In-situ Flow Information Telemetry
draft-song-opsawg-ifit-framework-14

Abstract

   As network scale increases and network operation becomes more
   sophisticated, traditional Operation, Administration and Maintenance
   (OAM) methods, which include proactive and reactive techniques,
   running in active and passive modes, are no longer sufficient to meet
   the monitoring and measurement requirements. On-path telemetry
   techniques which provide high-precision flow insight and real-time
   issue notification are emerging to support suitable quality of
   experience for users and applications, and fault or network
   deficiency identification before they become critical.

   Centering on the new data plane on-path telemetry techniques, this
   document outlines a high-level framework to provide an operational
   environment that utilizes these techniques to enable the collection
   and correlation of performance information from the network. The
   framework identifies the components that are needed to coordinate the
   existing protocol tools and telemetry mechanisms, and addresses key
   deployment challenges for flow-oriented on-path telemetry techniques,
   especially in carrier networks.

   The framework is informational and intended to guide system designers
   attempting to apply the referenced techniques as well as to motivate
   further work to enhance the ecosystem .

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 4, 2021.

Copyright Notice

Table of Contents

1. Introduction

   Efficient network operation increasingly relies on high-quality data-
   plane telemetry to provide the necessary visibility. Traditional
   Operation, Administration and Maintenance (OAM) methods, which
   include proactive and reactive techniques, running both active and
   passive modes, are no longer sufficient to meet the monitoring and
   measurement requirements. The complexity of today's networks and
   service quality requirements demand new high-precision and real-time
   techniques.

   The ability to expedite network failure detection, fault
   localization, and recovery mechanisms, particularly in the case of
   soft failures or path degradation is expected, without causing
   service disruption. Networks also become application-aware.
   Application-aware networking is an industry term used to describe the
   capacity of a network to maintain current information about users and
   application connections which may be used to optimize the network
   resource usage and improve the quality of service.

   The emerging on-path telemetry techniques can provide high-precision
   flow insight and real-time network issue notification (e.g., jitter,
   latency, packet loss, significant bit error variations, and unequal
   load-balancing). On-path telemetry refers to the data-plane
   telemetry techniques that directly tap and measure network traffic by
   embedding instructions or metadata into user packets. The data
   provided by on-path telemetry are especially useful for SLA
   compliance, user experience enhancement, service path enforcement,

fault diagnosis, and network resource optimization. It is essential
to recognize that existing work on this topic includes a variety of
on-path telemetry techniques, including In-situ OAM(IOAM)
[I-D.ietf-ippm-ioam-data], IOAM Direct Export (DEX)
[I-D.ietf-ippm-ioam-direct-export], Postcard-based Telemetry(PBT)
[I-D.song-ippm-postcard-based-telemetry], Enhanced Alternate Marking
(EAM) [I-D.zhou-ippm-enhanced-alternate-marking], and Hybrid Two
Steps (HTS) [I-D.mirsky-ippm-hybrid-two-step], have been proposed,
which can provide flow information on the entire forwarding path on a
per-packet basis in real-time. The aforementioned on-path telemetry
techniques differ from the active and passive OAM schemes discussed
earlier in that, they directly modify and monitor the user packets so
as to achieve high measurement accuracy. Formally, these on-path
telemetry techniques can be classified as the OAM hybrid type I,
since they involve "augmentation or modification of the stream of
interest, or employment of methods that modify the treatment of the
streams", according to [RFC7799].

On-path telemetry is useful for application-aware networking
operations not only in data center and enterprise networks but also
in carrier networks which may cross multiple domains. Carrier
network operators have shown interest in utilizing such techniques
for various purposes. For example, it is critical for the operators
who offer high-bandwidth, latency and loss-sensitive services such as
video streaming and online gaming to closely monitor the relevant
flows in real-time as the basis for any further optimizations.

This framework document is intended to guide system designers
attempting to use the referenced techniques as well as to motivate
further work to enhance the telemetry ecosystem. It highlights
requirements and challenges, outlines vital techniques that are
applicable, and provides examples of how these might be applied for
critical use cases.

The document scope is discussed in Section 1.3.

1.1. Classification and Modes of On-path Telemetry

The operation of on-path telemetry differs from both active OAM and
passive OAM as defined in [RFC7799]. It does not generate any active
probe packets or passively observes unmodified user packets.
Instead, it modifies selected user packets in order to collect useful
information about them. Therefore, the operation can be categorized
as the hybrid OAM type I mode per [RFC7799], which can provide more
flexible and accurate network monitoring and measurement.

This hybrid type OAM can be further partitioned into two modes.
[passport-postcard] first uses the metaphor of "passport" and

"postcard" to describe how the on-path data can be collected and
exported. In the passport mode, each node on the path adds the
telemetry data to the user packets (i.e., stamp the passport). 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 (i.e., send a postcard) while the user packets are
intact. It is possible to combine the two modes together in one
solution. We call this the hybrid mode.

Figure 1 shows the classification of the existing on-path telemetry
techniques.

```
 +-----------+-------------+-------------+--------------+
 |  Mode     | Passport    | Postcard    | Hybrid       |
 +-----------+-------------+-------------+--------------+
 |           | IOAM Trace  | IOAM DEX    | Multicast Te-|
 | Technique | IOAM E2E    | PBT-M       | lemetry      |
 |           | EAM         |             | HTS          |
 +-----------+-------------+-------------+--------------+
```

        Figure 1: ON-path Telemetry Technique Classification

IOAM Trace and E2E options are described in
[I-D.ietf-ippm-ioam-data]. EAM is described in
[I-D.zhou-ippm-enhanced-alternate-marking]. IOAM DEX option is
described in [I-D.ietf-ippm-ioam-direct-export] PBT-M is described in
[I-D.song-ippm-postcard-based-telemetry]. Multicast Telemetry is
described in [I-D.song-multicast-telemetry]. HTS is described in
[I-D.mirsky-ippm-hybrid-two-step].

The advantages of the passport mode include :

o It naturally retains the telemetry data correlation along the
   entire path. The self-describing feature eases the data
   consumption.

o The on-path data for a packet is only exported once so the data
   export overhead is low.

o Only the head and end nodes of the paths need to be configured so
   the configuration overhead is low.

