target flows/packets and telemetry data set, the encapsulation and tunneling scheme based on the underlying network architecture, and the iFIT-capable nodes to decide the initial telemetry data export policy. Based on the network condition and the analysis results of the telemetry data, the iFIT application can change the telemetry technique, the flow/data selection policy, and the data export approach in realtime without breaking the normal network operation. Many of such dynamic changes can be done through loading and unloading DNPs.

We should avoid confusion between this closed telemetry loop and the closed control loop.  The latter term is often used in the context of network automation.  In such a closed control loop, telemetry also plays an important role.  Based on the telemetry results, applications can automatically change the network policy or configuration.  In such a context, iFIT is just a part of the loop.

## 4. Intelligent Closed-Loop Network Telemetry Applications

The closed-loop nature of the iFIT framework allows numerous new applications which enable future network operation architecture.

In general, it is resource consuming to monitor continuously all the flows and all the paths of the network. So a flexible and dynamic performance monitoring approach is desired. Some concepts like the Interactive Query with Dynamic Network Probes, as previously described, go in this direction.

Another example is shown in [I-D.ietf-ippm-multipoint-alt-mark], which that describes an intelligent performance management based on the network condition. The idea is to split the monitoring network into Clusters. The Cluster partition that can be applied to every type of network graph and the possibility to combine Clusters at different levels enable the so called Network Zooming. It allows a Controller to calibrate the network telemetry, so that it can start without examining in depth and monitor the network as a whole. In case of necessity (packet loss or too high delay), an immediate detailed analysis can be reconfigured. In particular, the Controller, that is aware of the network topology, can set up the most suited Cluster partition by changing the traffic filter or activate new measurement points and the problem can be localized with a step-by-step process.

To apply this mechanism an iFIT application on top of the controllers can manage and the iFIT closed loop allows its dynamic and flexible operation.

## 5. Summary and Future Work

iFIT is a framework for applying on-path data plane telemetry techniques. Combining with algorithmic and architectural schemes that fit into the framework components, iFIT framework enables a practical telemetry solution based on two basic on-path traffic data collection modes: passport and postcard.

The operation of iFIT differs from both active OAM and passive OAM as defined in [RFC7799]. It does not generate any active probe packets or passively observe unmodified user packets. Instead, it modifies selected user packets to collect useful information about them. Therefore, the iFIT operation can be considered the hybrid type III mode, which can provide more flexible and accurate network OAM.

More challenges and corresponding solutions for iFIT may need to be covered. For example, how iFIT can fit in the big picture of autonomous networking and support closed control loops. A complete

iFIT framework should also consider the cross-domain operations. We leave these topics for future revisions.

## 6. Security Considerations

No specific security issues are identified other than those have been discussed in the drafts on on-path flow information telemetry.

## 7. IANA Considerations

This document includes no request to IANA.

## 8. Contributors

Other major contributors of this document include Giuseppe Fioccola and Daniel King.

## 9. Acknowledgments

We thank Shwetha Bhandari and Joe Clarke for their constructive suggestions for improving this document.

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

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

### 10.2. Informative References

[CMSketch] Cormode, G. and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications", 2005, <http://dx.doi.org/10.1016/j.jalgor.2003.12.001>.

[I-D.brockners-inband-oam-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-brockners-inband-oam-data-07 (work in progress), July 2017.

[I-D.herbert-ipv4-eh]
Herbert, T., "IPv4 Extension Headers and Flow Label", draft-herbert-ipv4-eh-01 (work in progress), May 2019.

[I-D.ietf-ippm-multipoint-alt-mark]
Fioccola, G., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate Marking method for passive and hybrid performance monitoring", draft-ietf-ippm-multipoint-alt-mark-02 (work in progress), July 2019.

[I-D.kumar-ippm-ifa]
Kumar, J., Anubolu, S., Lemon, J., Manur, R., Holbrook, H., Ghanwani, A., Cai, D., Ou, H., and L. Yizhou, "Inband Flow Analyzer", draft-kumar-ippm-ifa-01 (work in progress), February 2019.

[I-D.mirsky-ippm-hybrid-two-step]
Mirsky, G., Lingqiang, W., and G. Zhui, "Hybrid Two-Step Performance Measurement Method", draft-mirsky-ippm-hybrid-two-step-04 (work in progress), October 2019.

[I-D.song-ippm-ioam-data-validation-option]
Song, H. and T. Zhou, "In-situ OAM Data Validation Option", draft-song-ippm-ioam-data-validation-option-02 (work in progress), April 2018.

[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-ippm-postcard-based-telemetry]
Song, H., Zhou, T., Li, Z., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry", draft-song-ippm-postcard-based-telemetry-05 (work in progress), September 2019.

[I-D.song-mpls-extension-header]
Song, H., Li, Z., Zhou, T., and L. Andersson, "MPLS Extension Header", draft-song-mpls-extension-header-02 (work in progress), February 2019.

[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.zhou-ippm-enhanced-alternate-marking]
Zhou, T., Fioccola, G., Li, Z., Lee, S., Cociglio, M., and Z. Li, "Enhanced Alternate Marking Method", draft-zhou-ippm-enhanced-alternate-marking-03 (work in progress), July 2019.

[passport-postcard]
Handigol, N., Heller, B., Jeyakumar, V., Mazieres, D., and N. McKeown, "Where is the debugger for my software-defined network?", 2012, <https://doi.org/10.1145/2342441.2342453>.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <https://www.rfc-editor.org/info/rfc2113>.

[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>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

### 10.3. URIs

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

Authors' Addresses

Haoyu Song (editor)
Futurewei
2330 Central Expressway
Santa Clara
USA

Email: haoyu.song@futurewei.com

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

Email: lizhenbin@huawei.com

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

Email: zhoutianran@huawei.com

Fengwei Qin
China Mobile
No. 32 Xuanwumenxi Ave., Xicheng District
Beijing, 100032
P.R. China

Email: qinfengwei@chinamobile.com

Huanan Chen
China Telecom
P. R. China

Email: chenhuan6@chinatelecom.cn

Jaewhan Jin
LG U+
South Korea

Email: daenamu1@lguplus.co.kr

Jongyoon Shin
SK Telecom
South Korea

Email: jongyoon.shin@sk.com