Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 14, 2017

F. Brockners
S. Bhandari
S. Dara
C. Pignataro
Cisco
H. Gredler
RtBrick Inc.
J. Leddy
Comcast
S. Youell
JMPC
D. Mozes
Mellanox Technologies Ltd.
T. Mizrahi
Marvell
P. Lapukhov
Facebook
R. Chang
Barefoot Networks
March 13, 2017

Requirements for In-situ OAM
draft-brockners-inband-oam-requirements-03

Abstract

This document discusses the motivation and requirements for including specific operational and telemetry information into data packets while the data packet traverses a path between two points in the network. This method is referred to as "in-situ" Operations, Administration, and Maintenance (OAM), given that the OAM information is carried with the data packets as opposed to in "out-of-band" packets dedicated to OAM. In situ OAM complements other OAM mechanisms which use dedicated probe packets to convey OAM information.

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 http://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

This document discusses requirements for "in-situ" Operations, Administration, and Maintenance (OAM) mechanisms. In this context, "in-situ OAM" refers to the concept of directly encoding telemetry information within the data packet as it traverses the network or telemetry domain. Mechanisms which add tracing or other types of telemetry information to the regular data traffic, sometimes also referred to as "in-band" OAM can complement active, probe-based mechanisms such as ping or traceroute, which are sometimes considered as "out-of-band", because the messages are transported independently from regular data traffic. In terms of "active" or "passive" OAM, "in-situ" OAM can be considered a hybrid OAM type. While no extra packets are sent, in-situ OAM adds information to the packets therefore cannot be considered passive. In terms of the classification given in [RFC7799] in-situ OAM could be portrayed as "hybrid OAM, type 1". "In-situ" mechanisms do not require extra packets to be sent and hence don't change the packet traffic mix within the network. Traceroute and ping for example use ICMP messages: New packets are injected to get tracing information. Those add to the number of messages in a network, which already might be highly loaded or suffering performance issues for a particular path or traffic type.

A number of in-situ as well as in-band OAM mechanisms have been discussed, such as the INT spec for the P4 programming language [P4] or the SPUD prototype [I-D.hildebrand-spud-prototype]. The SPUD prototype uses a similar logic that allows network devices on the path between endpoints to participate explicitly in the tube outside the end-to-end context. Even the IPv4 route-record option defined in [RFC0791] can be considered an in-situ OAM mechanism. Per what was already stated, in-situ OAM complements "out-of-band" mechanisms such as ping or traceroute, or more recent active probing mechanisms, as described in [I-D.lapukhov-dataplane-probe]. In-situ OAM mechanisms can be leveraged where current out-of-band mechanisms do not apply or do not offer the desired characteristics or requirements, such as

proving that a certain set of traffic takes a pre-defined path, strict congruency between overlay and underlay transports is in place, checking service level agreements for the live data traffic, detailed statistics or verification of path selections within a domain, or scenarios where probe traffic is potentially handled differently from regular data traffic by the network devices. [RFC7276] presents an overview of OAM tools.

Compared to probably the most basic example of "in-situ OAM" which is IPv4 route recording [RFC0791], an in-situ OAM approach has the following capabilities:

a. A flexible data format to allow different types of information to be captured as part of an in-situ OAM operation, including but not limited to path tracing information, operational and telemetry information such as timestamps, sequence numbers, or even generic data such as queue size, geo-location of the node that forwarded the packet, etc.

b. A data format to express node as well as link identifiers to record the path a packet takes with a fixed amount of added data.

c. The ability to determine whether any nodes were skipped while recording in-situ OAM information (i.e., in-situ OAM is not supported or not enabled on those nodes).

d. The ability to actively process information in the packet, for example to prove in a cryptographically secure way that a packet really took a pre-defined path using some traffic steering method such as service chaining or traffic engineering.

e. The ability to include OAM data beyond simple path information, such as timestamps or even generic data of a particular use case.

f. The ability to carry in-situ OAM data in various different transport protocols.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Abbreviations used in this document:

ECMP: Equal Cost Multi-Path

IOAM: In-situ Operations, Administration, and Maintenance

LISP: Locator/ID Separation Protocol

MTU: Maximum Transmit Unit

NSH: Network Service Header

NFV: Network Function Virtualization

OAM: Operations, Administration, and Maintenance

PMTU: Path MTU

SFC: Service Function Chain

SLA: Service Level Agreement

SR: Segment Routing

SID: Segment Identifier

VXLAN-GPE: Virtual eXtensible Local Area Network, Generic Protocol Extension

This document defines in-situ Operations, Administration, and Maintenance (in-situ OAM), as the subset in which OAM information is carried along with data packets. This is as opposed to "out-of-band OAM", where specific packets are dedicated to carrying OAM information.