The disadvantages of the passport mode include :

o The telemetry data carried by user packets inflate the packet
   size, which is undesirable or prohibitive.

o Approaches for encapsulating the instruction header and data in
   transport protocols need to be standardized, which is challenging.

o Carrying sensitive data along the path is vulnerable to security
   and privacy breach.

o If a packet is dropped on the path, the data collected so far are
   also lost.

The postcard mode complements the passport mode by addressing the
issues of the passport mode. The advantages of the postcard mode
include:

o Either there is no packet header overhead (e.g., PBT-M) or the
   overhead is small and fixed (e.g., IOAM DEX).

o The encapsulation requirement can be avoided (e.g., PBT-M).

o The telemetry data can be secured.

o Even if a packet is dropped on the path, the partial data
   collected so far are still available.

The disadvantages of the postcard mode include:

o Telemetry data are spread in multiple postcards so extra effort is
   needed to correlate the data.

o Every node exports a postcard for a packet which increases the
   data export overhead.

o In case of PBT-M, every node on the path needs to be configured,
   so the configuration overhead is high.

o In case of IOAM DEX, the encapsulation requirement remains.

The hybrid mode either tailors for some specific application scenario
(e.g., Multicast Telemetry) or provides some alternative approach
(e.g., HTS).

1.2. Requirements and Challenges

Although on-path telemetry is beneficial, successfully applying such
techniques in carrier networks must consider performance,
deployability, and flexibility. Specifically, we need to address the
following practical deployment challenges:

o C1: On-path telemetry incurs extra packet processing which may
   cause stress on the network data plane. The potential impact on
   the forwarding performance creates an unfavorable "observer
   effect". This will not only damages the fidelity of the
   measurement but also defies the purpose of the measurement. For
   example, the growing IOAM data per hop can negatively affect
   service levels by increasing the serialization delay and header
   parsing delay.

o C2: On-path telemetry can generate a considerable amount of 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, if IOAM is applied to all the traffic, one node may
   collect a few tens of bytes as telemetry data for each packet.
   The whole forwarding path might accumulate a data-trace with a
   size similar to or even exceeding that of the original packet.
   Transporting the telemetry data alone is projected to consume
   almost half of the network bandwidth, plus it creates significant
   back-end data handling and storage requirements.

o 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, wireline and wireless convergence, and packet-
   optical integration continue, more data is needed in an on-demand
   and interactive fashion. Flexibility and extensibility on data
   defining, aggregation, acquisition, and filtering, must be
   considered.

o C4: Applying only a single underlying on-path telemetry technique
   may lead to a defective result. For example, packet drop can
   cause the loss of the flow telemetry data and the packet drop
   location and reason remains unknown if only the 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. Thus, system-level
   orchestration is needed.

o C5: If we were to apply some on-path telemetry technique in
   today's carrier operator 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, IPv6, and MPLS with backward
   compatibility and properly handle various transport tunnels.

o C6: The development of simplified on-path telemetry primitives and
   models for configuration and queries is essential. Telemetry
   models may be utilized via an API-based telemetry service for
   external applications, for end-to-end performance measurement and
   application performance monitoring. The standard-based protocols
   and methods are needed for network configuration and programming,
   and telemetry data processing and export, to provide
   interoperability.

1.3. Scope

Following the network telemetry framework discussed in
[I-D.ietf-opsawg-ntf], this document focuses on the on-path
telemetry, a specific class of data-plane telemetry techniques, and
provides a high-level framework which addresses the aforementioned
challenges for deployment, especially in carrier operator networks.

This document aims to clarify the problem space, essential
requirements, and summarizes best practices and general system design
considerations. The framework helps to analyze the current standard
status and identify gaps, and to motivate new standard works to
complete the ecosystem. This document provides some examples to show
some novel network telemetry applications under the framework.

As an informational document, it describes an open framework with a
few key components. The framework does not enforces any specific
implementation on each component, neither does it define interfaces
(e.g., API, protocol) between components. The choice of underlying
on-path telemetry techniques and other implementation details is
determined by application implementer. Therefore, the framework is
not a solution specification. It only provides a high-level overview
and is not necessarily a mandatory recommendation for on-path
telemetry applications. Implementation of the framework is
implementor specific and may utilize functional components and
techniques outlined in this document.

The standardization of the underlying techniques and interfaces
mentioned in this document is undertaken by various working groups.
Due to the limited scope and intended status of this document, it has
no overlap or conflict with those works.

1.4. Glossary

This section defines and explains the acronyms and terms used in this
document.

On-path Telemetry: Remotely acquiring performance and behavior data
   about network flows on a per-packet basis on the packet's

forwarding path. The term refers to a class of data plane
telemetry techniques, including IOAM, PBT, EAM, and HTS. Such
techniques may need to mark user packets, or insert instruction or
metadata to the headers of user packets.

IFIT: In-situ Flow Information Telemetry, pronounced as "I-Fit".
The name of a high-level reference framework that shows how
network data-plane monitoring applications can address the
deployment challenges of the flow-oriented on-path telemetry
techniques.

IFIT Domain: A network domain in which an on-path telemetry
application operates. The network domain contains multiple
forwarding devices, such as routers and switches, that are capable
of IFIT-specific functions. It also contains a logically
centralized controller whose responsibility is to apply IFIT-
specific configurations and functions to IFIT-capable forwarding
devices, and to collect and analyze the on-path telemetry data
from those devices. An IFIT domain contains multiple network node
capable of IFIT-specific functions. We name all the entry nodes
to an IFIT domain head nodes and all the exit nodes end nodes. A
path in an IFIT domain starts from a head node and ends at an end
node. Usually the instruction header encapsulation or packet
marking, if needed, happens at the head nodes; the instruction
header decapsulation or packet unmarking, if needed, happens at
the end nodes.

Reflective Telemetry: The telemetry functions in a dynamic and
interactive fashion. A new telemetry action is provisioned as a
result of self-knowledge acquired through prior telemetry actions.

1.5. 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.

