Workgroup: CATS
Internet-Draft: draft-fu-cats-oam-fw-00
Published: 3 March 2024
Intended Status: Standards Track
Expires: 4 September 2024
Authors: H. Fu
ZTE Corporation
B. Liu
China Mobile
Z. Li
China Mobile
D.H. Huang
ZTE Corporation
C. Huang
ZTE Corporation
L. Ma
ZTE Corporation
W. Duan
ZTE Corporation

Operations, Administration and Maintenance (OAM) for Computing-Aware Traffic Steering

Abstract

This document describes an OAM framework for Computing-Aware Traffic Steering (CATS). The proposed OAM framework enables the fault and the performance management of end-to-end connections from clients to networks and finally to computing instances. In the following sections, the major components of the framework, the functionalities, and the deployment considerations are elaborated in detail.

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 4 September 2024.

Copyright Notice

Table of Contents

1. Introduction

As described in [I-D.ietf-cats-usecases-requirements], edge computing provides lower response time and higher transmission rate than cloud computing by moving computing instances to the network edge. To meet the requirements of users that are highly distributive, service providers deploy the same type of service instances at multiple edge sites, which involves steering traffic from clients to the most appropriate computing instance.

Compute-aware traffic steering (CATS) [I-D.ldbc-cats-framework] is a traffic engineering approach [I-D.ietf-teas-rfc3272bis] developed to address the aforementioned traffic steering problem. This approach takes into account the dynamic nature of both the computing resources and the network states to optimize the way that traffic is forwarded towards a given service instance. Various metrics can be taken into account to devise and enforce such service-specific and computing-aware traffic steering policies.

To achieve better service assurance, it is necessary to not only rapidly detect whether the QoS provided by the computing networks meets the SLA requirements of clients, but also dynamically trigger

the calculation and the adjustment of both the computing and the networking services. There are OAM technologies developed for Carrier Networks, but these technologies are only deployed in the network domain to facilitate the operations and the maintenance of network operators, and cannot provide measurements of an end-to-end connection from a client to a computing instance.

To this end, this document proposes an OAM architecture based on the CATS framework to extend the coverage of the existing OAM technologies from purely the network to an end-to-end connection from a client to the network and finally to the computing instances. Besides the architecture, the major components and the associated deployment considerations are also described.

2. 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.

3. Terminology

This document makes use of the terms defined in [I-D.ldbc-cats-framework].

*FM: Fault Management.

*PM: Performance Monitoring.

*SI-OAM: Service Instance OAM.

*TC-OAM: Traffic Classifier OAM.

*AF-OAM: Application Flow OAM.

*IOAM: In-situ OAM.

4. Requirements and Motivation

The main objectives of OAM are to detect anomalies before they intensify, reduce the number of traffic flows impacted by these abnormalities, and ensure that network operators fulfill their QoS guarantee commitments to meet the Service Level Agreement(SLA) of clients.

As a traffic engineering method, computing-aware traffic steering (CATS) takes into account the dynamic nature of both the computing resources and the network states to optimize the way that traffic is

forwarded toward a given service instance. However, existing OAM technologies developed for the carrier network cannot be used to collect metrics associated with the computing resources. Therefore, it is necessary to extend the existing OAM technologies to build an end-to-end OAM for CATS. Key objectives include:

*Accelerating the convergence of the CATS control plane: In CATS, the status information of the computing instances is collected by the CATS Service Metric Agent (C-SMA) component and processed at the control plane for performance monitoring and failure detection. However, such a processing process cannot adapt to the rapid change of the computing instance status. Consequently, it is necessary to rapidly detect the degradation of both the computing instances and the network states on the data plane, and trigger CATS Path Selector (C-PS) convergence to avoid black holes.

*Closed-loop network SLA evaluation guarantee: In CATS, the CATS Path Selector (C-PS) calculates and selects the paths towards appropriate egress PEs and computing service instances. In this process, it is necessary to verify whether the calculation and the selection results meet the SLA requirements of clients taking into account both the network states and the computing instance status.

*Closed-loop guarantee of service flow SLAs: In CATS, subsequent packets of service flows in an established session are forwarded through the CATS Traffic Classifier (C-TC) to the same service instance. However, during such a process, the computing/network performance may degrade. To ensure consistent experience for end users, it is necessary to measure the flow-level performance of service instances and make appropriate adjustments, e.g., change segments of routing paths or enable backup paths, according to the SLA requirements.

*Service fault delimiting and troubleshooting: When user experience deteriorates, it is necessary to rapidly locate the fault on the end-to-end path from the user terminal through the network to the service instance to implement fast end-to-end fault location and troubleshooting.