3. Motivation for in-situ OAM

In several scenarios it is beneficial to make information about the path a packet took through the network or through a network device as well as associated telemetry information available to the operator. This includes not only tasks like debugging, troubleshooting, as well as network planning and network optimization but also policy or service level agreement compliance checks. This section discusses the motivation to introduce new methods for enhanced in-situ network diagnostics.

3.1. Path Congruency Issues with Dedicated OAM Packets

Packet scheduling algorithms, especially for balancing traffic across equal cost paths or links, often leverage information contained within the packet, such as protocol number, IP-address or MAC-address. Probe packets would thus either need to be sent from the exact same endpoints with the exact same parameters, or probe packets would need to be artificially constructed as "fake" packets and

inserted along the path. Both approaches are often not feasible from an operational perspective, be it that access to the end-system is not feasible, or that the diversity of parameters and associated probe packets to be created is simply too large. An in-situ mechanism is an alternative in those cases.

In-situ mechanisms are not impacted by differences in the handling of probe traffic compared to other data packets, where probe traffic is handled differently (and potentially forwarded differently) by a router than regular data traffic. This obviously assumes that the addition of in-situ information does not change the forwarding behavior of the packet. Note that in certain implementations, the addition information to a transport protocol changes the forwarding behavior. IPv6 extension header processing is one example. Some implementations process IPv6 packets with extension headers in the "slow" path of a router, as opposed to the "fast" path.

3.2. Results Sent to a System Other Than the Sender

Traditional ping and traceroute tools return the OAM results to the sender of the probe. Even when the ICMP messages that are used with these tools are enhanced, and additional telemetry is collected (e.g., ICMP Multi-Part [RFC4884] supporting MPLS information [RFC4950], Interface and Next-Hop Identification [RFC5837], etc.), it would be advantageous to separate the sending of an OAM probe from the receiving of the telemetry data. In this context, it is helpful to eliminate the requirement that there be a working bidirectional path.

3.3. Overlay and Underlay Correlation

Several network deployments leverage tunneling mechanisms to create overlay or service-layer networks. Examples include VXLAN-GPE, GRE, or LISP. One often observed attribute of overlay networks is that they do not offer the user of the overlay any insight into the underlay network. This means that the path that a particular tunneled packet takes, nor other operational details such as the per-hop delay/jitter in the underlay are visible to the user of the overlay network, giving rise to diagnosis and debugging challenges in case of connectivity or performance issues. The scope of OAM tools like ping or traceroute is limited to either the overlay or the underlay which means that the user of the overlay has typically no access to OAM in the underlay, unless specific operational procedures are put in place. With in-situ OAM the operator of the underlay can offer details of the connectivity in the underlay to the user of the overlay. This could include the ability to find out which underlay elements are shared by overlays and ability to know which overlays are mapped to the same underlay elements. Deployment dependent

underlay transit nodes can be configured to update OAM information in the overlay transport encapsulation. The operator of the egress tunnel router could choose to share the recorded information about the path with the user of the overlay.

Coupled with mechanisms such as Segment Routing (SR) [I-D.ietf-spring-segment-routing], overlay network and underlay network can be more tightly coupled: The user of the overlay has detailed diagnostic information available in case of failure conditions. The user of the overlay can also use the path recording information as input to traffic steering or traffic engineering mechanisms, to for example achieve path symmetry for the traffic between two endpoints. [I-D.brockners-lisp-sr] is an example for how these methods can be applied to LISP.

3.4. SLA Verification

In-situ OAM can help users of an overlay-service to verify that negotiated SLAs for the real traffic are met by the underlay network provider. Different from solutions which rely on active probes to test an SLA, in-situ OAM based mechanisms avoid wrong interpretations and "cheating", which can happen if the probe traffic that is used to perform SLA-check is prioritized by the network provider of the underlay. In active/standby deployments in-situ OAM would only allow for SLA verification of the active path.