2. IFIT Overview

To address the challenges mentioned above, we present a high-level
framework based on multiple network operators' requirements and
common industry practice, which can help to build a workable and
efficient on-path telemetry application. We name the framework "In-
situ Flow Information Telemetry" (IFIT) to reflect the fact that this
framework is dedicated to on-path telemetry data about user and
application traffic flows. As a reference framework, IFIT covers a

class of on-path telemetry techniques and works a level higher than
any specific underlying technique. The framework is comprised of
some key functional components (Section 3). By assembling these
components, IFIT supports reflective telemetry that enables
autonomous network operations (Section 4).

2.1. Typical Deployment of IFIT

Figure 2 shows a typical deployment scenario of IFIT.

```
                           Application
              +--------------------------------------+
              |                Controller            |
              | +------------+       +-----------+ |
              | | Configure  |       | Collector | |
              | |     &      |<-------|     &     | |
              | | Control    |       | Analyzer  | |
              | +-----:------+       +-----------+ |
              |       :                      ^      |
              +-------:--------------------|-------+
                      :configuration       |telemetry data
                      :& action            |
           ...........:...................|..........
           :          :               :   |        :
           :    +---------:---+------------:---++---------:---+
           :    |         :   |            :   |         :   |
           V    |         V   |            V   |         V   |
        +------+-+      +-----+--+      +------+-+      +------+-+
  packets| Head   |      | Path   |      | Path   |      | End    |
     ==>| Node   |====>| Node   |==//==>| Node   |====>| Node   |==>
        |        |      | A      |      | B      |      |        |
        +--------+      +--------+      +--------+      +--------+

        |<---                IFIT Domain                --->|
```
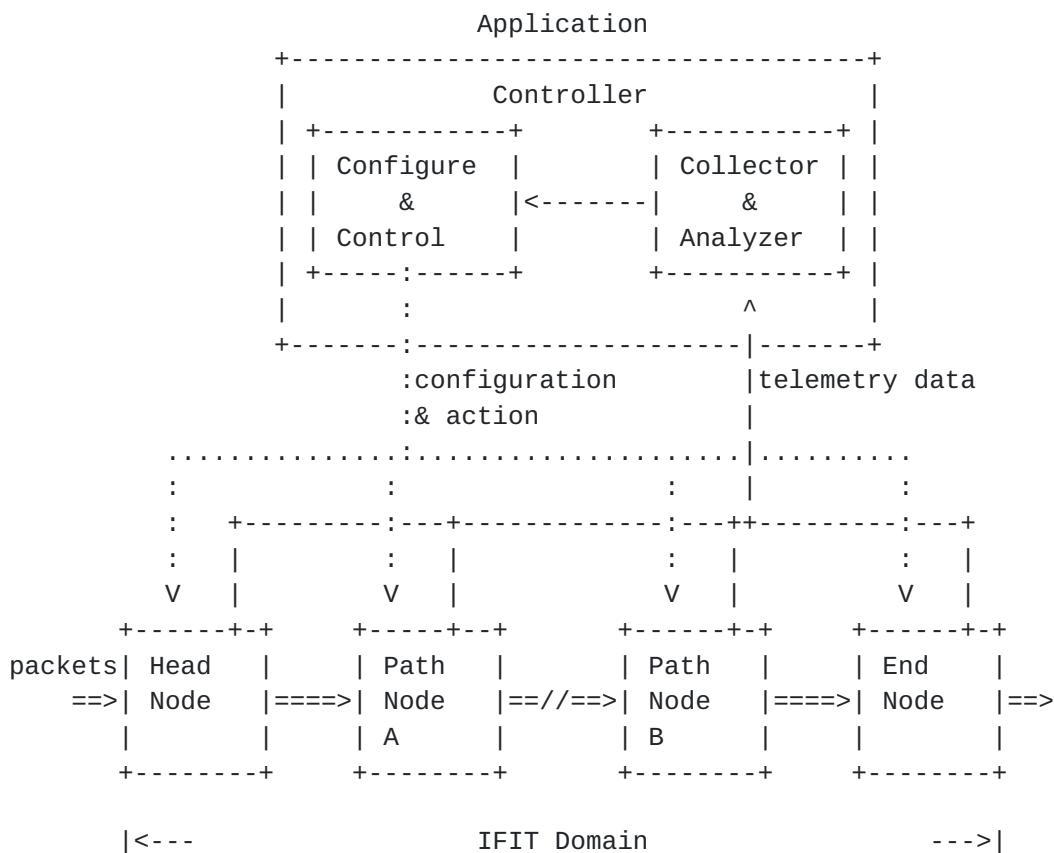
Figure 2: IFIT Deployment Scenario

An on-path telemetry application can conduct some network data plane
monitoring and measurement tasks over an IFIT domain by applying one
or more underlying techniques. The application needs to contains
multiple elements, including configuring the network nodes and
processing the telemetry data. The application usually runs in a
logically centralized controller which is responsible for configuring
the network nodes in the IFIT domain, and collecting and analyzing
telemetry data. The configuration determines which underlying

technique is used, what telemetry data are of interest, which flows
and packets are concerned with, how the telemetry data are collected,
etc. The process can be dynamic and interactive: after the telemetry
data processing and analyzing, the application may instruct the
controller to modify the configuration of the nodes in the IFIT
domain, which affects the future telemetry data collection.

From the system-level view, it is recommended to use the standardized
configuration and data collection interfaces, regardless of the
underlying technique. However, the specification of these interfaces
and the implementation of the controller are out of scope for this
document.

The IFIT domain encompasses the head nodes and the end nodes. An
IFIT domain may cross multiple network domains. The head nodes are
responsible for enabling the IFIT-specific functions and the end
nodes are responsible for terminating them. All capable nodes in an
IFIT domain will be capable of executing the instructed IFIT-specific
function. It is important to note that any IFIT application must,
through configuration and policy, guarantee that any packet with
IFIT-specific header and metadata will not leak out of the IFIT
domain. The end nodes must be able to capture all packets with IFIT-
specific header and metadata and recover their format before
forwarding them out of the IFIT domain.

The underlying on-path telemetry techniques covered by IFIT can be of
any modes discussed in Section 1.1.

