

CATS
Internet-Draft
Intended status: Standards Track
Expires: 2 January 2026

H. Fu
ZTE Corporation
X. Yi
China Unicom
B. Pang
Beijing Jiaotong University
D. Yuan
W. Duan
C. Miao
ZTE Corporation
1 July 2025

Hybird Fordwarding of Computing-Aware Traffic Steering (CATS)
draft-fu-cats-hybrid-fwd-00

Abstract

This document proposes a hybrid forwarding mechanism for CATS (Computing-Aware Traffic Steering). The mechanism integrates the CATS forwarding table with the IP forwarding table, optimizing the utilization of forwarding table capacity. By customizing the forwarding model based on service identifiers, it can accommodate both experience-sensitive and non-experience-sensitive services. Additionally, it supports flow-granularity load balancing, enhancing the utilization of computing and networking resources while ensuring differentiated service requirements are satisfied.

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 2 January 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	4
5. Hybird Fordwarding Model	4
5.1. Table Management and Working Mechanism	6
5.2. Considerations for Hybrid Forwarding	8
6. Security Considerations	11
7. Acknowledgements	11
8. IANA Considerations	11
9. References	11
9.1. Normative References	11
9.2. Informative References	12
Authors' Addresses	13

1. Introduction

Computing-Aware Traffic Steering (CATS) [I-D.ldbcb-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.

As the trend of the integration of networking and computing deepens, users are placing increasingly higher demands on both the experience of network services and the efficiency of resource utilization. This is particularly evident for services with low latency requirements, such as AR/VR, and services with explicit requirements for computing resources. Traditional traffic steering mechanisms are not capable of meeting these evolving demands. Relevant studies have

demonstrated that traffic steering that takes into account computing resource conditions can significantly improve service performance and user experience [I-D.ietf-cats-usecases-requirements] .

To address these challenges, the Computing-Aware Traffic Steering (CATS) framework has been proposed in [I-D.ldbc-cats-framework]. The primary goal of the CATS framework is to efficiently steer traffic at the edge device of the network instructed by traffic steering decisions referred to both computing and network conditions. The CATS framework is designed as an overlay mechanism for selecting the optimal service instances, without relying on specific data-plane or control-plane solutions.

Against this backdrop, this document proposes a hybrid forwarding mechanism for CATS. The key features of this mechanism are as follows:

- * ***Integration of Forwarding Tables***: The CATS forwarding table is integrated with the IP forwarding table to achieve shared and efficient utilization of the forwarding table capacity.
- * ***Customized Forwarding Model***: The forwarding model is customized based on service identifiers to meet the demands of both experience-sensitive and non-experience-sensitive services.
- * ***Flow-Level Load Balancing***: The mechanism supports flow-granularity load balancing, enhancing the utilization of computing and networking resources while meeting the diverse requirements for differentiated services.

This hybrid forwarding mechanism provides a flexible and efficient solution for network vendors and service providers, facilitating the deployment of computing-aware traffic steering.

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 adopts the terminology defined in [I-D.ldbc-cats-framework], [I-D.lbdd-cats-dp-sr] and [I-D.fu-cats-flow-lb].

4. Problem Statement

To effectively steering service request packets to the appropriate forwarding paths and instances within suitable computing instances, CATS faces several issues and challenges with the existing technical solutions:

- * ***Issue 1*:** Flow Affinity Mechanism Complexity: To ensure continuous access to the same computing resource during a service session, a flow affinity mechanism based on five-tuples or three-tuples is required. However, incorporating such lookups into the general routing process introduces additional processing overhead, which significantly increases packet forwarding latency and consumes obvious chip resources. This additional processing burden can degrade the performance of latency-sensitive services.
- * ***Issue 2*:** Control Plane Overhead: Computing services are characterized by ubiquity and state changes over time. If the control plane indiscriminately performs periodic or event-driven route computation and table updates for all managed computing service types, it will excessively consume CPU resources and increase the burden of system management and maintenance.
- * ***Issue 3*:** Initial Packet Handling and Latency: Some solutions attempt to alleviate the control plane load by throttling and sending the first computing request packet to the control plane to generate a flow affinity table before the packet hits the flow affinity table. However, there is a risk of packet loss before the table entry takes effect, and the control plane processing of the packet can also lead to high latency for the first frame, which affects the quality of experience and service continuity.