3.5. Analytics and Diagnostics

Network planners and operators benefit from knowledge of the actual traffic distribution in the network. When deriving an overall network connectivity traffic matrix one typically needs to correlate data gathered from each individual device in the network. If the path of a packet is recorded while the packet is forwarded, the entire path that a packet took through the network is available to the egress system. This obviates the need to retrieve individual traffic statistics from every device in the network and correlate those statistics, or employ other mechanisms such as leveraging traffic engineering with null-bandwidth tunnels just to retrieve the appropriate statistics to generate the traffic matrix.

In addition, with individual path tracing, information is available at packet level granularity, rather than only at aggregate level - as is usually the case with IPFIX-style methods which employ flow-filters at the network elements. Data-center networks which use equal-cost multipath (ECMP) forwarding are one example where detailed statistics on flow distribution in the network are highly desired. If a network supports ECMP, one can create detailed statistics for the different paths packets take through the network at the egress

system, without a need to correlate/aggregate statistics from every router in the system. Transit devices are off-loaded from the task of gathering packet statistics.

In high-speed networks one can leverage and benefit from packet-accurate measurements with for example hardware-accurate timestamping (i.e., nanosecond-level verification) to support optimized packet scheduling and queuing mechanisms.

3.6. Frame Replication/Elimination Decision for Bi-casting/Active-active Networks

Bandwidth- and power-constrained, time-sensitive, or loss-intolerant networks (e.g., networks for industry automation/control, health care) require efficient OAM methods to decide when to replicate packets to a secondary path in order to keep the loss/error-rate for the receiver at a tolerable level - and also when to stop replication and eliminate the redundant flow. Many Internet of Things (IoT) networks are time sensitive and cannot leverage automatic retransmission requests (ARQ) to cope with transmission errors or lost packets. Transmitting the data over multiple disparate paths (often called bi-casting or live-live) is a method used to reduce the error rate observed by the receiver. Time sensitive networks (TSN) receive a lot of attention from the manufacturing industry as shown by a various standardization activities and industry forums being formed (see e.g., IETF 6TiSCH, IEEE P802.1CB, AVnu).

3.7. Proof of Transit

Several deployments use traffic engineering, policy routing, segment routing or Service Function Chaining (SFC) [RFC7665] to steer packets through a specific set of nodes. In certain cases regulatory obligations or a compliance policy require to prove that all packets that are supposed to follow a specific path are indeed being forwarded across the exact set of nodes specified. If a packet flow is supposed to go through a series of service functions or network nodes, it has to be proven that all packets of the flow actually went through the service chain or collection of nodes specified by the policy. In case the packets of a flow weren’t appropriately processed, a verification device would be required to identify the policy violation and take corresponding actions (e.g., drop or redirect the packet, send an alert etc.) corresponding to the policy. In today’s deployments, the proof that a packet traversed a particular service chain is typically delivered in an indirect way: Service appliances and network forwarding are in different trust domains. Physical hand-off-points are defined between these trust domains (i.e., physical interfaces). Or in other terms, in the "network forwarding domain" things are wired up in a way that traffic

is delivered to the ingress interface of a service appliance and received back from an egress interface of a service appliance. This "wiring" is verified and trusted. The evolution to Network Function Virtualization (NFV) and modern service chaining concepts (using technologies such as Locator/ID Separation Protocol (LISP), Network Service Header (NSH), Segment Routing (SR), etc.) blurs the line between the different trust domains, because the hand-off-points are no longer clearly defined physical interfaces, but are virtual interfaces. Because of that very reason, networks operators require that different trust layers not to be mixed in the same device. For an NFV scenario a different proof is required. Offering a proof that a packet traversed a specific set of service functions would allow network operators to move away from the above described indirect methods of proving that a service chain is in place for a particular application.

Deployed service chains without the presence of a "proof of transit" mechanism are typically operated as fail-open system: The packets that arrive at the end of a service chain are processed. Adding "proof of transit" capabilities to a service chain allows an operator to turn a fail-open system into a fail-close system, i.e. packets that did not properly traverse the service chain can be blocked.

A solution approach could be based on OAM data which is added to every packet for achieving Proof Of Transit (POT).The OAM data is updated at every hop and is used to verify whether a packet traversed all required nodes. When the verifier receives each packet, it can validate whether the packet traversed the service chain correctly. The detailed mechanisms used for path verification along with the procedures applied to the OAM data carried in the packet for path verification are beyond the scope of this document. Details are addressed in [I-D.brockners-proof-of-transit]. In this document the term "proof" refers to a discrete set of bits that represents an integer or string carried as OAM data. The OAM data is used to verify whether a packet traversed the nodes it is supposed to traverse.