2.2. IFIT Architecture

The IFIT architecture is shown in Figure 3, which contains several
key components. These components aim to address the deployment
challenges discussed in Section 1. The detailed block diagram and
description for each component are given in Section 3. Here we only
provide a high level overview.

```
         +------------------------------------+
         |          On-demand Technique       |
         |          Selection & Integration   |
         +------------------------------------+
     Control Plane      |                    ^
    --------------------+------------------+-------------
     Forwarding Plane  V                   |
         +----------------+------------------+
         | Flexible Flow, | Flexible         |
         | Packet, & Data | Data Export      |
         | Selection      |                  |
         +----------------+------------------|
         |      Dynamic Network Probe        |
         +----------------------------------|
```
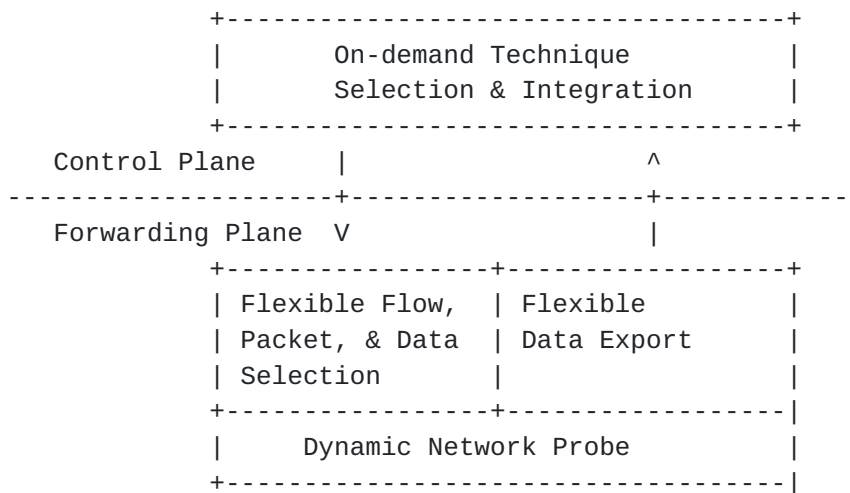
                  Figure 3: IFIT Architecture

   Based on the monitoring and measurement requirements, an application
   needs to choose one or more underlying on-path telemetry techniques
   and decide the policies to apply them. Depending on the forwarding-
   plane protocol and tunneling configuration, the instruction header
   and metadata encapsulation method, if needed, is also determined.
   The encapsulation happens at the head nodes and the decapsulation
   happens at the end nodes.

   Based on the network condition and application requirement, the head
   nodes also need to be able to choose flows and packets to enable the
   IFIT-specific functions, and decide the set of data to be collected.
   All the nodes who are responsible for exporting telemetry data are
   configured with special functions to prepare the data. The IFIT-
   specific functions can be deployed into the network nodes as dynamic
   network probes.

2.3. Relationship with Network Telemetry Framework (NTF)

   [I-D.ietf-opsawg-ntf] describes a Network Telemetry Framework (NTF).
   One dimension used by NTF to partition network telemetry techniques
   and systems is based on the three planes in networks plus external
   data sources. IFIT fits in the category of forwarding-plane
   telemetry and deals with the specific on-path technical branch of the
   forwarding-plane telemetry.

   According to NTF, an on-path telemetry application mainly subscribes
   event-triggered or streaming data. The key functional components of
   IFIT also match the components in NTF. "On-demand Technique
   Selection and Integration" is an application layer function, matching
   the "Data Query, Analysis, and Storage" component in NTF; "Flexible
   Flow, Packet, and Data Selection" matches the "Data Configuration and

Subscription" component; "Flexible Data Export" matches the "Data
Encoding and Export" component; "Dynamic Network Probe" matches the
"Data Generation and Processing" component.

3. Key Components of IFIT

As shown in the IFIT architecture, the key components of IFIT are as
follows:

o Flexible flow, packet, and data selection policy, addressing the
   challenge C1 described in Section 1;

o Flexible data export, addressing the challenge C2;

o Dynamic network probe, addressing C3;

o On-demand technique selection and integration, addressing C4.

Note that the challenges C5 and C6 are mostly standard related, which
are fundamental to IFIT. We discuss the standard status and gaps in
Section 5.

In the following section, we provide a detailed description of each
component.

3.1. Flexible Flow, Packet, 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 impact. Therefore, a workable solution
usually need to select only a subset of flows and flow packets to
enable the data collection, even though this means the loss of some
information and accuracy.

In the data plane, the Access Control List (ACL) provides an ideal
means to determine the subset of flow(s). An application can set a
sample rate or probability to a flow to allow only a subset of flow
packets to be monitored, collect a different set of data for
different packets, and disable or enable data collection on any
specific network node. An application can further allow any node to
accept or deny the data collection process in full or partially.

Based on these flexible mechanisms, IFIT allows applications to apply
flexible 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.

3.1.1. Block Diagram

```
        +----------------------------+
        | +----------+  +----------+ |
        | |Flow      |  |Data      | |
        | |Selection |  |Selection | |
        | +----------+  +----------+ |
        | +----------+               |
        | |Packet    |               |
        | |Selection |               |
        | +----------+               |
        +----------------------------+
```
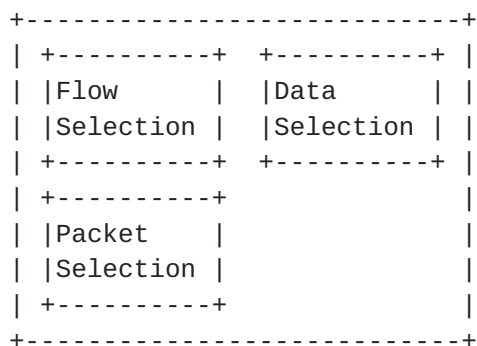
             Figure 4: Flexible Flow, Packet, and Data Selection

   Figure 4 shows the block diagram of this component. The flow
   selection block defines the policies to choose target flows for
   monitoring. Flow has different granularity. A basic flow is defined
   by 5-tuple IP header fields. Flow can also be aggregated at
   interface level, tunnel level, protocol level, and so on. The packet
   selection block defines the policies to choose packets from a target
   flow. The policy can be either a sampling interval, a sampling
   probability, or some specific packet signature. The data selection
   block defines the set of data to be collected. This can be changed
   on a per-packet or per-flow basis.

