

CATS
Internet-Draft
Intended status: Standards Track
Expires: 20 February 2026

H. Fu
D.H. Huang
L. Ma
W. Duan
B. Tan
ZTE Corporation
19 August 2025

Flow-Level Load Balancing of Computing-Aware Traffic Steering (CATS)
draft-fu-cats-flow-lb-02

Abstract

This document proposes a flow-level load balancing solution for CATS, and is designed to effectively manage CS-ID traffic by addressing issues like frequent control plane operations and uneven use of computing resources. The approach entails concurrently identifying multiple next-hop choices, factoring in both network pathways and service instances. Traffic is then distributed among these service instances using flow-based load balancing, which relies on the five-tuple characteristics of packets.

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 20 February 2026.

Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	3
3. Terminology	3
4. Problem Statement	3
5. Flow-level load balancing	4
5.1. Designing principles	4
5.2. Control plane Considerations	6
5.3. Forwarding table entries	6
5.4. Work flow example of flow-level load balance for CATS . .	8
5.5. Control Plane Load Reduction	9
6. Security Considerations	9
7. Acknowledgements	9
8. IANA Considerations	9
9. References	9
9.1. Normative References	9
9.2. Informative References	10
Authors' Addresses	11

1. Introduction

Computing-Aware Traffic Steering (CATS) [I-D.ldbrc-cats-framework] targets efficient routing at the network edge, directing traffic between service clients and providers. It relies on real-time computing and network status data for informed decisions. CATS operates as an overlay system, choosing optimal service instances for requests, yet the CATS framework does not assume any specific data plane and control plane solutions.

This proposal suggests deploying a flow-level load balancing mechanism for CATS to tackle issues related to frequent control plane activities and imbalanced resource utilization. The approach focuses on CS-ID traffic and involves determining multiple next-hop alternatives by considering both network routes and service instance identifiers. Traffic is then distributed based on the five-tuple of packets, ensuring efficient workload allocation. The control plane concurrently identifies multiple paths and service instances that adhere to Service Level Agreement (SLA) Requirements, while the data plane enhances forwarding effectiveness through equal-cost multi-path routing techniques.

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].

- * UCMP: Unequal Cost Multiple Path.
- * ECMP: Equal Cost Multiple Path.
- * SLA: Service Level Agreement.
- * VRF-ID: Virtual Routing and Forwarding Identifier.

4. Problem Statement

The current CATS network technologies utilize periodic or threshold-triggered resource status reports to optimize the selection of service instances and paths to meet quality of service requirements. However, this approach faces two primary challenges:

*Firstly, there is an uneven utilization of computing resources, resulting in imbalances in resource distribution when longer reporting intervals or threshold triggers are employed. This can lead to a situation where the same service instance receives multiple requests, causing a temporary imbalance in resource distribution.

*Secondly, the frequent operations on the control plane due to resource imbalances can result in increased calculation and update tasks on the control plane. While incremental calculation and policy delivery may provide some relief, they do not address the underlying issue.

In order to address these obstacles and achieve a fairer and more effective distribution of resources while meeting service level agreement requirements, it is crucial to tackle issues related to uneven resource usage and manage excessive load.

5. Flow-level load balancing

5.1. Designing principles

To address the aforementioned challenges, a flow-level load balancing solution has been developed based on the following guiding principles:

1. Minimizing the impact of state changes on individual calculation instances and reducing the frequency of calculations and updates in the control plane.
2. Extending the update intervals for CATS routing table entries to ensure load balancing on the data plane.

The proposed solution, as detailed in this document, involves the simultaneous calculation of multiple network paths and service instances that meet SLA requirements. Each unique next-hop entry in the CATS routing table contains both a network path and a service instance, facilitating non-equivalent load balancing during service forwarding to optimize overall performance.

Furthermore, the Flow-Level Load Balancing of Computing-Aware Traffic Steering is constructed based on the framework established in the CATS architecture [I-D.ldb-cats-framework](Figure 1 for a visual representation).

