In-situ Flow Information Telemetry
draft-song-opsawg-ifit-framework-10

Abstract

As networks increase in scale and network operations become more sophisticated, traditional Operation, Administration and Maintenance (OAM) methods, which include proactive and reactive techniques, running in active and passive modes, become more susceptible to measurement accuracy and misconfiguration errors. With the advent of programmable data-plane, emerging on-path telemetry techniques provide unprecedented flow insight and realtime notification of network issues.

This document enumerates the key deployment challenges for flow-oriented on-path telemetry techniques, especially in carrier operator networks. To address these issues, a high-level framework, In-situ Flow Information Telemetry (iFIT), is outlined. iFIT includes several essential functional components that can be materialized and assembled to implement a complete solution for on-path telemetry.

This informational document aims to clarify the problem domain, and summarize the best practices and sensible system design considerations. The iFIT framework helps to guide the analysis on the current standard status and gaps, and motivate new works to complete the ecosystem. It also helps to inspire innovative network telemetry applications supporting advanced network operations. As a reference and open framework, iFIT does not specify the implementation of the components and the interfaces between the components. The compliance with iFIT framework is not mandatory for telemetry applications either.
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 July 3, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ............................................. 3
   1.1. Requirements and Challenges .......................... 4
   1.2. Scope .................................................. 6
   1.3. Glossary ................................................ 6
   1.4. Requirements Language ................................... 7
2. iFIT Framework Overview ................................... 7
   2.1. iFIT Network Architecture .............................. 8
       2.1.1. On-path Telemetry Models: Passport vs. Postcard ... 9
   2.2. iFIT Framework Architecture ............................ 10
   2.3. Relationship with Network Telemetry Framework (NTF) ... 11
3. Key Components of iFIT ................................. 11
   3.1. Smart Flow, Packet, and Data Selection ............... 11
       3.1.1. Block Diagram ................................... 12
       3.1.2. Example: Sketch-guided Elephant Flow Selection ... 12
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 in active and passive modes, become more susceptible to measurement accuracy and misconfiguration errors, as networks increase in scale and network operations become more sophisticated.

The sheer complexity of today’s networks and stringent service requirements require new traffic monitoring and measurement solutions for a wide range of use cases with high performance and high precision. Furthermore, the ability to expedite failure detection, fault localization, and recovery mechanisms, particularly in the case of soft failures or path degradation are expected, without causing service disruption.

Future networks also need to be application-aware. 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 can optimize the network resource usage and monitoring to ensure application and traffic optimality.

With the advent of programmable data-plane, emerging on-path telemetry techniques provide unprecedented flow insight and realtime notification of network issues (e.g., jitter, increased 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 network operations that need user SLA compliance, service path enforcement, fault diagnosis, and network resource optimization. A family of on-path telemetry techniques, including In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data], 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. These on-path telemetry techniques are very different from the previous active and passive OAM schemes in that they directly modify the user packets and can guarantee 100% accuracy. 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 invaluable 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 strong interest in utilizing such techniques for various purposes. For example, it is vital for the operators who offer bandwidth intensive, latency and loss sensitive services such as video streaming and online gaming to closely monitor the relevant flows in real time as the indispensable first step for any further measure.

1.1. Requirements and Challenges

The potential benefits of on-path telemetry are substantial. However, successfully applying such techniques in carrier networks needs to consider performance, deployability, and flexibility. Specifically, we need to address the following practical deployment challenges:

- C1: On-path 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. For example, the growing IOAM data per hop can negatively affect service levels by increasing the serialization delay and header parsing delay.

- **C2**: On-path telemetry can generate a huge 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 will consume almost half of the network bandwidth, not to mention the back-end data handling load.

- **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 will be needed in an on-demand and interactive fashion. Flexibility and extensibility on data defining, aggregation, 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 on-path telemetry technique may lead to 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. The system level orchestration is needed.

- **C6**: The development of simplified on-path telemetry primitives and models for configuration and query is important and necessary. These may be used by an API-based telemetry service for external...
applications, for end-to-end performance measurement of network paths and application performance monitoring.

1.2. 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 technique, and provides a high level application framework which addresses the aforementioned challenges for deployment especially in carrier operator networks.

This document aims to clarify the problem domain, and summarize the best practices and sensible system design considerations. The framework helps to guide the analysis on the current standard status and gaps, and motivate new works to complete the ecosystem. It also helps to inspire innovative network telemetry applications supporting advanced network operations.

As an informational document, it describes an open framework with a few key components. The framework does not enforce 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. The compliance of the reference framework is not mandatory either.