3.8. Use Cases

In-situ OAM could be leveraged for several use cases, including:

- Traffic Matrix: Derive the network traffic matrix: Traffic for a given time interval between any two edge nodes of a given domain. Could be performed for all traffic or on a per Quality of Service (QoS) class.

- Flow Debugging: Discover which path(s) a particular set of traffic (identified by an n-tuple) takes in the network. Such a procedure

is particularly useful in case traffic is balanced across multiple paths, like with link aggregation (LACP) or equal cost multi-pathing (ECMP).

- Loss Statistics per Path: Retrieve loss statistics per flow and path in the network.

- Path Heat Maps: Discover highly utilized links in the network.

- Trend Analysis on Traffic Patterns: Analyze if (and if so how) the forwarding path for a specific set of traffic changes over time (can give hints to routing issues, unstable links etc.)

- Network Delay Distribution: Show delay distribution across network by node or links. If enabled per application or for a specific flow then display the path taken along with the delay incurred at every hop.

- SLA Verification: Verify that a negotiated service level agreement (SLA), e.g., for packet drop rates or delay/jitter is conformed to by the actual traffic.

- Low-power Networks: Include application level OAM information (e.g., battery charge level, cache or buffer fill level) into data traffic to avoid sending extra OAM traffic which incur an extra cost on the devices. Using the battery charge level as example, one could avoid sending extra OAM packets just to communicate battery health, and as such would save battery on sensors.

- Path Verification or Service Function Path Verification: Proof and verification of packets traversing check points in the network, where check points can be nodes in the network or service functions.

- Geo-location Policy: Network policy implemented based on which path packets took. Example: Only if packets originated and stayed within the trading-floor department, access to specific applications or servers is granted.

- Device-level Troubleshooting and Optimization: In many cases, network operators could benefit from information specific to a single device. A non-exhaustive list of useful information includes: queue-depths, buffer utilization (either shared or per-port), packet latency measured from a known starting point, packet latency introduced by a single device, and resource utilization (CPU, memory, link bandwidth) of a given device or link. In some cases, this information changes over per-packet timescales (i.e., nanoseconds) and as such it is extremely challenging to collect

and report this info in an accurate and scalable manner. By encoding the information from the forwarding element directly within a data packet (i.e., within the 'fast-path') this information can be added to some or all data packets and then collected and analyzed by human or machine tools. This type of information is particularly valuable for troubleshooting low-level device errors as well as providing a knowledge feedback loop for network and device optimization.

- Custom Network Probing: Active network probing and in-situ OAM can be combined for customized and efficient network probing. This could for example be a customized traceroute.

4. Considerations for In-situ OAM

The implementation of an in-situ OAM mechanism needs to take several considerations into account, including administrative boundaries, how information is recorded, Maximum Transfer Unit (MTU), Path MTU Discovery (PMTUD) and packet size, etc.

4.1. Type of Information to be Recorded

The information gathered for in-situ OAM can be categorized into three main categories: Information with a per-hop scope, such as path tracing; information which applies to a specific set of hops, such as path or service chain verification; information which only applies to the edges of a domain, such as sequence numbers. Note that a single network device could comprise several in-situ OAM hops, for example in case one wants to trace the path of a packet through that device.

- "edge to edge": Information that needs to be shared between network edges (the "edge" of a network could either be a host or a domain edge device): Edge to edge data e.g., packet and octet count of data entering a well-defined domain and leaving it is helpful in building traffic matrix, sequence number (also called "path packet counters") is useful for the flow to detect packet loss.

- "selected hops": Information that applies to a specific set of nodes only. In case of path verification, only the nodes which are "check points" are required to interpret and update the information in the packet.

- "per hop": Information that is gathered at every hop along the path a packet traverses within an administrative domain:

  * Hop by Hop information e.g., Nodes visited for path tracing, Timestamps at each hop to find delays along the path

* Stats collection at each hop to optimize communication in resource constrained networks e.g., battery, CPU, memory status of each node piggy backed in a data packet is useful in low power lossy networks where network nodes are mostly asleep and communication is expensive

4.2. MTU and Packet Size

The recorded data at every hop might lead to packet size exceeding the Maximum Transmit Unit (MTU). A detailed discussion of the implications of oversized IPv6 header chains is found in [RFC7112]. The Path MTU restricts the amount of data that can be recorded for purpose of OAM within a data packet.