5. Hybird Fordwarding Model

To address the aforementioned issues, we propose a hybrid forwarding mechanism. This independently designed solution features high flexibility, enabling customized service strategies tailored to different service characteristics:

- * ***Experience-Sensitive Services*:** Utilizes direct table-based forwarding to guarantee stringent QoS requirements and optimal user experience.
- * ***Non-Sensitive Services*:** Employs packet-driven table updates to

minimize resource overhead while maintaining forwarding efficiency.

Key advantages of this mechanism include:

- (i) Reduced Control/Forwarding Plane Overhead Mitigates update pressure through optimized table management.
- (ii) Decoupled Legacy Router Dependency Enhances the flexibility of deployment in modern network environments.
- (iii) High Implementability and Scalability Supports evolving service demands while maintaining backward compatibility.

This approach provides an efficient and scalable solution for heterogeneous service delivery in next-generation networks.

Hybrid Forwarding of Computing-Aware Traffic Steering (based on the framework defined in the CATS architecture [I-D.ldb-cats-framework](see Figure 1, which is reproduced here for better understanding).

CATS ROUTING TABLE

VPN ID	Prefix IP	CATS-IND	MODE-IND	NextHop	
				SR-Policy	Service SID
1	CS-ID-1	1	1	SR-P-1	END-DX6-1
1	CS-ID-1	1	0	SR-P-2	END-DX6-2

CATS FORWARDING TABLE

VPN ID	Prefix IP	CATS-IND	MODE-IND	NextHop	
				SR-Policy	Service SID
1	CS-ID-1	1	1	SR-P-1	END-DX6-1
1	CS-ID-1	1	0	NULL	NULL

Figure 2: Illustration of CATS Table Management by the Control Plane

As shown in Figure 2, the figure presents a specific form of managing the CATS table entries on the control plane. The explanation for the entries in the table is as follows:

- * ***CS-ID Usage***: The CS-ID typically employs anycast IP addresses to identify services. As specified in [I-D.lbdd-cats-dp-sr] and [I-D.fu-cats-muti-dp-solution], the forwarding mechanism varies based on service instance connectivity: (1) when an egress gateway is connected to multiple Service instances, traffic is forwarded to a specific instance via either a tunnel or a general interface (on the same CATS-Forwarder) using an END.DX4/6 Service SID; (2) when a single service instance is involved, an END.DT4/6 Service SID enables direct forwarding via anycast IP. Consequently, each Service SID maintains a strict one-to-one correspondence with its associated service instance.
- * ***C-PS Deployment and Function***: The C-PS component is deployed on the head node or a centralized Computation-network controller. It collects and reports CS-ID, CIS-ID, and Metric information related to service instance instances via C-SMA, as well as network-related capabilities and status information via C-NMA. Based on the collected information, the C-PS calculates the optimal network forwarding path and service instance, which is

then referred to as the CATS routing table. The process of inserting or downloading these table entries to the forwarding plane is called the CATS forwarding table update.

- * ***Design and Function of the CATS Forwarding Table***: The CATS forwarding table shares space with the traditional IP forwarding table. To distinguish CATS service packets from other types of packets, CS-ID is used for marking and identification, with CATS-IND serving as an indicator. Additionally, to differentiate service requirements (e.g., delay-sensitive or other types), MODE-IND can be used for marking and labeling services based on CS-ID. As shown in Figure 2, relevant table entries are provided for reference. For example, MODE-IND = 1 indicates experience-sensitive services, while MODE-IND = 0 indicates non-sensitive services. It should be noted that although Figure 1 shows only one NextHop for the same CS-ID in the CATS routing and forwarding tables, multiple NextHops can still be supported to achieve flow-level load balancing according to [I-D.fu-cats-flow-lb].
- * ***CATS Flow Affinity Table Mechanism***: To ensure that service sessions consistently access the same service instance, CATS introduces the flow affinity table. This table uses a five-tuple or three-tuple as the KEY value, with network path and service instance as related attributes. By looking up the table with the five-tuple or three-tuple learned from the service packet, the relevant attributes are retrieved, and the forwarding action is executed accordingly.

