

OPSAWG
Internet-Draft
Intended status: Informational
Expires: May 22, 2020

H. Song, Ed.
Futurewei
F. Qin
China Mobile
H. Chen
China Telecom
J. Jin
LG U+
J. Shin
SK Telecom
November 19, 2019

**In-situ Flow Information Telemetry
draft-song-opsawg-ifu-framework-08**

Abstract

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

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

This document outlines a reference architectural framework, In-situ Flow Information Telemetry (iFIT), which enumerates several high level components and describes how these components can be assembled to achieve a complete and closed-loop working solution for on-path telemetry.

iFIT addresses several deployment challenges for on-path telemetry techniques, especially in carrier networks. As an open framework, it does not detail the implementation of the components as well as the interface between the components. Rather, the document summarizes the current status and analyzes the gaps on related standard developments.

Requirements Language

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

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 22, 2020.

Copyright Notice

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

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

Table of Contents

1.	Requirements and Challenges	3
2.	Glossary	5
3.	iFIT Framework Overview	6
3.1.	Passport vs. Postcard	7
4.	Architectural Components of iFIT	8
4.1.	Smart Flow and Data Selection	8
4.1.1.	Example: Sketch-guided Elephant Flow Selection . . .	9
4.1.2.	Example: Adaptive Packet Sampling	9
4.2.	Smart Data Export	9
4.2.1.	Example: Event-based Anomaly Monitor	10
4.3.	Dynamic Network Probe	10

4.3.1. Examples	11
4.4. Encapsulation and Tunneling	11
4.5. On-demand Technique Selection and Integration	12
5. iFIT Closed-Loop Architecture	12
5.1. Example: Intelligent Multipoint Performance Monitoring .	14
5.2. Example: Intent-based Network Monitoring	14
6. Standard Status and Gap Analysis	15
7. Summary and Future Work	15
8. Security Considerations	16
9. IANA Considerations	16
10. Contributors	16
11. Acknowledgments	16
12. References	16
12.1. Normative References	16
12.2. Informative References	17
12.3. URIs	19
Authors' Addresses	19

1. Requirements and Challenges

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

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

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

Application-aware network operation is important for user SLA compliance, service path enforcement, fault diagnosis, and network resource optimization. A family of on-path flow telemetry

techniques, including In-situ OAM (IOAM) [[I-D.brockners-inband-oam-data](#)], Postcard Based Telemetry (PBT) [[I-D.song-ippm-postcard-based-telemetry](#)], In-band Flow Analyzer (IFA) [[I-D.kumar-ippm-ifa](#)], Enhanced Alternate Marking (EAM) [[I-D.zhou-ippm-enhanced-alternate-marking](#)], and Hybrid Two Steps (HTS) [[I-D.mirsky-ippm-hybrid-two-step](#)], are emerging, which can provide flow information on the entire forwarding path on a per-packet basis in real time. These on-path flow telemetry techniques are very different from the previous active and passive OAM schemes in that they directly modify the user packets. Given the unique characteristics of the aforementioned techniques, we may categorize these on-path telemetry techniques as the hybrid OAM type III, supplementing the classification defined in [[RFC7799](#)].

These techniques are invaluable for application-aware network operations not only in data center and enterprise networks but also in carrier networks which may cross multiple domains. Carrier network operators have shown strong 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 gaming to closely monitor the relevant flows in real time as the indispensable first step for any further measure.

However, successfully applying such techniques in carrier networks needs to consider performance, deployability, and flexibility. Specifically, several practical challenges need to be addressed:

- o C1: On-path flow telemetry incurs extra packet processing which may strain the network data plane. The potential impact on the forwarding performance creates an unfavorable "observer effect" which not only damages the fidelity of the measurement but also defies the purpose of the measurement.
- o C2: On-path flow telemetry can generate a huge amount of OAM data which may claim too much transport bandwidth and inundate the servers for data collection, storage, and analysis. Increasing the data handling capacity is technically viable but expensive. For example, assume IOAM is applied to all the traffic. One node will collect a few tens of bytes as telemetry data for each packet. The whole forwarding path might accumulate a data trace with a size similar to the average size of the original packets. Exporting the telemetry data will consume almost half of the network bandwidth.
- 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, network convergence, and packet-optical integration continue, more data will be needed in an on-demand and interactive fashion. Flexibility and extensibility on data defining, acquisition, and filtering, must be considered.