5. Framework and Components

The CATS OAM architecture is shown in Fig. 1. In this architecture, both the CATS router and the Underlay node are deployed with the existing OAM technologies that are developed for the Carrier Network. These OAM technologies are used to detect anomalies and monitor

service performance in the network domain, and can be divided into three categories: link OAM, tunnel OAM, and service OAM.

- In link OAM, anomaly detection and performance monitoring are conducted for a single Ethernet link. The link layer is an optional sublayer implemented in the data link layer between the Logical Link Control (LLC) and the MAC sublayer in the Open Systems Interconnection (OSI) model. Common detection tools of link OAM include IEEE-802 .3ah.

- A tunnel bears multiple services so the tunnel OAM must ensure that the performance of a given service is not degraded when the network fails or the number of services in the tunnel increases. As a result, failure detection and performance monitoring are conducted on the LSP layer to implement service protection. Common detection tools of tunnel OAM include ITU-T Y.1711, MPLS-LM-DM, BFD, etc.

- Service OAM is generally conducted for the L2VPN/L3VPN service layer that is provided by the network to evaluate the service quality and protect services. Common detection tools of service OAM include ITU-T Y.1731, TWAMP, STAMP, etc.

Figure 1: CATS OAM Functional Components

5.1. Component

To achieve the four objectives mentioned in Chapter 3, we designed a CATS OAM architecture based on the CATS architecture and the existing OAM technologies that are developed for the carrier network. This CATS OAM architecture can flexibly support existing OAM detection tools, e.g., the ones mentioned in the previous section, and consists of the following three components:

*SI-OAM component: The functions of this component include (but are not limited to) detecting the failures that happen between the

CATS-Forwarder 2 and the service instance, and measuring the associated metrics such as latency, packet loss, and bandwidth. The SI-OAM component generally would not dive into the internal structure of the network between the CATS-Forwarder 2 and the service instance and only makes the measurements of the end-to-end connection. These measurements are generally fed back to the C-SMA component to achieve faster failure detection and performance monitoring than the CATS control plane, which fulfills the first objective.

*TC-OAM component: The functions of this component include but are not limited to detecting the failures that happen between the CATS-Forwarder 1 and the service instance of a certain specific ID, and measuring the associated metrics such as delay and packet loss. The testing packets are delivered through the CATS Path Selector (C-PS) to the associated service instance according to the corresponding forwarding table entry of the CATS Traffic Classifier (C-TC) to verify whether the measurements of the connection meet the service level agreement (SLA) requirements. And if it does not, recalculation is triggered, which fulfills the second objective.

*AF-OAM component: The functions of this component include but are not limited to measuring the metrics such as delay, packet loss, and bandwidth, of the service flow in CATS. In general, the user experience of an active connection may be affected by a number of factors, such as the processing latency of the service instances may increase or the network performance may degrade due to the increase of the incoming traffic to the service instance. For CATS-Forwarder 1, it is necessary to evaluate whether the SLA requirements of service flows are achieved, and if the SLA requirements are not achieved, conduct appropriate path adjustments to compensate for the deviation as much as possible to ensure the clients have consistent experience. For client terminals, if the experience is degraded, it is necessary to accurately locate where the problem occurs and quickly conduct troubleshooting. Consequently, this component fulfills the third and fourth objectives. It should be noted that related OAM tools can also be developed, so that the entire network stack (L2-L7) can be observed for applications and the entire network stack, instead of merely traditional application-level visibility or network-level visibility, providing a comprehensive solution for operators' efficiency.

5.2. Deployment Consideration

To demonstrate the complete CATS OAM procedure, a proper OAM detection tool needs to be selected and deployed on the network and

service instance hosts of the CATS OAM architecture. The selection of OAM detection tools is out of the scope of this document.