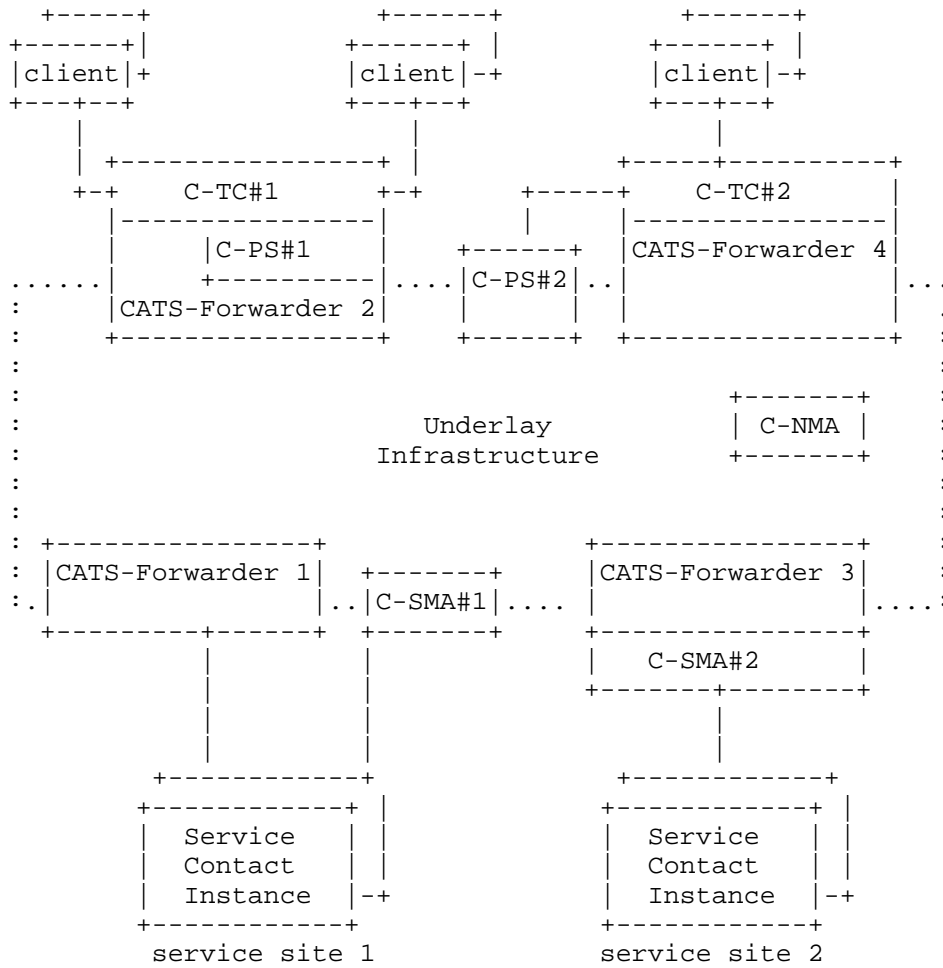


Figure 1: CATS-Functional-Components

Both documents, [I-D.lbdd-cats-dp-sr] and [I-D.fu-cats-muti-dp-solution], utilize anycast IP addresses for computing services in CS-ID. When the egress CATS-forwarder is connected to multiple service instances, traffic is steered to the appropriate instance via END.DX4/6 Service SID. Conversely, with a single service instance, traffic is steered using the END.DT4/6 Service SID along with the anycast IP address.

To simplify the expression, the selection result of C-PS is called CATS routing table, and the entry used for forwarding packets on the forwarding plane is called CATS forwarding table.

5.2. Control plane Considerations

The C-PS component is conventionally situated in the head node or central network controller. Here, it collects service instance status like CS-ID , CIS-ID, and Metrics through the C-SMA component. Furthermore, it obtains network capacity and status information via the C-NMA component.

The C-PS component, considering the SLA requirements associated with the CS-ID, processes the collected data to determine viable network path and service instances that conform to the SLA. Subsequently, it allocates traffic share ratios among these identified paths.

The outcome is translated into VRF-ID, CS-ID, and a set of multiple next-hop destinations (such as SR-Policy and service SID) which incorporate load sharing proportions to direct the forwarding of service packets within the data plane. It is crucial to limit the number of next-hops in accordance with hardware capabilities and opt for the most efficient paths that adhere to the SLA requirements.