- o 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.
- o C5: Applying only a single underlying 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 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.
- o C6: Development of simplified on-path telemetry primitives and models, including: telemetry data (e.g., nodes, links, ports, paths, flows, timestamps) query primitives. These may be used by an API-based telemetry service for external applications, for monitoring end-to-end latency measurement of network paths and application latency calculation.

2. Glossary

This section defines and explains some terms used in this document.

On-path Telemetry: Acquiring data about a packets on its forwarding path. The term refers to a class of data plane telemetry techniques which collect data about user flows and packets along their forwarding paths. IOAM, PBT, IFA, EAM, and HTS are all on-path telemetry techniques. Such techniques may need to mark user packets, or insert instruction or data to the headers of user packets.

iFIT: In-situ Flow Information Telemetry

iFIT Framework: A reference framework that supports network OAM applications to apply dataplane on-path telemetry techniques.

iFIT Application: A network OAM application that applies the iFIT framework.

iFIT Domain: The network domain that participates in an iFIT application.

iFIT Node: A network node that is in an iFIT domain and is capable of iFIT-specific functions.

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

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

3. iFIT Framework Overview

To address the aforementioned challenges, we propose an architectural framework based on multiple network operators' requirements and common industry practice, which can help to build a workable on-path flow telemetry solution. We name the framework "In-situ Flow Information Telemetry" (iFIT) to reflect the fact that this framework is dedicated to the on-path telemetry data about user/application flow experience. As an architectural framework for building a complete solution, iFIT works a level higher than specific data plane OAM techniques, be it active, passive, or hybrid. The framework is built up on a few high level architectural components ([Section 4](#)). By assembling these components, a closed-loop can be formed to provide a complete solution for static, dynamic, and interactive telemetry applications ([Section 5](#)).

iFIT is an open framework. It 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 network architecture that applies iFIT is shown in Figure 1. The iFIT domain is confined between the iFIT head nodes and the iFIT end nodes. An iFIT domain may cross multiple network domains. An iFIT application uses a controller to configure all the iFIT nodes. The configuration determines what telemetry data are collected. After the telemetry data processing and analyzing, the iFIT application may instruct the controller to modify the iFIT node configuration and affect the future telemetry data collection. How applications communicate with the controller is out of scope for this document

iFIT supports two basic on-path telemetry data collection modes: passport mode (e.g., IOAM trace option and IFA), in which telemetry data are carried in user packets and exported at the iFIT end nodes, and postcard mode (e.g., PBT), in which each node in the iFIT domain

may export telemetry data through independent OAM packets. Note that the boundary between the two modes can be blurry. An application only need to mix the two modes.