3.1.2. Example: Sketch-guided Elephant Flow Selection

   Network operators are usually more interested in elephant flows which
   consume more resource and are sensitive to changes in network
   conditions. A CountMin Sketch [CMSketch] can be used 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.

3.1.3. Example: Adaptive Packet Sampling

   Applying on-path telemetry on all packets of selected flows can still
   be out of reach. A sample rate should be set 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.

   An adaptive approach can be used 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 may use 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. An AIMD policy similar to the TCP flow
control mechanism for the rate adjustment can be used.

3.2. Flexible 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 efficient export data encoding (e.g., IPFIX [RFC7011]
or protobuf [1]), nodes have several other ways to reduce the export
data by taking advantage of network device's capability and
programmability. Nodes can 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.

Such events can be expressed as policies. An policy can request data
export only on change, on exception, on timeout, or on threshold.

3.2.1. Block Diagram

```
+--------------------------------------------+
| +-----------+ +-----------+ +-----------+ |
| |Data       | |Data       | |Export     | |
| |Encoding   | |Batching   | |Protocol   | |
| +-----------+ +-----------+ +-----------+ |
| +-----------+ +-----------+ +-----------+ |
| |Data       | |Data       | |Data       | |
| |Compression| |Dedup.     | |Filter     | |
| +-----------+ +-----------+ +-----------+ |
| +-----------+ +-----------+               |
| |Data       | |Data       |               |
| |Computing  | |Aggregation|               |
| +-----------+ +-----------+               |
+--------------------------------------------+
```
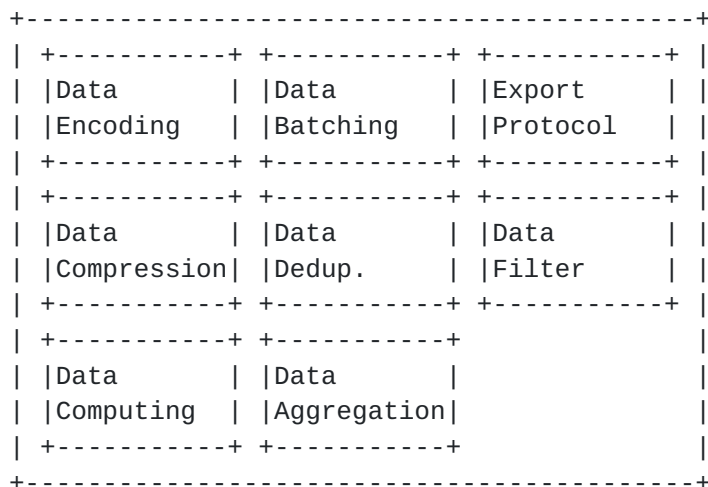
Figure 5: Flexible Data Export

Figure 5 shows the block diagram of this component. The data
encoding block defines the method to encode the telemetry data. The
data batching block defines the size of batch data buffered at the
device side before export. The export protocol block defines the
protocol used for telemetry data export. The data compression block
defines the algorithm to compress the raw data. The data
deduplication block defines the algorithm to remove the redundancy in
the raw data. The data filter block defines the policies to filter
the needed data. The data computing block defines the policies to
prepocess the raw data and generate some new data. The data
aggregation block defines the procedure to combine and synthesize the
data.

3.2.2. Example: Event-based 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). Such
anomalies can be described as events and programmed 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.

The export data reduction due to such optimization is substantial.
For example, given a single 5-hop 10Gbps path, assume a moderate
number of 1 million packets per second are monitored, and the
telemetry data plus the export packet overhead consume less than 30
bytes per hop. Without such optimization, the bandwidth consumed by
the telemetry data can easily exceed 1Gbps (more than 10% of the path

bandwidth), When the optimization is used, the bandwidth consumed by
the telemetry data is negligible. Moreover, the pre-processed
telemetry data greatly simplify the work of data analyzers.

3.3. Dynamic Network Probe

Due to limited data plane resource and network bandwidth, it is
unlikely one can monitor 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). 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.

3.3.1. Block Diagram

```
     +----------------------------+
     | +----------+  +----------+ |
     | |ACL       |  |YANG      | |
     | |          |  |Model     | |
     | +----------+  +----------+ |
     | +----------+  +----------+ |
     | |Hardware  |  |Software  | |
     | |Function  |  |Function  | |
     | +----------+  +----------+ |
     +----------------------------+
```
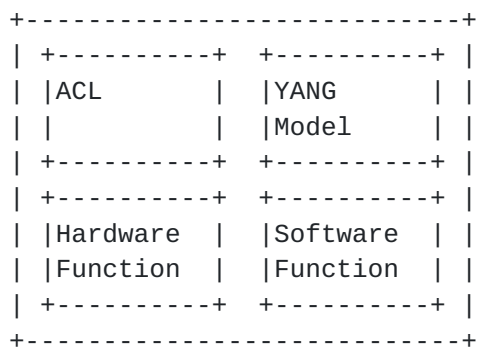
Figure 6: Dynamic Network Probes

Figure 6 shows the block diagram of this component. The Access
Control List (ACL) block is available in most hardware and it defines
DNPs through dynamically update the ACL policies (including flow
filtering and action). YANG models can be dynamically deployed to
enable different data processing and filtering functions. Some
hardware allows dynamically loading hardware-based functions into the
forwarding path at runtime through mechanisms such as reserved

pipelines and function stubs. Dynamically loadable software
functions can be implemented in the control processors in IFIT nodes.

3.3.2. Examples

Following are some possible DNPs that can be dynamically deployed to
support applications.

On-demand Flow Sketch: A flow sketch is a compact online data
   structure (usually a variation of multi-hashing table) for
   approximate estimation of multiple flow properties. It can be
   used to facilitate flow selection. The aforementioned CountMin
   Sketch [CMSketch] is such an example. Since a sketch consumes
   data plane resources, it should only be deployed when actually
   needed.

Smart Flow Filter: The policies that choose flows and packet
   sampling rate can change during the lifetime of an application.

Smart Statistics: An application may need to count flows based on
   different flow granularity or maintain hit counters for selected
   flow table entries.

Smart Data Reduction: DNP can be used to program the events that
   conditionally trigger data export.

3.4. 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, switching
from IOAM to PBT should be supported.