If in-situ OAM data is inserted at the edge of the domain (e.g., by intermediate routers) then the MTU on all interfaces with the domain (MTU_INT) MUST be >= the maximum MTU on any "external" facing interfaces (MTU_EXT) and the total size of in-situ OAM data to be recorded MUST be <= (MTU_INT - MTU_EXT).

In-situ OAM comprises two approaches to insert OAM data fields in the packets:

- Pre-allocated: In this case, the encapsulating node inserts empty data fields into the packet to cover the entire domain. The data fields will be incrementally updated/filled as the packet progresses through the network. With pre-allocation the packet size is only changed at the encapsulating node and is kept constant throughout the domain. The pre-allocated approach is beneficial for software data-plane implementations where allocating the required space only once and index into the array to populate the data during transit avoids copy operations at every hop.

- Incremental: Every node that desires to include in-situ OAM information extends the packet as needed. The incremental approach is beneficial for hardware data-plane implementations as it eliminates the need for the transit nodes to read the full array and lookup the pointer in the option prior to updating the data fields contents.

The "incremental" or the "pre-allocated" approaches could even be combined in the same deployment - in which case two in-situ OAM headers would be present in the packet: One for the incremental approach and one for the pre-allocated approach. In such a case one would expect that nodes with a hardware data-plane would update the incremental header, whereas nodes with a software data-plane would process the pre-allocated header.

4.3. Administrative Boundaries

There are several challenges in enabling in-situ OAM in the public Internet as well as in corporate/enterprise networks across administrative domains, which include but are not limited to:

- Deployment dependent, the data fields that in-situ OAM requires as part of a specific transport protocol may not be supported across administrative boundaries.

- Current OAM implementations are often done in the slow path, i.e., OAM packets are punted to router's CPU for processing. This leads to performance and scaling issues and opens up routers for attacks such as Denial of Service (DoS) attacks.

- Discovery of network topology and details of the network devices across administrative boundaries may open up attack vectors compromising network security.

- Specifically on IPv6: At the administrative boundaries IPv6 packets with extension headers are dropped for several reasons described in [RFC7872].

The following considerations will be discussed in a future version of this document: If the packet is dropped due to the presence of the in-situ OAM; If the policy failure is treated as feature disablement and any further recording is stopped but the packet itself is not dropped, it may lead to every node in the path to make this policy decision.

4.3.1. Layered In-Situ OAM Domains

Like any OAM domain, in-situ OAM domains could also be layered/ nested. Layering/nesting of in-situ OAM follows the general approach of OAM layering: An in-situ OAM domain consists of maintenance end-points (MEP) and maintenance intermediate points (MIP). MEP add to or remove the entire set of in-situ OAM data fields from the traffic, while only MIP update or add in-situ OAM data fields. When in-situ OAM layering is employed, a MEP of one layer becomes a MIP in the layer above, while MIP of the lower layer are not visible to the layer above - unless specifically configured otherwise.

Consider the following examples:

- NSH over IPv6: In-situ OAM data fields could be present in both transport protocols: NSH and IPv6, with NSH forming the overlay network and IPv6 forming the underlay network. The network which deploys NSH would form an in-situ OAM domain. In addition each

IPv6 underlay network which connects two NSH nodes forms an in-situ OAM domain. The in-situ OAM domain with NSH as transport could be considered as layered on top of the different in-situ OAM domains which use IPv6 as transport.

- NSH using an in-situ OAM aware transport: Consider a case where the underlay network would not natively support in-situ OAM, still the individual transport nodes would have the capability to "look deep into the packet" and update/add in-situ OAM information in the NSH header. The in-situ OAM domain with NSH as transport could be considered as layered on top of the different in-situ OAM domains which are in-situ OAM aware and connect the individual NSH nodes.

4.4. Selective Enablement

The ability to selectively enable in-situ OAM is valuable. While it may be desirable to enable data collection on all traffic or devices, this may not always be feasible. In-situ OAM collection may also come with a performance impact to forwarding rates or feature capabilities, which may be acceptable in only some locations. For example, the SPUD prototype uses the notion of "pipes" to describe the portion of the traffic that could be subject to in-path inspection. Mechanisms to decide which traffic would be subject to in-situ OAM are outside the scope of this document.