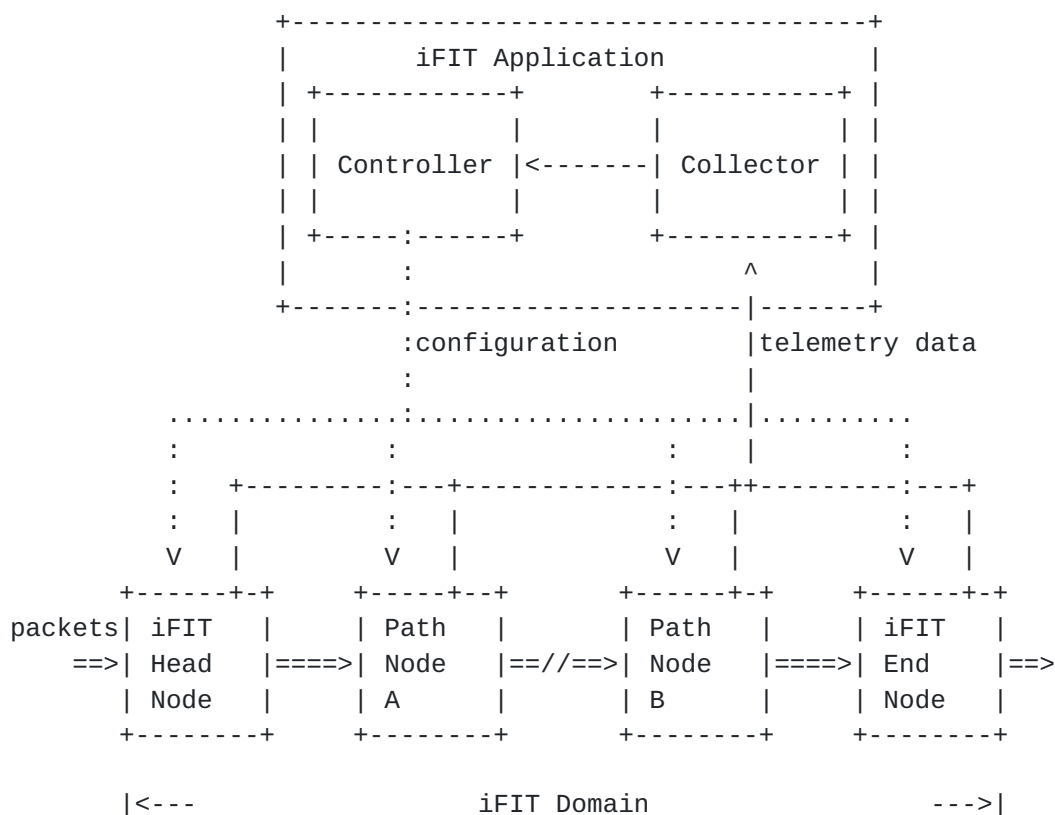


Figure 1: iFIT Network Architecture

3.1. Passport vs. Postcard

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

A prominent advantage of the passport mode is that it naturally retains the telemetry data correlation along the entire path. The passport mode also reduces the number of data export packets and the bandwidth consumed by the data export packets. These can help to

make the data collector and analyzer's work easier. On the other hand, the passport mode requires more processing on the user packets and increases the size of user packets, which can cause various problems. Some other issues are documented in [\[I-D.song-ippm-postcard-based-telemetry\]](#).

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

4. Architectural Components of iFIT

The high level components of iFIT are listed as follows:

- o Smart flow and data selection policy to address the challenge C1 described in [Section 1](#).
- o Smart data export to address the challenge C2.
- o Dynamic network probe to address C3.
- o Encapsulation and tunneling to address C4.
- o On-demand technique selection and integration to address C5.

Note that this document does not directly address the challenge C6 which is left to be a concern for iFIT application implementers.

Next we provide a detailed description of each component.

4.1. Smart Flow and Data Selection

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

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

Based on these flexible mechanisms, iFIT allows applications to apply smart flow and data selection policies to suit the requirements. The applications can dynamically change the policies at any time based on the network load, processing capability, focus of interest, and any other criteria.

4.1.1. 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.

4.1.2. 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.

4.2. 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.

4.2.1. 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.

4.3. 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) [[I-D.song-opsawg-dnp4iq](#)]. DNP is the technique to enable probes for customized data collection in different network planes. When working with IOAM or PBT, DNP is loaded to the data plane through incremental programming or configuration. The DNP can effectively conduct data generation, processing, and aggregation.

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

[4.3.1.](#) Examples

Following are some possible DNPs 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.