IFIT can further integrate multiple data plane monitoring and
measurement techniques together and present a comprehensive data
plane telemetry solution.

Based on the application requirements and the real-time telemetry
data analysis results, new configurations and actions can be
deployed.

3.4.1. Block Diagram

```
        +-------------------------------------------------+
        | +-----------+  +-------------+  +---------+ |
        | |Application |  |Configuration|  |Telemetry| |
        | |Requirements|->|& Action     |<-|Data     | |
        | |            |  |             |  |Analysis | |
        | +-----------+  +-------------+  +---------+ |
        +-------------------------------------------------+
        | Passport Mode:                                  |
        | +----------+   +----------+   +----------+   |
        | |IOAM E2E  |   |IOAM Trace|   |EAM       |   |
        | +----------+   +----------+   +----------+   |
        | Postcard Mode:                                  |
        | +----------+   +----------+                     |
        | |PBT-M     |   |IOAM DEX  |                     |
        | +----------+   +----------+                     |
        | Hybrid Mode:                                    |
        | +----------+   +----------+                     |
        | |HTS       |   |Multicast |                     |
        | |          |   |Telemetry |                     |
        | +----------+   +----------+                     |
        +-------------------------------------------------+
```
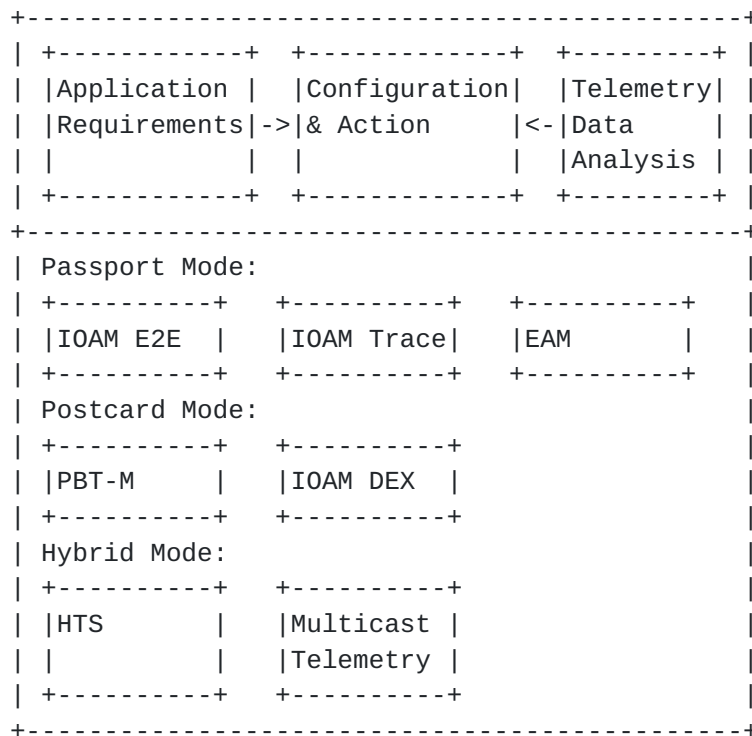
Figure 7: Technique Selection and Integration

   Figure 7 shows the block diagram of this component, which lists the
   candidate on-path telemetry techniques.

   Located in the logically centralized controller of an IFIT domain,
   this component makes all the control and configuration dynamically to
   the capable nodes in the domain which will affect the future
   telemetry data. The configuration and action decisions are based on
   the inputs from the application requirements and the realtime
   telemetry data analysis results. Note that here the telemetry data
   source is not limited to the data plane. The data can come form all
   the sources mentioned in [I-D.ietf-opsawg-ntf], including external
   data sources.

4. IFIT for Reflective Telemetry

   The IFIT components can work together to support reflective
   telemetry, as shown in Figure 8.

```
                         +--------------------+
                         |                    |
                +------+     Applications      |<------+
                |      |                    |          |
                |      +--------------------+          |
                |          Technique Selection         |
                |            and Integration           |
                |                                       |
                |Flexible                     Flexible  |
                |Flow,      reflection-loop      Data   |
                |Packet,                        Export  |
                |and Data                               |
                |Selection                    +----+----+
                V                              +---------+|
          +----------+ Encapsulation           +---------+||
          |  Head    | and Tunneling           |  Path   |||
          |  Node    |----------------------->|  Nodes  ||+
          |          |                         |         |+
          +----------+                         +---------+
             DNP                                   DNP
```
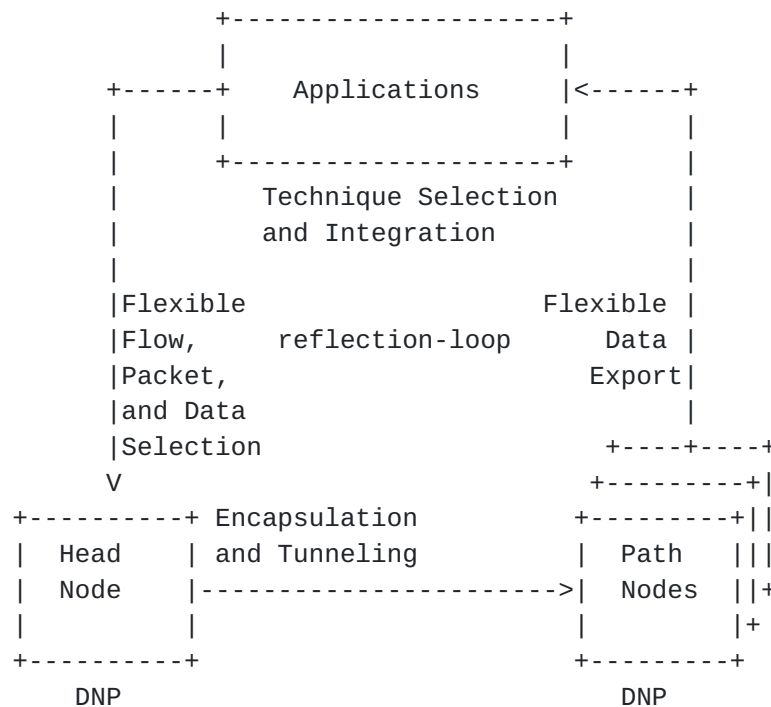
               Figure 8: IFIT-based Reflective Telemetry

   An application may pick a suite of telemetry techniques based on its
   requirements and apply an initial technique to the data plane. It
   then configures the 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 application can change the telemetry technique,
   the flow/data selection policy, and the data export approach in real
   time without breaking the normal network operation. Many of such
   dynamic changes can be done through loading and unloading DNPs.