5.2. Considerations for Hybrid Forwarding

To meet the demands of both experience-sensitive and non-sensitive services, we propose the following forwarding mechanism in conjunction with the key table entries discussed in Section 5.1.

Since the CS-ID would be encoded in a form of anycast addresses, the CATS forwarding table and the IP forwarding table can share the same space. Typically, traffic is distinguished by looking up the IP forwarding table. Once CATS packets are identified, further flow forwarding processing is carried out. This approach reduces invasive modifications to the forwarding mechanism while maintains high-performance forwarding for both traditional services and CATS services.

Specifically:

- (i) * For the case in which `MODE-IND = 1`, the issued CATS forwarding table includes SR-policy and Service SID information. The control plane only sends updates to the data plane when table entries change, ensuring that both the first packet and subsequent packets are processed directly at the forwarding plane, while the control plane is solely responsible for generating the flow affinity table.
- (ii) * For the case in which `MODE-IND = 0`, the issued CATS forwarding table does not include SR-policy and Service SID information. In this case, regardless of how the attributes of the CATS routing table change dynamically, the C-PS will not update the forwarding plane to reduce the overhead of table updates. The first packet is uploaded to the control plane, which queries the CATS routing table, encapsulates and forwards the packet, and generates the flow affinity table to be sent to the forwarding plane. Subsequent packets are forwarded based on the flow affinity table in the data plane.

- (i) * Upon arrival, a packet first looks up the IP forwarding table. If the CATS_IND in the lookup result is 0, the packet is identified as a conventional non-CATS service packet; otherwise, it is identified as a CATS service packet, and the flow forwarding process is initiated.
- (ii) * After entering the flow forwarding processing, the MODE-IND identifier in the lookup result is further recognized, and different processing is performed based on its value.

- (iii) * If MODE-IND = 1, the first packet extracts five-tuple/three-tuple information, sends it to the control plane to generate the flow affinity table and send it to the forwarding plane. At the same time, the first packet is encapsulated and forwarded directly at the forwarding plane based on the SR-policy and Service SID information carried in the lookup result (Option 1).
- (iv) * If MODE-IND = 0, the forwarding table entry does not carry SR-policy and Service SID information. The first packet is sent directly to the control plane, which queries the CATS routing table, generates the flow affinity table, and sends it to the forwarding plane. At the same time, the first packet is encapsulated and forwarded at the control plane based on the SR-policy and Service SID information carried in the lookup result (Option 2).

Based on this, for experience-sensitive services, the first packet can be forwarded directly at the forwarding plane to reduce first-packet latency; for non-experience-sensitive services, the first packet can be processed at the control plane to reduce the processing overhead of control-plane table updates.

In summary, by adopting the above measures, a flexible CATS hybrid forwarding solution can be selected according to the services corresponding to the characteristics of different services associated with the CS-ID, ensuring optimal performance for different service types.

6. Security Considerations

TBD.

7. Acknowledgements

To be added upon contributions, comments and suggestions.

8. IANA Considerations

TBA

9. References

9.1. Normative References

[I-D.ldbrc-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>>.
- [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] Filts, 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] Filts, 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] Filts, 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-flow-lb] 付楷, Huang, D., Ma, L., Duan, W., and B. Tan, "Flow-Level Load Balancing of Computing-Aware Traffic Steering (CATS)", Work in Progress, Internet-Draft, draft-fu-cats-flow-lb-01, 25 February 2025, <<https://datatracker.ietf.org/doc/html/draft-fu-cats-flow-lb-01>>.
- [I-D.fu-cats-muti-dp-solution] 付楷, 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-04, 28 January 2025, <<https://datatracker.ietf.org/doc/html/draft-lbdd-cats-dp-sr-04>>.

[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

Xinxin Yi
China Unicom
Beijing
China
Email: yixx3@chinaunicom.cn

Bo Pang
Beijing Jiaotong University
Beijing
China
Email: bopang@bjtu.edu.cn

Dongyu Yuan
ZTE Corporation
Nanjing
China
Email: yuan.dongyu@zte.com.cn

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

Chuanyang Miao
ZTE Corporation
Nanjing
China
Email: miao.chuanyang@zte.com.cn