Figure 2 shows an example of a representation of multi-next-hop CATS routing table designed for a specific CS-ID1.

VRF-ID	PREFIX	NEXT HOP		
		SR-Policy	Service SID	Load Sharing Ratio
100	CS-ID1	SR-Policy1(2ms)	END.DX-1	20%
		SR-Policy1(2ms)	END.DX-2	30%
		SR-Policy2(1.5ms)	END.DX-3	30%
		SR-Policy2(1.5ms)	END.DX-4	20%

Figure 2: An example of CATS routing table

5.3. Forwarding table entries

The C-PS component calculates the CATS routing table, which is subsequently translated into a data plane strategy. This strategy entails decomposing the Unequal-Cost Multiple Path (UCMP) routing for traffic load balancing into multiple Equal-Cost Multi-Path (ECMP) entries. This process resembles the conventional conversion of IP UCMP to ECMP at the hardware level.

The C-PS component calculates the CATS routing table, which is subsequently translated into a data plane strategy. This strategy entails decomposing the Unequal-Cost Multiple Path (UCMP) routing for traffic load balancing into multiple Equal-Cost Multi-Path (ECMP) entries. This process resembles the conventional conversion of IP UCMP to ECMP at the hardware level.

With reference to [I-D.fu-cats-hybrid-fwd] , depending on the selected forwarding mode, multiple network paths and service-instance entries may persist in the control plane. When the first service packet is received, the control plane shall select the corresponding network path and service instance based on the packet's tuple, generate a flow-affinity table, and use this table to direct the forwarding of subsequent packets.

For instance, if the original CATS routing table indicates four next-hops with a load-sharing ratio of 2:3:3:2, this would result in 10 ECMP routing entries upon conversion. To maintain consistency with the ECMP load-balancing rule, each of these entries is then duplicated according to a predefined UCMP ratio. This ensures that packet forwarding occurs efficiently and aligns with the ECMP balance principle.

Figure 3 shows an example of the CATS forwarding table following the changes.

VRF-ID	PREFIX	SR-Policy	Service SID	offset
100	CS-ID1	SR-Policy1(2ms)	END.DX-1	0
		SR-Policy1(2ms)	END.DX-1	1
		SR-Policy1(2ms)	END.DX-2	2
		SR-Policy1(2ms)	END.DX-2	3
		SR-Policy1(2ms)	END.DX-2	4
		SR-Policy2(1.5ms)	END.DX-3	5
		SR-Policy2(1.5ms)	END.DX-3	6
		SR-Policy2(1.5ms)	END.DX-3	7
		SR-Policy2(1.5ms)	END.DX-4	8
		SR-Policy2(1.5ms)	END.DX-4	9

Figure 3: An example of CATS forwarding table

5.4. Work flow example of flow-level load balance for CATS

The following procedure describe how it works in general.

- 1) Ingress CATS-Forwarder gets user's computing service request, extracting VRF-ID, interface, and CS-ID.
- 2) Ingress CATS-Forwarder checks the forwarding table with these IDs. If found, it proceeds to Step 3; otherwise, it discards the packet.
- 3) Ingress CATS-Forwarder searches the flow affinity table. With a match, it retrieves SR-Policy and Service SID to forward the packet in Step 5; if not, it goes to Step 4.
- 4) Ingress CATS-Forwarder hashes packet attributes, finds next-hop in the forwarding table, gets SR-Policy and Service SID, and creates a flow affinity table entry for future packets. This ensures consistent routing and load balancing.
- 5) Ingress CATS-Forwarder adds SRH based on gathered information and forwards the IPv6 packet using SRH for underlay routing.

6)Egress CATS-forwarder removes SRH and sends the packet based on Service SID: END.DX sends to a tunnel, END.DT uses destination IP according to VRF-ID.

7)The service instance processes the request and sends a response.

5.5. Control Plane Load Reduction

To reduce the pressure on the control plane, in addition to calculating multiple next hops at once for flow-level load balancing, a threshold mechanism is also used on the control plane to manage the pressure brought by the diffusion of perceived computing instance states.