   The reflective telemetry enabled by the IFIT allows numerous new
   applications suitable for future network operation architecture.

4.1. Example: Intelligent Multipoint Performance Monitoring

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

An application on top of the controllers can manage such mechanism
and IFIT's architecture allows its dynamic and reflective operation.

4.2. Example: Intent-based Network Monitoring

```
                      User Intents
                           |
                           V           Per-packet
                   +------------+      Telemetry
              ACL  |            |      Data
            +--------+ Controller |<--------+
            |       |            |         |
            |       +--+---------+         |
            |          |       ^           |
            |          |DNPs   |Network    |
            |          |       |Information|
            |          V       |           |
      +------+-------------------+-----------+---+
      |     |                                    |
      |     V                     +------+       |
      | +-------+                 +------+|       |
      | | Head  |    IFIT Domain  +------+||      |
      | | Node  |                 |Path  ||+      |
      | |       |                 |Nodes |+       |
      | +-------+                 +------+        |
      +----------------------------------------+
```
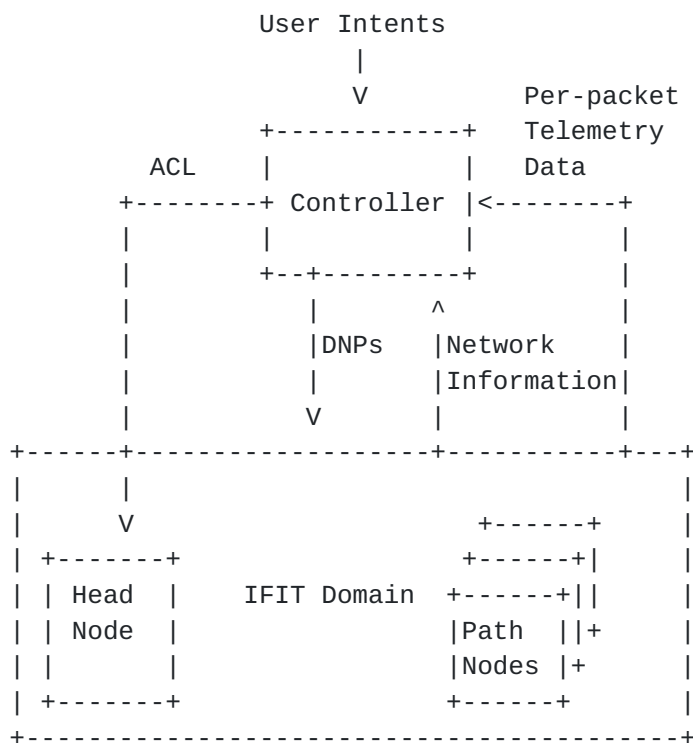
Figure 9: Intent-based Monitoring

In this example, a user can express high level intents for network
monitoring. The controller translates an intent and configure the
corresponding DNPs in IFIT-capable nodes which collect necessary
network information. Based on the real-time information feedback,
the controller runs a local algorithm to determine the suspicious
flows. It then deploys ACLs to the head node to initiate the high
precision per-packet on-path telemetry for these flows.

[5](#). Standard Status and Gaps

A complete IFIT-based solution needs standard interfaces for
configuration and data extraction, and standard encapsulation on
various transport protocols. It may also need standard API and
primitives for application programming and deployment. The draft
[I-D.brockners-opsawg-ioam-deployment] summarizes some current
proposals on encapsulation and data export for IOAM. These works
should be extended or modified to support other types of on-path
telemetry techniques and other transport protocols. The high-level
IFIT helps to develop coherent and universal standard encapsulation
and data export approaches.

[5.1](#). Encapsulation in Transport Protocols

Since the introduction of IOAM, the IOAM option header encapsulation
schemes in various network protocols have been proposed. Similar
encapsulation schemes need to be extended to cover the other on-path
telemetry techniques. On the other hand, the encapsulation scheme
for some popular protocols, such as MPLS and IPv4, are noticeably
missing. It is important to provide the encapsulation schemes for
these protocols because they are still prevalent in carrier networks.
IFIT needs to provide 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. While there are some proposals which allow new
header encapsulation in MPLS packets (e.g.,
[I-D.song-mpls-extension-header]) or in IPv4 packets (e.g.,
[I-D.herbert-ipv4-eh]), they are still in their infancy stage and
require significant future work. For the meantime, in a confined
IFIT domain, pre-standard encapsulation approaches may be applied.

[5.2](#). Tunneling Support

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.

[5.3](#). Deployment Automation

In addition, standard approaches that automates the function
configuration, and capability query and advertisement, either in a
centralized fashion or a distributed fashion, are still immature.

The draft [I-D.zhou-ippm-ioam-yang] provides the YANG model for IOAM
configuration. Similar models needs to be defined for other
techniques. It is also helpful to provide standards-based approaches
for configuration in various network environments. For example, in
segment routing networks, extensions to BGP or PCEP can be defined to
distribute SR policies carrying IFIT information, so that IFIT
behavior can be enabled automatically when the SR policy is applied.
[I-D.chen-pce-sr-policy-ifit] proposes to extend PCEP policy for IFIT
configuration in segment routing networks.
[I-D.qin-idr-sr-policy-ifit] proposes to extend BGP policy instead
for IFIT configuration in segment routing networks. Additional
capability discovery and dissemination will be needed for other types
of networks.

To realize the potential of IFIT, programming and deploying DNPs are
important. ForCES [RFC5810] is a standard protocol for network
device programming, which can be used for DNP deployment. Currently
some related works such as [I-D.wwx-netmod-event-yang] and
[I-D.bwd-netmod-eca-framework] have proposed to use YANG model to
define the smart policies which can be used to implement DNPs. In
the future, other approaches for hardware and software-based
functions can be development to enhance the programmability and
flexibility.