4.5. Forwarding Behavior

In-situ OAM adds additional data fields to live user traffic and as such changes the packet which is also why in-situ OAM is characterized as "hybrid, type 1" OAM. The effectiveness of in-situ OAM as a tool for operations depends on forwarding nodes not altering their forwarding behavior in case of in-situ OAM data fields being present in the packet. As a consequence, an implementation of in-situ OAM should not change the forwarding behavior of the packet, i.e. packets with or without in-situ OAM data fields should be handled the same way by a forwarding node (see also the associated requirement further below). Note that there are implementations where the addition of meta-data to live user traffic might cause the forwarding behavior of the packet to change, e.g. certain implementation handle IPv6 packets with or without extension headers differently (see [RFC7872]).

4.6. Optimization of Node and Interface Identifiers

Since packets have a finite maximum size, the data recording or carrying capacity of one packet in which the in-situ OAM metadata is present is limited. In-situ OAM should use its own dedicated

namespace (confined to the domain in-situ OAM operates in) to represent node and interface IDs to save space in the header. Generic representations of node and interface identifiers which are globally unique (such as a UUID) would consume significantly more bits of in-situ OAM data.

4.7. Loop Communication Path (IPv6-specifics)

When recorded data is required to be analyzed on a source node that issues a packet and inserts in-situ OAM data, the recorded data needs to be carried back to the source node.

One way to carry the in-situ OAM data back to the source is to utilize an ICMP Echo Request/Reply (ping) or ICMPv6 Echo Request/ Reply (ping6) mechanism. In order to run the in-situ OAM mechanism appropriately on the ping/ping6 mechanism, the following two operations should be implemented by the ping/ping6 target node:

1. All of the in-situ OAM fields would be copied from an Echo Request message to an Echo Reply message.

2. The Hop Limit field of the IPv6 header of these messages would be copied as a continuous sequence. Further considerations are addressed in a future version of this document.

5. Requirements for In-situ OAM Data Types

The above discussed use cases require different types of in-situ OAM data. This section details requirements for in-situ OAM derived from the discussion above.

5.1. Generic Requirements

REQ-G1: Classification: It should be possible to enable in-situ OAM on a selected set of traffic (e.g., per interface, based on an access control list specifying a specific set of traffic, etc.) The selected set of traffic can also be all traffic.

REQ-G2: Scope: If in-situ OAM is used only within a specific domain, provisions need to be put in place to ensure that in-situ OAM data stays within the specific domain only.

REQ-G3: Transport independence: Data formats for in-situ OAM shall be defined in a transport independent way. In-situ OAM applies to a variety of transport protocols. Encapsulations should be defined how the generic data formats are carried by a specific protocol.

REQ-G4: Layering: It should be possible to have in-situ OAM information for different transport protocol layers be present in several fields within a single packet. This could for example be the case when tunnels are employed and in-situ OAM information is to be gathered for both the underlay as well as the overlay network. Layering support should not be limited to just underlay and overlay, but include more than two layers.

REQ-G5: MTU size: With in-situ OAM information added, packets MUST NOT become larger than the path MTU.

REQ-G5.1: If due to some reason a packet which contains in situ OAM data fields cannot be forwarded due to the presence of in-situ OAM data fields, the node SHOULD remove the in situ OAM data fields and forward the packet, rather than drop the entire packet.

REQ-G5.2: If the encapsulating router is unable to insert in-situ OAM data fields into a packet, e.g., due to MTU issues, even though it is configured to do so, it should use some operational means to inform the operator (e.g., syslog) about the inability to add in-situ OAM data fields. Even if the in-situ OAM encapsulating node fails to add in-situ OAM data fields, it should forward the packet normally.

REQ-G5.3: MTU size consideration for in-situ OAM MUST take domain specifics into account, e.g., changes of the domain topology due to path protection mechanisms might extend the hop count of a path etc.

REQ-G6: Data structure reuse: The data fields and associated types defined and used for in-situ OAM ought to be reusable for out-of-band OAM telemetry as well.

REQ-G7: Data fields: It is desirable that the format of in-situ OAM data fields leverages already defined data formats for OAM as much as feasible.

