

CATS  
Internet-Draft  
Intended status: Standards Track  
Expires: 30 August 2026

H. Fu  
ZTE Corporation  
B. Liu  
Z. Li  
China Mobile  
D.H. Huang  
D. Yuan  
W. Duan  
ZTE Corporation  
26 February 2026

Analysis for Multiple Data Plane Solutions of Computing-Aware Traffic  
Steering  
draft-fu-cats-muti-dp-solution-04

Abstract

This document proposes a data plane framework for Computing-Aware Traffic Steering (CATS), detailing multiple optional solutions, comparing their key characteristics and distinctions, and analyzing applicable scenarios to provide a reference for implementation.

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 30 August 2026.

Copyright Notice

Copyright (c) 2026 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. Overview . . . . .	3
5. Solution 1: Service ID carried in an anycast IP with bidirectional address translation mode . . . . .	5
6. Solution 2: Service ID carried in the IPv6 EH with unidirectional address translation mode . . . . .	7
7. Solution 3: Service ID carried in an anycast IP with TUNNEL/MAC mode . . . . .	8
8. Solution Comparison Analysis . . . . .	10
9. Security Considerations . . . . .	11
10. Acknowledgements . . . . .	11
11. IANA Considerations . . . . .	11
12. References . . . . .	11
12.1. Normative References . . . . .	11
12.2. Informative References . . . . .	12
Authors' Addresses . . . . .	13

## 1. Introduction

As described in [I-D.ietf-cats-usecases-requirements] , traffic steering which takes computing resource conditions and metrics into account would benefit computing-related services, including latency-sensitive services which rely upon the use of augmented reality or virtual reality (AR/VR) techniques.

Computing-Aware Traffic Steering (CATS) [I-D.ietf-cats-framework] aims to solve the problem that how the network edge can steer traffic between clients of a service and sites offering the service. To enable the computing- and network-aware traffic steering decisions, awareness of computing information and network information are fundamental premises. The CATS architecture is an overlay framework for the selection of the most appropriate service contact instance for placing a service request. However, the CATS framework does not assume any specific data plane and control plane solutions.

This document proposes several potential data plane solutions for the realization of CATS, and compares their key features and main application scenarios. These solutions use an anycast IP address or digital identification as the Computing-aware Service ID (CS-ID) associated with a service.

## 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.ietf-cats-framework] .

## 4. Overview

As illustrated in Figure 1, underlay network infrastructure and devices are deployed between an ingress CATS-Router and service contact instances, where corresponding CATS functionality is deployed at the ingress CATS-Router 1 and egress CATS-Router 2/3. An egress CAT-router connects to multiple service sites. At a specific service site, single service contact instance or multiple service contact instances are deployed.

CATS overlay encapsulation is established from the ingress CATS-Router to the egress CATS-Router connected to the service site. For ease of description in this document, it is assumed that a specific tunnel between CATS-Routers is an SRv6 Policy [RFC8986].