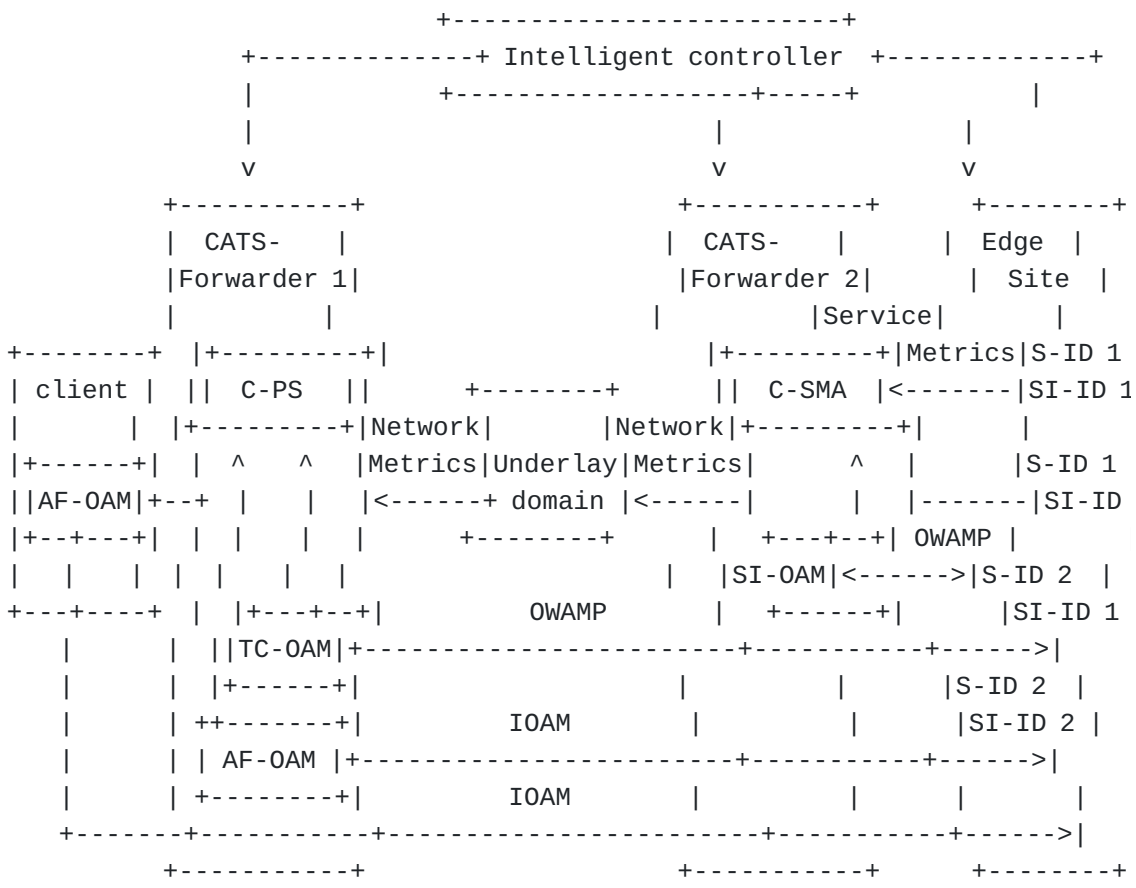

Figure 2: An Example Of CATS OAM Deployment

As illustrated in Fig. 2, the OWAMP and the IOAM tools are selected as examples to describe how the CATS OAM component works with these detection tools to fulfill the four objectives :

*Accelerating the convergence of the CATS control plane: The SI-OAM component is deployed on the CATS-Forwarder 2 and the OWAMP tool is used to measure the delay and packet loss from the CATS-Forwarder 2 to the associated service instance. The source and the destination IP of the detection packets are the CATS-Forwarder 2 interface IP and the service instance IP, respectively. According to the returned packets, the status and the metrics of both the service instance and the network that connects the service instance with the clients are obtained. The SI-OAM component feeds back the measurement results to the C-SMA component, which further spreads the computing resource information in the CATS network to

accelerate CATS Path Selector (C-PS) convergence to avoid black holes.

*Closed-loop network SLA guarantee: The TC-OAM component is deployed on the CATS-Forwarder 1 and the OWAMP tool is used to measure the delay and packet loss from the CATS-Forwarder 1 to the associated service instance. To ensure OWAMP packets are delivered according to the table item of TC, the source and the destination IP addresses of the detection packets are set to the IP address of the interface of CATS-Forwarder 1 and the IP address corresponding to the service ID, respectively. OWAMP packets usually pass through the tunnel to the egress network and are forwarded to the service instance. According to the returned OWAMP packets, the TC-OAM obtains the measurement results and feeds back the results to the C-PS component. If the measurement results deviate from the expected SLAs, recalculation is triggered to fulfill the closed-loop network SLA guarantee for the service ID.

*Closed-loop SLA guarantee for service flow: for service flows that have been initiated, the flow affinity function is executed to guarantee that subsequent packets reach the same service instance as the first packet. To conduct measuring and performance monitoring for the entire end-to-end flows, the flow-based detection tool such as IOAM is selected and the AF-OAM component is deployed on the CATS-Forwarder 1. Note that the PostCard or the PassPort modes are generally used in the flow-based detection and a centralized collector is required to obtain the measurement results and feed the results back to the C-PS. The network path can be adjusted according to the difference between the OAM measurement results and the SLA requirements to ensure a consistent user experience.