REQ-G8: Combination with active OAM mechanisms: In-situ OAM should be usable for active network probing, like for example a customized version of traceroute. Decapsulating in-situ OAM nodes may have an ability to send the in-situ OAM

information retrieved from the packet back to the source address of the packet or to the encapsulating node.

REQ-G9: Unaltered forwarding behavior of in-situ OAM nodes: The addition of in-situ OAM data fields should not change the way packets are forwarded within the in-situ OAM domain.

REQ-G10: Layering of in-situ OAM domains: It should be possible to layer in-situ OAM domains on each other. Layering should be supported within the same, as well as with different transport protocols which carry in-situ OAM data fields.

5.2. In-situ OAM Data with Per-hop Scope

REQ-H1: Missing nodes detection: Data shall be present that allows a node to detect whether all nodes that might participate in in-situ OAM operations have indeed participated.

REQ-H2: Node, instance or device identifier: Data shall be present that allows to retrieve the identity of the entity reporting telemetry information. The entity can be a device, or a subsystem/component within a device. The latter will allow for packet tracing within a device in much the same way as between devices.

REQ-H3: Ingress interface identifier: Data shall be present that allows the identification of the interface a particular packet was received from. The interface can be a logical and/or physical entity.

REQ-H4: Egress interface identifier: Data shall be present that allows the identification of the interface a particular packet was forwarded to. Interface can be a logical or physical entity.

REQ-H5: Time-related requirements

REQ-H5.1: Delay: Data shall be present that allows to retrieve the delay between two or more points of interest within the system. Those points can be within the same device or on different devices.

REQ-H5.2: Jitter: Data shall be present that allows to retrieve the jitter between two or more points of interest within the system. Those points can be within the same device or on different devices. Jitter can be derived from the different

timestamps gathered and does not necessarily need to be an explicit data field.

REQ-H5.3: Wall-clock time: Data shall be present that allows to retrieve the wall-clock time visited a particular point of interest in the system.

REQ-H5.4: Time precision: Time with different precision should be supported. Use-case dependent, the required precision could e.g., be nanoseconds, microseconds, milliseconds, or seconds.

REQ-H6: Generic data fields (like e.g., GPS/Geo-location information): It should be possible to add user-defined OAM data at select hops to the packet. The semantics of the data are defined by the user.

5.3. In-situ OAM with Selected Hop Scope

REQ-S1: Proof of transit: Data shall be present which allows to securely prove that a packet has visited or ore several particular points of interest (i.e., a particular set of nodes).

REQ-S1.1: In case "Shamir’s secret sharing scheme" is used for proof of transit, two data fields, "random" and "cumulative" shall be present. The number of bits used for "random" and "cumulative" data fields can vary between deployments and should thus be configurable.

REQ-S1.2: Enable a fail-open service chaining system to be converted into a fail-closed service chaining system.

5.4. In-situ OAM with End-to-end Scope

REQ-E1: Sequence numbering:

REQ-E1.1: Reordering detection: It should be possible to detect whether packets have been reordered while traversing an in situ OAM domain.

REQ-E1.2: Duplicates detection: It should be possible to detect whether packets have been duplicated while traversing an in situ OAM domain.

REQ-E1.3: Detection of packet drops: It should be possible to detect whether packets have been dropped while traversing an in-situ OAM domain.

6. Security Considerations and Requirements

6.1. General considerations

General Security considerations will be expanded on in a later version of this document.

In-situ OAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring in-situ OAM according to their needs. Still operators need to properly secure the in-situ OAM domain to avoid malicious configuration and use, which could include injecting malicious in-situ OAM packets into a domain.

6.2. Proof of Transit

Threat Model: Attacks on the deployments could be due to malicious administrators or accidental misconfiguration resulting in bypassing of certain nodes. The solution approach should meet the following requirements:

REQ-SEC1: Sound Proof of Transit: A valid and verifiable proof that the packet definitively traversed through all the nodes as expected. Probabilistic methods to achieve this should be avoided, as the same could be exploited by an attacker.

REQ-SEC2: Tampering of meta data: An active attacker should not be able to insert or modify or delete meta data in whole or in parts and bypass few (or all) nodes. Any deviation from the expected path should be accurately determined.