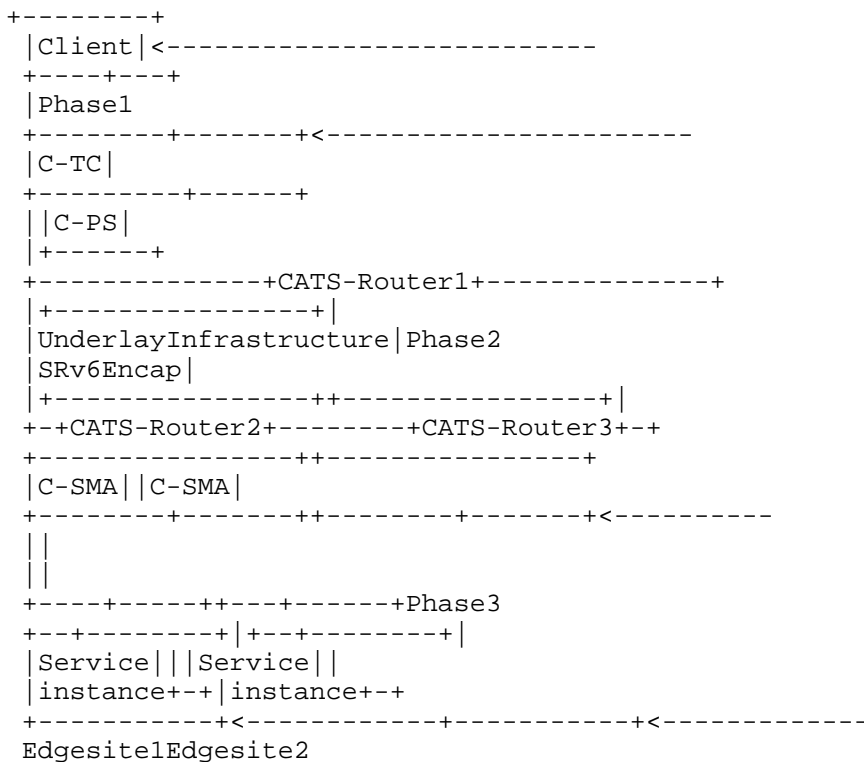


Figure 1: CATS Data Plane Workflow

Control plane: The ingress CATS-Router ("CATS-Router 1") receives service routes from the egress CATS-Routers ("CATS-Router 2/3"), including network and computing indicators. The C-PS determines an associated egress CATS-Router by selecting the most appropriate service site and corresponding network forwarding path based on routing strategies and policies, utilizing collected network and computing metrics. The ingress CATS-Router generates the SRv6 tunnel encapsulation from itself to the egress CATS router based on a calculated and matched SR policy.

Data plane: From the client to the service contact instance, packet processing and handling procedures are generally divided into at least the following successive three phases:

- \* Phase I: A service request carrying a CATS Service ID (CS-ID) needs to be routed to the ingress CATS-Router through the access network. The CS-ID can be carried in multiple methods: 1) placing the CS-ID in the destination address field, where the destination address would be encoded with the CS-ID as specific anycast IPs;

2) placing the CS-ID in an IPv6 extension header, and the destination address would inherit the IP address of the ingress CATS-Router.

- \* Phase II: When packets of a service request are received by an ingress CATS-Router, it is classified and determined by the C-TC component. When a matching entry of service routes is found for this request, the ingress CATS-Router encapsulates it and forwards the packets to the C-PS selected egress CATS-Router via the matching SR tunnel.
- \* Phase III: At egress CATS-Router, the packets are decapsulated and successively forwarded according to local entries, and the packets are sent to a corresponding selected service contact instance.

5. Solution 1: Service ID carried in an anycast IP with bidirectional address translation mode

```

+-----+
+-----++-----++-----++ IP1+--+
| Client | | IngressCATS-Router | | EgressCATS-Router | +-+ IP2 |
+-----++-----++-----++-----+
| | | |
| +-----+ | |
| DATA1 | | |
| +-----+ | |
| innerheader | | |
| +-----+ | |
| SA=clientIP | | |
+-----+ | | +-----+ | |
| DATA1 | | | DA=Sid1 | | |
+-----+ | | +-----+ | |
| innerheader | | +-----+ | |
+-----+ | | | outerheader | | +-----+ |
| SA=clientIP | | | +-----+ | | DATA1 | |
+-----+ | | | SRH | | | +-----+ |
| DA=Sid1 | | | +-----+ | | innerheader | |
+-----+ | | | SA=C-R-I | | | +-----+ | |
+-----+ | | | +-----+ | | | SA=clientIP | | |
| | DA=C-R-E | | | +-----+ | |
| +-----+ | | | DA=IP1 | | |
| +-----+ | | | +-----+ | |
+-----> | | +-----+ |
Phase1+-----> | |
Phase2+-----> |
Phase3 |
| | |

```

Figure 2: CATS Dataplane Workflow for solution 1

Figure 2 illustrates successive phases of workflow in Solution 1

- \* Phase I: The packet of a service request carries a CS-ID in its destination address field. The destination address is encoded as an anycast IP address and would be routable within the access network. This prerequisite requires that anycast prefix routes are distributed throughout the access network. Service packets can be forwarded to the ingress CATS-Router for processing in a successive Phase II.
- \* Phase II: The ingress CATS-Router looks up with the corresponding service routing table indexed by the service ID in the destination address of the packets, determines appropriate service route entries and routes the packet to the SR policy by encapsulating the SRH tunnel header, and forwards the packet to the egress CATS-Router for afterwards procedures in phase III.
- \* Phase III: The egress CATS-Router decapsulates the SRH tunnel encapsulation of the packets. Since there might be multiple service instances which provide the same service deployed under the same CATS-Router, the packets carrying anycast IP addresses cannot be directly routed to a corresponding service contact instance. NAT (Network Address Translation) needs to be performed for the packets, and the packets are forwarded to the corresponding service contact instance by the lookup among service routes entries and a corresponding translation of destination address.