[4.4.](#) Encapsulation and Tunneling

Since the introduction of IOAM, the IOAM option header encapsulation schemes in various network protocols have been proposed with the omission of some protocols, such as MPLS and IPv4, which are still prevalent in carrier networks. iFIT provides solutions to apply the on-path flow telemetry techniques in such networks. PBT-M [[I-D.song-ippm-postcard-based-telemetry](#)] does not introduce new headers to the packets so the trouble of encapsulation for a new header is avoided. In case a technique that requires a new header is preferred, [[I-D.song-mpls-extension-header](#)] provides a means to encapsulate the extra header using an MPLS extension header. As for IPv4, it is possible to encapsulate the new header in an IP option. For example, RAO [[RFC2113](#)] can be used to indicate the presence of

the new header. A recent proposal [[I-D.herbert-ipv4-eh](#)] that introduces the IPv4 extension header may lead to a long term solution.

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

[4.5.](#) 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.

[5.](#) iFIT Closed-Loop Architecture

The iFIT architectural components can work together to form closed-loop applications, as shown in Figure 2.

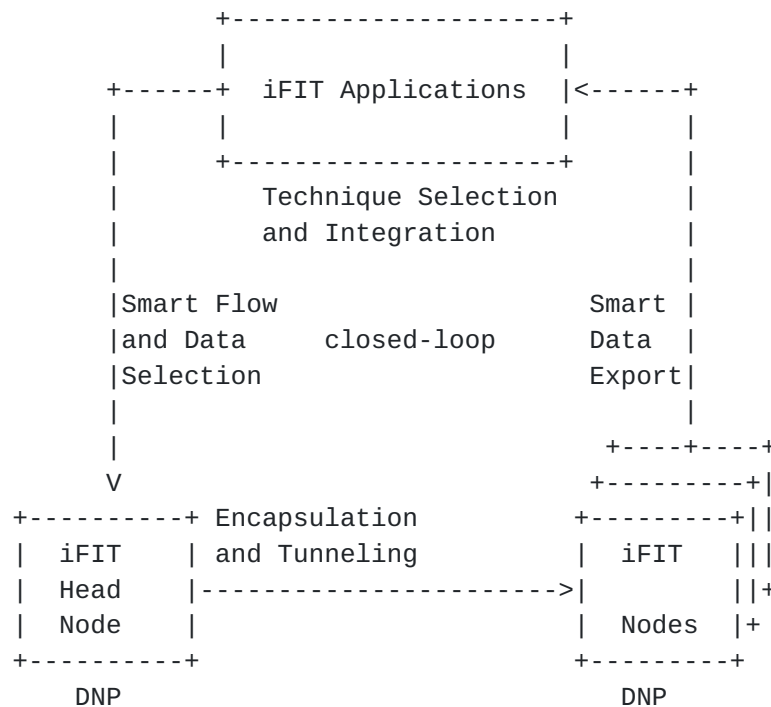


Figure 2: iFIT Closed-Loop Architecture

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 DNPs.

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

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

5.1. 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 the iFIT closed-loop architecture allows its dynamic and flexible operation.