REQ-SEC3: Replay Attacks: A attacker (active/passive) should not be able to reuse the POT bits in the packet by observing the OAM data in the packet, packet characteristics (like IP addresses, octets transferred, timestamps) or even the proof bits themselves. The solution approach should consider usage of these parameters for deriving any secrets cautiously. Mitigating replay attacks beyond a window of longer duration could be intractable to achieve with fixed number of bits allocated for proof.

REQ-SEC4: Pre-play Attacks: A active attacker should not be able to generate or reuse valid POT bits from legitimate packets, in order to prove to the verifier as valid packets. This

slight variant of replay attacks. The attacker extracts POT bits from legitimate packets and ensure they do not reach the verifier. Subsequently reuse those POT bits in crafted packets.

REQ-SEC5: Recycle Secrets: Any configuration of the secrets (like cryptographic keys, initialization vectors etc.) either in the controller or service functions should be re-configurable. Solution approach should enable controls, API calls etc. needed in order to perform such recycling. It is desirable to provide recommendations on the duration of rotation cycles needed for the secure functioning of the overall system.

REQ-SEC6: Secret storage and distribution: Secrets should be shared with the devices over secure channels. Methods should be put in place so that secrets cannot be retrieved by non-authorized personnel from the devices.

7. IANA Considerations

[RFC Editor: please remove this section prior to publication.]

This document has no IANA actions.

8. Acknowledgements

The authors would like to thank Jen Linkova, LJ Wobker, Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra Babu, Akshaya Nadahalli, Ignas Bagdonas, LJ Wobker, Erik Nordmark, Vengada Prasad Govindan, and Andrew Yourtchenko for the comments and advice. This document leverages and builds on top of several concepts described in [I-D.kitamura-ipv6-record-route]. The authors would like to acknowledge the work done by the author Hiroshi Kitamura and people involved in writing it.

9. References

9.1. 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, <http://www.rfc-editor.org/info/rfc2119>.

9.2. Informative References

[I-D.brockners-lisp-sr]
Brockners, F., Bhandari, S., Maino, F., and D. Lewis, "LISP Extensions for Segment Routing", draft-brockners-lisp-sr-01 (work in progress), February 2014.

[I-D.brockners-proof-of-transit]
Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Leddy, J., Youell, S., Mozes, D., and T. Mizrahi, "Proof of Transit", draft-brockners-proof-of-transit-02 (work in progress), October 2016.

[I-D.hildebrand-spud-prototype]
Hildebrand, J. and B. Trammell, "Substrate Protocol for User Datagrams (SPUD) Prototype", draft-hildebrand-spud-prototype-03 (work in progress), March 2015.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-10 (work in progress), November 2016.

[I-D.kitamura-ipv6-record-route]
Kitamura, H., "Record Route for IPv6 (PR6) Hop-by-Hop Option Extension", draft-kitamura-ipv6-record-route-00 (work in progress), November 2000.

[I-D.lapukhov-dataplane-probe]
Lapukhov, P. and r. remy@barefootnetworks.com, "Data-plane probe for in-band telemetry collection", draft-lapukhov-dataplane-probe-01 (work in progress), June 2016.

[P4] Kim, , "P4: In-band Network Telemetry (INT)", September 2015.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <http://www.rfc-editor.org/info/rfc791>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <http://www.rfc-editor.org/info/rfc4884>.

[RFC4950] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "ICMP Extensions for Multiprotocol Label Switching", RFC 4950, DOI 10.17487/RFC4950, August 2007, <http://www.rfc-editor.org/info/rfc4950>.

[RFC5837] Atlas, A., Ed., Bonica, R., Ed., Pignataro, C., Ed., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, DOI 10.17487/RFC5837, April 2010, <http://www.rfc-editor.org/info/rfc5837>.

[RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <http://www.rfc-editor.org/info/rfc7112>.

[RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <http://www.rfc-editor.org/info/rfc7276>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <http://www.rfc-editor.org/info/rfc7665>.

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

[RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <http://www.rfc-editor.org/info/rfc7872>.

Authors' Addresses

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Sashank Dara
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: sadara@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
Comcast

Email: John_Leddy@cable.comcast.com

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom

Email: stephen.youell@jpmorgan.com

David Mozes
Mellanox Technologies Ltd.

Email: davidm@mellanox.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam 20692
Israel

Email: talmi@marvell.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
USA

URI: petr@fb.com

Remy Chang
Barefoot Networks

Email: remy@barefootnetworks.com