Anycast IP is used as the destination address of the end-to-end session. Since the destination address of the user packet is translated to the IP address of the service contact instance in phase III. After the service contact instance receives the packet, the service contact instance correspondingly utilizes the incoming source address as the destination address of the response packet, and uses the IP address of the service contact instance as the source address. To eliminate influences on the host protocol stack for service contact and session establishment, the source address of the response packet must be translated to the corresponding upstream destination address, for which an SNAT process should be performed for the response packets in a downstream workflow.

Specifically, the NAT (Network Address Translation) function can be provided by either the egress CATS-Router or the ingress CATS-Router. In the case of the egress CATS-Router, a special SID (Segment ID) needs to be extended to indicate the NAT translation. However, for the ingress CATS-Router, no such special SID is required; it can

determine the need for NAT based on the context. This allows for flexibility in choosing different solutions based on actual circumstances.

6. Solution 2: Service ID carried in the IPv6 EH with unidirectional address translation mode

Among up-to-date application scenarios, some newly introduced transport protocols would support the change of connecting IP address without interrupting the traffic flows and disconnecting the connection session. In these cases, the CATS-Routers would modify the destination address of the corresponding packets to the IP address of the selected service contact instance when the client sends its upstream packet, and establish a connection through the downstream response packet from the service instance even the IP address is modified. Corresponding capable transport protocols are outside the scope of this document.

```

+-----+
+-----+ +-----+ +-----+ | IP1 +--+
| Client | | IngressCATS-Router | | EgressCATS-Router | +--+ IP2 |
+-----+ +-----+ +-----+ +-----+
| | | |
| +-----+ | |
| DATA1 | | |
| +-----+ | |
| innerheader | | |
| +-----+ | |
| SA=clientIP | | |
+-----+ | | +-----+ | | |
| DATA1 | | | DA=Sid1 | | | |
+-----+ | | +-----+ | | |
| innerheader | | +-----+ | | | | | |
| +-----+ | | | outerheader | | +-----+ |
| SA=clientIP | | | +-----+ | | | DATA1 | |
+-----+ | | | SRH | | | +-----+ |
| DA=Sid1 | | | +-----+ | | | innerheader | |
+-----+ | | | SA=C-R-I | | | +-----+ | |
+-----+ | | | +-----+ | | | SA=clientIP | | |
| | DA=C-R-E | | | +-----+ | |
| +-----+ | | | DA=IP1 | | |
+-----+ | | +-----+ | |
+-----+ -> | | +-----+ |
Phase1 +-----+ -> | |
Phase2 +-----+ -> |
Phase3 |
| |

```

Figure 3: CATS Dataplane Workflow for solution 2

Figure 3 illustrates successive phases of workflow in Solution 2

- \* Phase I: Forward packets to the ingress CATS-Router, the destination address of the packet might be set to the ingress CATS-Router's IP address, and the CS-ID is carried in the IPv6 extension header. The packet is then forwarded through the access network to the ingress CATS-Router, at which it is processed in phase II.
- \* Phase II: The ingress CATS-Router looks up with the corresponding service routing table indexed by the service ID in the IPv6 extension header of the packets, determines appropriate service route entries and routes the packet to the SR policy by encapsulating the SRH tunnel header, and forwards the packet to the egress CATS-Router for afterwards procedures in phase III.
- \* Phase III: The egress CATS-Router decapsulates the SRH tunnel encapsulation of the packets. The destination address in the packet needs to be replaced with the IP address of the corresponding service contact instance. The packet is then forwarded to the corresponding service contact instance.

To adapt to the modification of the destination address on the terminal and service host side, the protocol stack should be correspondingly modified and upgraded. To minimize the changes, a new protocol header can be added in the extension header, which will handle the address changes. As a result, downstream packets do not require Source Network Address Translation (SNAT) translation. Under the above conditions, there is only a unidirectional address translation process in Solution 2. In this case, the CATS-Router does not even need to maintain and manage session states for traffic flows, including unidirectional translation entries, etc. This reduces the complexity of the router but increases the workload on the terminal-side protocol.