The standardization of the underlying techniques and interfaces 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.

1.3. Glossary

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

On-path Telemetry: Remotely acquiring performance and behavior data about a network flow 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".

iFIT Framework: A high-level reference framework that supports network data-plane monitoring applications which apply one or more
of the underlying on-path telemetry techniques and materialize the
iFIT functional components for practical deployment. iFIT
framework is dedicated for flow-oriented data plane telemetry.

iFIT Application: A network monitoring application that conforms to
the iFIT framework.

iFIT Domain: A network domain in which an iFIT 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.

iFIT Node: A network node, usually a forwarding device, that is in
an iFIT domain and is capable of iFIT-specific functions.

iFIT Head Node: A special iFIT node. It is the entry node to an
iFIT domain. Usually the instruction header encapsulation, if
needed, happens here.

iFIT End Node: A special iFIT node. It is the exit node of an iFIT
domain. Usually the instruction header decapsulation, if needed,
happens here.

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

1.4. 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 Framework Overview

To address the aforementioned challenges, 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 solution. 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/
application traffic flows. As a reference framework for building a
complete solution, iFIT covers a class of on-path telemetry
techniques and works a level higher than any specific underlying technique. The framework is built up on a few key functional components (Section 3). By assembling these components, iFIT supports reflective telemetry that enables autonomous network operations (Section 4).

2.1. iFIT Network Architecture

The network architecture that applies iFIT is shown in Figure 1.

An iFIT application conducts some network data plane monitoring and measurement tasks over an iFIT domain through applying one or more underlying on-path telemetry techniques. 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 iFIT application may instruct the controller to modify the iFIT node configuration which affects the future telemetry data collection.

From 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 is confined between the iFIT head nodes and the iFIT end nodes. An iFIT domain may cross multiple network domains. The iFIT head nodes are responsible for enabling the iFIT-specific functions and the iFIT end nodes are responsible for annulling them. All active iFIT nodes in an iFIT domain will then execute the instructed iFIT-specific function. Any iFIT application must guarantee that any packet with iFIT-specific header and metadata will not leak out from the iFIT domain. The iFIT 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.

iFIT supports two basic on-path telemetry modes: passport mode (e.g., IOAM trace option), in which telemetry data are carried in user packets and only 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 dedicated packets. An on-path telemetry application may need to mix or switch between the two modes.

2.1.1. On-path Telemetry Models: 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. These help to simplify the data collector and analyzer’s work. 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 effort to correlate the postcard packets.

2.2. iFIT Framework Architecture

The iFIT framework architecture is shown in Figure 2, 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                   |
+-----------------+------------------+
| Smart Flow,     | Smart Data       |
| Packet, & Data  | Export           |
| Selection       |                  |
+-----------------+------------------+
| Dynamic Network Probe          |
+-----------------------------+
| Encapsulation & Tunneling     |
+-----------------------------+
```

Figure 2: iFIT Framework Architecture

Based on the monitoring and measurement requirements, an iFIT 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 iFIT head nodes and the decapsulation happens at the iFIT end nodes.

Based on the network condition and application requirement, the iFIT 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 iFIT nodes who are responsible for exporting telemetry data are configured with special functions to prepare the data. The iFIT-specific functions can be dynamically deployed into the iFIT 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 framework fits in the forwarding-plane telemetry category and deals with the specific on-path technical branch of the forwarding-plane telemetry.

According to NTF, an iFIT application mainly subscribes event-triggered or streaming data. The key functional components of iFIT framework also match the components in NTF. On-demand Technique Selection and Integration is basically an application layer function, matching the Data Query, Analysis, and Storage component in NTF; Smart Flow, Packet, and Data Selection matches the Data Configuration and Subscription component; Smart Data Export matches the Data Encoding and Export component; The other two components match the Data Generation and Processing component.

3. Key Components of iFIT

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

- Smart flow, packet, and data selection policy, addressing the challenge C1 described in Section 1.
- Smart data export, addressing the challenge C2.
- Dynamic network probe, addressing C3.
- Encapsulation and tunneling, addressing C4.
- On-demand technique selection and integration, addressing C5.

Note that this document does not directly address the challenge C6 which is open for future standard proposals and left as the concern of application implementers.

Next we provide a detailed description of each component.

3.1. Smart 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 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.

3.1.1. Block Diagram