5.2. Example: Intent-based Network Monitoring

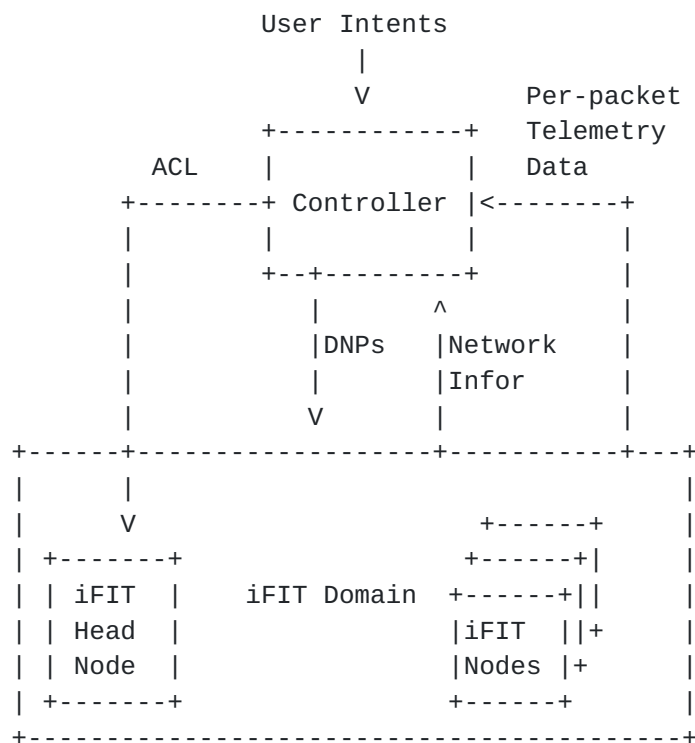


Figure 3: Intent-based Monitoring

In this example, a user can express high level intents for network monitoring. The controller translates an intent and configure the corresponding DNP in iFIT nodes which collect necessary network information. Based on the realtime 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.

6. Standard Status and Gap Analysis

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. However, these works can 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 DNP are important. Currently some related works such as [\[I-D.wwx-netmod-event-yang\]](#) have proposed to use YANG model to define the smart policies which can be used to implement DNP. In the future, other approaches can be development to enhance the programmability and flexibility.

7. Summary and Future Work

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 framework enables a practical telemetry solution based on two basic on-path traffic data collection modes: passport and postcard.

The operation of iFIT differs from both active OAM and passive OAM as defined in [\[RFC7799\]](#). It does not generate any active probe packets or passively observe unmodified user packets. Instead, it modifies selected user packets to collect useful information about them.

Therefore, the iFIT operation can be considered the hybrid type III mode, which can provide more flexible and accurate network OAM.

While iFIT addresses several key challenges for operators to deploy a complete on-path telemetry system, it does not provide detailed description or constraints on component implementation and interface specification. As a high level framework, iFIT can help to identify the standard gaps and innovation opportunities, and promote the development of a complete ecosystem.

More challenges and corresponding solutions for iFIT may need to be covered. For example, how iFIT can fit in the big picture of autonomous networking and support closed control loops. A complete iFIT framework should also consider the cross-domain operations. We leave these topics for future revisions.

8. Security Considerations

Specific security issues are discussed in each individual draft on on-path flow information telemetry.

9. IANA Considerations

This document includes no request to IANA.

10. Contributors

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

11. Acknowledgments

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

12. References

12.1. 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>>.

12.2. Informative References

[CMSketch]

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

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

Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and d. daniel.bernier@bell.ca, "Data Fields for In-situ OAM", [draft-brockners-inband-oam-data-07](#) (work in progress), July 2017.

[I-D.brockners-opsawg-ioam-deployment]

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

[I-D.herbert-ipv4-eh]

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

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

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

[I-D.kumar-ippm-ifa]

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

[I-D.mirsky-ippm-hybrid-two-step]

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

[I-D.song-ippm-ioam-data-validation-option]

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

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

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

[I-D.song-ippm-postcard-based-telemetry]

Song, H., Zhou, T., Li, Z., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry", [draft-song-ippm-postcard-based-telemetry-06](#) (work in progress), October 2019.

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

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

[I-D.song-opsawg-dnp4iq]

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

[I-D.wwx-netmod-event-yang]

Wang, Z., Wu, Q., Xie, C., Bryskin, I., Liu, X., Clemm, A., Birkholz, H., and T. Zhou, "A YANG Data model for ECA Policy Management", [draft-wwx-netmod-event-yang-04](#) (work in progress), November 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.

[passport-postcard]

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

- [RFC2113] Katz, D., "IP Router Alert Option", [RFC 2113](#), DOI 10.17487/RFC2113, February 1997, <<https://www.rfc-editor.org/info/rfc2113>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, [RFC 7011](#), DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.

12.3. URIs

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

Authors' Addresses

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

Email: haoyu.song@futurewei.com

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