The C-SMA component uses multi-level gradient thresholds to monitor the performance of service instances, such as latency and bandwidth. It sets different standards for delay (x_1, x_2, \dots, x_M) and bandwidth (y_1, y_2, \dots, y_N). Once the service instance delay or bandwidth reaches the critical status, the C-PS component immediately calculates and selects the path to the service location and instance.

To enhance the process, it is suggested to blend threshold alerts with session-based load balancing. This could evenly distribute user sessions across networks and instances, and minimize instances surpassing limits, creating a low-frequency feedback loop that lessens control overhead.

It is important to highlight that load balancing operations are conducted at the ingress CATS-Forwarder. Before creating a flow affinity table, the CATS forwarding table can be directly used by the data plane or control plane to process the first packet, and the next hop is determined by the 5-tuple HASH.

6. Security Considerations

TBD.

7. Acknowledgements

To be added upon contributions, comments and suggestions.

8. IANA Considerations

TBA

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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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] Filtsfils, 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] Filtsfils, 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>>.
- [RFC8986] Filtsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

9.2. Informative References

- [I-D.fu-cats-hybrid-fwd]
Huakai.Fu, Yi, X., Pang, B., Yuan, D., Duan, W., and C. Miao, "Hybird Fordwarding of Computing-Aware Traffic Steering (CATS)", Work in Progress, Internet-Draft, draft-fu-cats-hybrid-fwd-00, 30 June 2025, <<https://datatracker.ietf.org/doc/html/draft-fu-cats-hybrid-fwd-00>>.
- [I-D.fu-cats-muti-dp-solution]
莉倅克讎キ, Liu, B., Li, Z., Huang, D., Yuan, D., Ma, L., and W. Duan, "Analysis for Multiple Data Plane Solutions of Computing-Aware Traffic Steering", Work in Progress, Internet-Draft, draft-fu-cats-muti-dp-solution-02, 25 February 2025, <<https://datatracker.ietf.org/doc/html/draft-fu-cats-muti-dp-solution-02>>.

- [I-D.huang-service-aware-network-framework]
Huang, D., Tan, B., and D. Yang, "Service Aware Network Framework", Work in Progress, Internet-Draft, draft-huang-service-aware-network-framework-01, 22 November 2022, <<https://datatracker.ietf.org/doc/html/draft-huang-service-aware-network-framework-01>>.
- [I-D.ietf-cats-usecases-requirements]
Yao, K., Contreras, L. M., Shi, H., 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-07, 10 June 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-cats-usecases-requirements-07>>.
- [I-D.lbdd-cats-dp-sr]
Li, C., Du, Z., and J. Drake, "Computing-Aware Traffic Steering (CATS) Using Segment Routing", Work in Progress, Internet-Draft, draft-lbdd-cats-dp-sr-05, 1 August 2025, <<https://datatracker.ietf.org/doc/html/draft-lbdd-cats-dp-sr-05>>.
- [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>>.
- [I-D.li-dynicast-architecture]
Li, Y., Iannone, L., Trossen, D., Liu, P., and C. Li, "Dynamic-Anycast Architecture", Work in Progress, Internet-Draft, draft-li-dynicast-architecture-08, 16 January 2023, <<https://datatracker.ietf.org/doc/html/draft-li-dynicast-architecture-08>>.
- [RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil, "Architectural Considerations of IP Anycast", RFC 7094, DOI 10.17487/RFC7094, January 2014, <<https://www.rfc-editor.org/info/rfc7094>>.

Authors' Addresses

Huakai Fu
ZTE Corporation
Wuhan
China

Email: fu.huakai@zte.com.cn

Daniel Huang
ZTE Corporation
Nanjing
China
Email: huang.guangping@zte.com.cn

Liwei Ma
ZTE Corporation
Nanjing
China
Email: ma.liweil@zte.com.cn

Wei Duan
ZTE Corporation
Nanjing
China
Email: duan.weil@zte.com.cn

Bin Tan
ZTE Corporation
ShangHai
China
Email: tan.bin@zte.com.cn