```
+----------------------------+
| +----------+  +----------+ |
| |Flow      |  |Data      | |
| |Selection |  |Selection | |
| +----------+  +----------+ |
| +----------+               |
| |Packet    |               |
| |Selection |               |
| +----------+               |
+----------------------------+
```

Figure 3: Samrt Flow, Packet, and Data Selection

Figure 3 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. 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 efficient export data encoding (e.g., IPFIX [RFC7011] or protobuf [1]), iFIT nodes have several other ways to reduce the export data by taking advantage of network device’s capability and programmability. iFIT 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

![Figure 4: Smart Data Export](image)

Figure 4 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 preprocess 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 (>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 5: Dynamic Network Probes
Figure 5 shows the block diagram of this component. The ACL block is available in most hardware and it defines DNFs 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 DNFs that can be dynamically deployed to support iFIT applications.

On-demand Flow Sketch: A flow sketch is a compact online data structure for approximate flow statistics which can be used to facilitate flow selection. The aforementioned CountMin Sketch is such an example. Since a sketch consumes data plane resources, it should only be deployed when 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 interactively 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. Encapsulation and Tunneling

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.

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.

3.4.1. Block Diagram

```
+----------------------------+
| +----------+  +----------+ |
| |Uniform   |  |Pipe      | |
| |Tunnel    |  |Tunnel    | |
| +----------+  +----------+ |
| +------+ +------+ +------+ |
| |IPv6  | |SRv6  | |MPLS  | |
| +------+ +------+ +------+ |
| +------+ +------+ +------+ |
| |IPv4  | |Ether.| |Others| |
| +------+ +------+ +------+ |
+----------------------------+
```

Figure 6: Tunnel Mode and Encapsulation Scheme

Figure 6 shows the block diagram of this component, which lists two tunnel modes supported and various protocols with each needing an iFIT-specific header encapsulation solution.

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

3.5.1. Block Diagram

Figure 7: Technique Selection and Integration

Figure 7 shows the block diagram of this component, which lists the candidate on-path telemetry techniques. IOAM E2E and Trace options are described in [I-D.ietf-ippm-ioam-data]. EAM is described in [I-D.zhou-ippm-enhanced-alternate-marking]. PBT-M is described in [I-D.song-ippm-postcard-based-telemetry]. IOAM DEX option is described in [I-D.ioamteam-ippm-ioam-direct-export]. HTS is described in [I-D.mirsky-ippm-hybrid-two-step]. Multicast Telemetry is described in [I-D.song-multicast-telemetry].

Located in the logically centralized controller of an iFIT domain, this component makes all the control and configuration dynamically to the iFIT nodes 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.

An iFIT 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 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 real time without breaking the normal network operation. Many of such dynamic changes can be done through loading and unloading DNs.

The reflective telemetry enabled by the iFIT framework 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 iFIT 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

![Intent-based Monitoring diagram](image)

**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 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 iFIT head node to initiate the high precision per-packet on-path telemetry for these flows.

5. Standard Status and Gaps

A complete iFIT 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 framework helps to develop coherent and universal standard encapsulation and data export approaches.

In addition, standard approaches for function configuration, 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 helpful to provide standard approaches for distributed configuration in various network environments.

To realize the potential of iFIT, programming and deploying DNPs are important. 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 and open framework for applying on-path telemetry techniques. Combining with algorithmic and architectural schemes that fit into the framework components, iFIT 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 categorized as the hybrid OAM type I mode per [RFC7799], which can provide more flexible and accurate network monitoring and measurement.

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 framework 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. The iFIT framework can guide the analysis of the current standard status and gaps, and motivate new works to complete the ecosystem. It also helps to inspire innovative data-plane reflective telemetry applications supporting advanced network operations.

Having a 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. To foster a healthy ecosystem for network telemetry, we consider this essential.

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 iFIT 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 for their constructive suggestions for improving this document.
11. References

11.1. Normative References


11.2. Informative References


[I-D.brockners-opsawg-ioam-deployment] Brockners, F., Bhandari, S., and daniel.bernier@bell.ca, "In-situ OAM Deployment", draft-brockners-opsawg-ioam-deployment-00 (work in progress), October 2019.


[I-D.zhou-ippm-enhanced-alternate-marking]
Zhou, T., Fioccola, G., Li, Z., Lee, S., and M. Cociglio,
"Enhanced Alternate Marking Method", draft-zhou-ippm-
enhanced-alternate-marking-04 (work in progress), October
2019.

[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-04 (work in progress), June 2019.

[pasport-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.

DOI 10.17487/RFC2113, February 1997,

"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,

11.3. URIs


Authors’ Addresses

Haoyu Song (editor)
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: daenamul@lguplus.co.kr

Jongyoon Shin
SK Telecom
South Korea
Email: jongyoon.shin@sk.com