7. Solution 3: Service ID carried in an anycast IP with TUNNEL/MAC mode



Figure 4: CATS Dataplane Workflow For solution 3

Figure 4 illustrates successive phases of workflow in Solution 3

- \* Phase I: The packet of a service request carries a CS-ID in its destination address field. The destination address is encoded as an anycast IP address and would be routable within the access network. This prerequisite requires that anycast prefix routes are distributed throughout the access network. Service packets can be forwarded to the ingress CATS-Router for processing in a successive Phase II.
- \* Phase II: The ingress CATS-Router looks up with the corresponding service routing table indexed by the service ID in the destination address of the packets, determines appropriate service route entries and routes the packet to the SR policy by encapsulating the SRH tunnel header, and forwards the packet to the egress CATS-Router for afterwards procedures in phase III.
- \* Phase III: The egress CATS-Router decapsulates the SRH tunnel encapsulation of the packets. Since there might be multiple service instances which provide the same service deployed under the same CATS-Router, the packets carrying anycast IP addresses cannot be directly routed to a corresponding service contact instance. Two optional methods can be applied to forward service packets to the service contact instance: 1) Based on the IP addresses of the CATS-Router and service instance, a bidirectional tunnel (such as GRE and GIF) would be directly established between them for forwarding packets to the service instance. This method can be capable for both L2/L3 network; 2) With the learned ARPs in accordance with the service instance IP addresses, and utilizes the corresponding ARP MAC as the destination MAC addresses of user packets, and deliver the packets to the service instance through layer-2 switching. This method can only be capable for the L2 network.

It should be noted that the client and service host stacks of this solution are not modified, and there is no IP address translation process in the above solution. However, the service instance needs to support a mentioned tunneling model, and some protocol stacks might not support the tunneling functionality.

## 8. Solution Comparison Analysis

	Solution 1	Solution 2	Solution 3
CATS router requirement	High	Middle	Middle
Client requirement	None	Middle	None
Service host requirement	None	Middle	Low

Forwarding Performance	Low	High	High
------------------------	-----	------	------

Table 1: CATS Dataplane Comprehensive Comparison of Solutions

As illustrated in Table 1, different solutions have disparate requirements for clients, service hosts, and CATS-Routers, ultimately resulting in different forwarding performances. Generally, solution 1 has the lowest requirements for terminals and service hosts, yet its forwarding performance may be the worst. Solution 2 has the most strict requirements for terminals and service hosts. In most cases, protocol stack modification is applicable to new host protocol stacks. Solution 3 requires the service host's anycast IP to be configured and deployed, and a protocol stack to support tunnels. Both solution 2 and solution 3 can provide satisfying forwarding performance. Additionally, in solution 2 and 3, CATS-Routers are not required to support the full and standard functionality of NAT.

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

- [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]   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>>.
- [RFC8986]   Filsfils, 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>>.

## 12.2.   Informative References

- [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](https://datatracker.ietf.org/doc/html/draft-huang-service-aware-network-framework-01)>.
- [I-D.ietf-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-ietf-  
cats-framework-20, 26 February 2026,  
<[https://datatracker.ietf.org/doc/html/draft-ietf-cats-  
framework-20](https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-20)>.
- [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-14,  
2 February 2026, <[https://datatracker.ietf.org/doc/html/  
draft-ietf-cats-usecases-requirements-14](https://datatracker.ietf.org/doc/html/draft-ietf-cats-usecases-requirements-14)>.
- [I-D.lbdd-cats-dp-sr]  
Li, C., Du, Z., Drake, J., and shangyuxiang, "Computing-  
Aware Traffic Steering (CATS) Using Segment Routing", Work  
in Progress, Internet-Draft, draft-lbdd-cats-dp-sr-06, 13  
October 2025, <[https://datatracker.ietf.org/doc/html/  
draft-lbdd-cats-dp-sr-06](https://datatracker.ietf.org/doc/html/draft-lbdd-cats-dp-sr-06)>.

[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](https://datatracker.ietf.org/doc/html/draft-li-dyncast-architecture-08)>.

[RFC1631] Egevang, K. and P. Francis, "The IP Network Address  
Translator (NAT)", RFC 1631, DOI 10.17487/RFC1631, May  
1994, <<https://www.rfc-editor.org/info/rfc1631>>.

[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](mailto:fu.huakai@zte.com.cn)

Bo Liu  
China Mobile  
Beijing  
China  
Email: [liubo@chinamobile.com](mailto:liubo@chinamobile.com)

Zhenqiang Li  
China Mobile  
Beijing  
China  
Email: [lizhenqiang@chinamobile.com](mailto:lizhenqiang@chinamobile.com)

Daniel Huang  
ZTE Corporation  
Nanjing  
China  
Email: [huang.guangping@zte.com.cn](mailto:huang.guangping@zte.com.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