6. Summary

IFIT is a high-level framework for applying on-path telemetry
techniques, and this document has outlined how the framework may be
used to solve essential use cases. IFIT enables a practical data-
plane telemetry application based on two basic on-path traffic data
collection modes: passport and postcard.

IFIT addresses the key challenges for operators to deploy a complete
on-path telemetry solution. However, as a reference and open
framework, IFIT only describes the basic functions of each identified
component and suggests possible applications. It has no intention of
specifying the implementation of the components and the interfaces
between the components. The compliance of IFIT is by no means
mandatory either. Instead, this informational document aims to
clarify the problem domain, and summarize the best practices and
sensible system design considerations. IFIT can guide the analysis
of the current standard status and gaps, and motivate new works to
complete the ecosystem. IFIT enables data-plane reflective telemetry
applications for advanced network operations.

Having a high-level framework covering a class of related techniques
also promotes a holistic approach for standard development and helps
to avoid duplicated efforts and piecemeal solutions that only focus

on a specific technique while omitting the compatibility and
extensibility issues, which is important to a healthy ecosystem for
network telemetry.

7. Security Considerations

In addition to the specific security issues discussed in each
individual document on on-path telemetry, this document considers the
overall security issues at the IFIT system level. This should serve
as a guide to the on-path telemetry application developers and users.

8. IANA Considerations

This document includes no request to IANA.

9. Contributors

Other major contributors of this document include Giuseppe Fioccola,
Daniel King, Zhenqiang Li, Zhenbin Li, Tianran Zhou, and James
Guichard.

10. Acknowledgments

We thank Diego Lopez, Shwetha Bhandari, Joe Clarke, Adrian Farrel,
Frank Brockners, Al Morton, Alex Clemm, Alan DeKok, and Warren Kumari
for their constructive suggestions for improving this document.

11. References

11.1. Normative References

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", BCP 14, RFC 2119,
            DOI 10.17487/RFC2119, March 1997,
            <https://www.rfc-editor.org/info/rfc2119>.

[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>.

11.2. 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-opsawg-ioam-deployment]
              Brockners, F., Bhandari, S., and d.
              daniel.bernier@bell.ca, "In-situ OAM Deployment", draft-
              brockners-opsawg-ioam-deployment-02 (work in progress),
              September 2020.

   [I-D.bwd-netmod-eca-framework]
              Boucadair, M., WU, Q., Wang, Z., King, D., and C. Xie,
              "Framework for Use of ECA (Event Condition Action) in
              Network Self Management", draft-bwd-netmod-eca-
              framework-00 (work in progress), November 2019.

   [I-D.chen-pce-sr-policy-ifit]
              Chen, H., Yuan, H., Zhou, T., Li, W., Fioccola, G., and Y.
              Wang, "PCEP SR Policy Extensions to Enable IFIT", draft-
              chen-pce-sr-policy-ifit-02 (work in progress), July 2020.

   [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-ioam-data]
              Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields
              for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in
              progress), November 2020.

   [I-D.ietf-ippm-ioam-direct-export]
              Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F.,
              Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ
              OAM Direct Exporting", draft-ietf-ippm-ioam-direct-
              export-02 (work in progress), November 2020.

   [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-09 (work in progress), March 2020.

   [I-D.ietf-opsawg-ntf]
              Song, H., Qin, F., Martinez-Julia, P., Ciavaglia, L., and
              A. Wang, "Network Telemetry Framework", draft-ietf-opsawg-
              ntf-06 (work in progress), January 2021.

   [I-D.mirsky-ippm-hybrid-two-step]
              Mirsky, G., Lingqiang, W., Zhui, G., and H. Song, "Hybrid
              Two-Step Performance Measurement Method", draft-mirsky-
              ippm-hybrid-two-step-07 (work in progress), December 2020.

   [I-D.qin-idr-sr-policy-ifit]
              Qin, F., Yuan, H., Zhou, T., Fioccola, G., and Y. Wang,
              "BGP SR Policy Extensions to Enable IFIT", draft-qin-idr-
              sr-policy-ifit-04 (work in progress), October 2020.

   [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., Mirsky, G., Shin, J., and K.
              Lee, "Postcard-based On-Path Flow Data Telemetry using
              Packet Marking", draft-song-ippm-postcard-based-
              telemetry-08 (work in progress), October 2020.

   [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-multicast-telemetry]
              Song, H., McBride, M., Mirsky, G., and G. Mishra,
              "Multicast On-path Telemetry Solutions", draft-song-
              multicast-telemetry-07 (work in progress), January 2021.

   [I-D.wwx-netmod-event-yang]
              WU, Q., Bryskin, I., Birkholz, H., Liu, X., and B. Claise,
              "A YANG Data model for ECA Policy Management", draft-wwx-
              netmod-event-yang-10 (work in progress), November 2020.

   [I-D.zhou-ippm-enhanced-alternate-marking]
              Zhou, T., Fioccola, G., Lee, S., Cociglio, M., and W. Li,
              "Enhanced Alternate Marking Method", draft-zhou-ippm-
              enhanced-alternate-marking-06 (work in progress), January
              2021.

   [I-D.zhou-ippm-ioam-yang]
              Zhou, T., Guichard, J., Brockners, F., and S. Raghavan, "A
              YANG Data Model for In-Situ OAM", draft-zhou-ippm-ioam-
              yang-08 (work in progress), July 2020.

   [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>.

   [RFC5810] Doria, A., Ed., Hadi Salim, J., Ed., Haas, R., Ed.,
              Khosravi, H., Ed., Wang, W., Ed., Dong, L., Gopal, R., and
              J. Halpern, "Forwarding and Control Element Separation
              (ForCES) Protocol Specification", RFC 5810,
              DOI 10.17487/RFC5810, March 2010,
              <https://www.rfc-editor.org/info/rfc5810>.

   [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>.

11.3. URIs

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

Authors' Addresses

   Haoyu Song
   Futurewei
   2330 Central Expressway
   Santa Clara
   USA

   Email: haoyu.song@futurewei.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

Jaehwan Jin
LG U+
South Korea

Email: daenamu1@lguplus.co.kr

Jongyoon Shin
SK Telecom
South Korea

Email: jongyoon.shin@sk.com