*Service fault delimiting and troubleshooting: For fast delimitation and troubleshooting under user experience degradation, the AF-OAM component can be deployed on a user terminal when a flow detection tool such as IOAM is performed. The IOAM can use the postcard mode and can directly report the location where packet loss or longer delay occurs according to the measurement results obtained by a centralized collector. This is a typical scenario of IOAM, and details are not described herein.

6. Operation

The OAM architecture proposed in this document enables CATS to provide robust operations capabilities while forwarding and routing. It should be noted that both the testing packets and the data packets should be delivered via the same path i.e., performance monitoring must be conducted in-band, and the testing traffic must not affect the data traffic. As a result, the testing traffic does shares the

treatments with the data flow being monitored but does not introduce congestion when the network functions normally.

To be added.

7. Management

It is necessary to disclose a set of metrics to support the decision of the operator. The following performance metrics are useful:

- *Delay: elapsed time from the serving gateway to the service instance.

- *Packet loss: the number of lost packets divided by the total number of packets being transmitted from the serving gateway to the compute instance.

- *For each CATS traffic flow, at least one metric that reflects the end-to-end performance is reported.

- *If multiple paths are used for service protection, the paths that malfunction are detected.

- *The service instances that malfunction are detected.

To be added.

7.1. Indicator Collection

The number of metrics and the frequency that these metrics are collected need to be considered when designing the OAM mechanism. The OAM mechanism may be distributed, centralized, or both. The mechanism may be executed periodically or triggered by an event.

To be added.

8. Maintenance

Service protection is designed to mitigate simple network failures faster than the response time expected from the CATS control plane. In the events that affect network operations, e.g., link contexts change, network and computing devices crash/restart, and traffic starts/ends, the CATS control plane needs to perform remediation and re-optimization operations to ensure SLAs of all active flows are satisfied. The control plane should continuously obtain the network status and evaluate whether the current configurations are suitable.

To be added.

9. Security Considerations

TBD.

10. Acknowledgements

To be added upon contributions, comments and suggestions.

11. IANA Considerations

TBA

12. References

12.1. Normative References

[I-D.ldbc-cats-framework] Li, C., Du, Z., Boucadair, M., Contreras, L. M., and J. Drake, "A Framework for Computing-Aware Traffic Steering (CATS)", Work in Progress, Internet-Draft, draft-ldbc-cats-framework-06, 8 February 2024, <https://datatracker.ietf.org/doc/html/draft-ldbc-cats-framework-06>.

[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>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <https://www.rfc-editor.org/info/rfc4656>.

[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, <https://www.rfc-editor.org/info/rfc7276>.

[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>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header

(SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC9378] Brockners, F., Ed., Bhandari, S., Ed., Bernier, D., and T. Mizrahi, Ed., "In Situ Operations, Administration, and Maintenance (IOAM) Deployment", RFC 9378, DOI 10.17487/RFC9378, April 2023, <https://www.rfc-editor.org/info/rfc9378>.

12.2. Informative References

[I-D.ietf-cats-usecases-requirements]
Yao, K., Trossen, D., Boucadair, M., Contreras, L. M., Shi, H., Li, Y., Zhang, S., and Q. An, "Computing-Aware Traffic Steering (CATS) Problem Statement, Use Cases, and Requirements", Work in Progress, Internet-Draft, draft-ietf-cats-usecases-requirements-02, 1 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-usecases-requirements-02>.

[I-D.ietf-teas-rfc3272bis]
Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-27, 12 August 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-rfc3272bis-27>.

[I-D.li-dyncast-architecture] Li, Y., Iannone, L., Trossen, D., Liu, P., and C. Li, "Dynamic-Anycast Architecture", Work in Progress, Internet-Draft, draft-li-dyncast-architecture-08, 16 January 2023, <https://datatracker.ietf.org/doc/html/draft-li-dyncast-architecture-08>.

Authors' Addresses

Huakai Fu
ZTE Corporation
Wuhan
China

Email: fu.huakai@zte.com.cn

Bo Liu
China Mobile
Beijing
China

Email: liubo@chinamobile.com

Zhenqiang Li
China Mobile
Beijing
China

Email: lizhenqiang@chinamobile.com

Daniel Huang
ZTE Corporation
Nanjing
China

Email: huang.guangping@zte.com.cn

Cheng Huang
ZTE Corporation
Shanghai
China

Email: huang.cheng13@zte.com.cn

Liwei Ma
ZTE Corporation
Nanjing
China

Email: ma.liwei1@zte.com.cn

Wei Duan
ZTE Corporation
Nanjing
China

Email: duan.wei1@zte.com.cn