

CATS  
Internet-Draft  
Intended status: Informational  
Expires: 4 September 2025

Z. Du  
K. Yao  
China Mobile  
G. Huang  
ZTE  
Z. Fu  
New H3C Technologies  
3 March 2025

Default Policy and Related Metrics in CATS  
draft-du-cats-default-policy-and-metrics-00

Abstract

This document describes the considerations and requirements of the computing information that needs to be notified into the network in Computing-Aware Traffic Steering (CATS). Especially, it suggests that a default policy should be supported on the Ingress, and one or more default metrics should be included in the notification.

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 2025.

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. Definition of Terms . . . . .	3
3. General Procedure of Computing-Aware Traffic Steering . . . . .	3
4. Requirements of Computing Resource Modeling . . . . .	4
5. Design principle of Computing Metrics . . . . .	5
6. Computing Metric Considerations in CATS . . . . .	6
7. Default Policy Discussion On Decision Point . . . . .	7
8. Security Considerations . . . . .	8
9. IANA Considerations . . . . .	8
10. Acknowledgements . . . . .	9
11. Contributors . . . . .	9
12. Informative References . . . . .	9
Authors' Addresses . . . . .	10

## 1. Introduction

Computing-Aware Traffic Steering (CATS) is proposed to support steering the traffic among different service sites according to both the real-time network and computing resource status as mentioned in [I-D.ietf-cats-usecases-requirements]. It requires the network to be aware of computing resource information and select a service instance based on the joint metric of computing and networking.

The basic idea of CATS is that the network can be notified about the computing information, and change the steering policy accordingly. It is the first question that what should be aware of by the network in the system. There are many computing information in the computing domain, and we need to select one or several of them as the default ones.

This document describes the considerations and requirements of the computing information designation, and suggests a two-stage strategy for the metric definition. Firstly, we can define some default metrics that is necessary for enabling the basic steering mechanism in CATS. Afterwards, other metrics can be added as needed, which requires that the metric designation should be extensible.

The following paragraphs are organized as below. We firstly review the procedure of traffic steering in CATS, and show where the computing information is needed. After that, we analyses some requirements and the principles of computing metric designation. Then, we describe some computing metric considerations in CATS. Finally, we discuss the default policy on the decision point.

## 2. Definition of Terms

This document makes use of the following terms:

**Computing-Aware Traffic Steering (CATS):** A traffic engineering approach [I-D.ietf-teas-rfc3272bis] that takes into account the dynamic nature of computing resources and network state to optimize service-specific traffic forwarding towards a given service contact instance. Various relevant metrics may be used to enforce such computing-aware traffic steering policies.

**Service:** An offering that is made available by a provider by orchestrating a set of resources (networking, compute, storage, etc.).

**Service instance:** An instance of running resources according to a given service logic.

**Service identifier:** Used to uniquely identify a service, at the same time identifying the whole set of service instances that each represents the same service behavior, no matter where those service instances are running.

**Computing Capability:** The ability of nodes with computing resource achieve specific result output through data processing, including but not limited to computing, communication, memory and storage capability.

## 3. General Procedure of Computing-Aware Traffic Steering

It is assumed that the same service can be provided in multiple places in the CATS framework. In the different service instances, it is common that they have different kinds of computing resources, and different utilization rate of the computing resources.

In the CATS framework, the decision point, which is supposed to be a node in the network, should be aware of the network status and the computing status, and accordingly choose a proper service point for the client.

A general procedure to steer the CATS traffic is described as below. The CATS packets have an anycast destination address, which is also called the service ID, and it is announced by the different service points.

Firstly, the service points need to collect some specific computing information that need to be sent into the network following a uniform format so that the decision point can understand the computing information. In this step, only necessary computing information needs to be considered, so as to avoid exposing too much information of the service points.

Secondly, the service points send the computing information into the network by some means, and update it periodic or on demand.

Thirdly, the decision point receives the computing information, and makes a decision for the specific service related to the service ID. Hence, the route for the service ID on the Ingress is established or updated.

Fourthly, the traffic for the service ID reaching the Ingress node would be identified and steered according to the route policy in the previous step.

Fifthly, the Egress node receives the packets of the traffic for the service ID from the Ingress, and traffic would be identified and forwarded to the corresponding service point.

The scheduling of traffic in CATS is similar to the task allocation and scheduling in cloud computing. Normally, it is a multi-objective optimization. In task allocation strategies of cloud computing systems, the main optimization objectives considered are task execution time, execution cost, and load balancing. Execution time and execution cost are used to evaluate the user satisfaction, and load balancing is used to measure the reliability and availability of the system.

#### 4. Requirements of Computing Resource Modeling

After the computing metrics are generated, they are notified to the decision point in the network to influence the traffic steering. However, the decision point in the network, for example the Ingress Node, only cares about how to use to capability values to do the traffic steering, but does not care about the way how the capability values are generated.

From the aspect of services, they need an evaluating system to generate one or more capability values. To achieve the best LB result, different services or service types may have different ways to evaluate the capability.

From the aspect the decision point in the network, it only needs to understand the way to use the values, and trigger the implement of the related policy.

Some requirements related to the design of the computing resource modeling are listed below.

1. The optimization objective of the policy in the decision point may be various. For example, it may be the lowest latency of the sum of the network delay and the computing delay, or it may be an overall better load balance result, in which the clients would prefer the service points that could support more clients.
2. The update frequency of the computing metrics may be various. Some of the metrics may be more dynamic, and some are relatively static.
3. The notification ways of the computing metrics may be various. According to its update frequency, we may choose different ways to update the metric.
4. Metric merging process should be supported when multiple service instances are behind the same Egress.

## 5. Design principle of Computing Metrics

The target in CATS mainly concerns about the service point selection and traffic steering in Layer3, in which we do not need all computing information of the service points. Hence, we can start with simple cases in the work of the computing resource modeling in CATS. Some design principles can be considered.

1. **Simplicity:** The computing metrics in CATS SHOULD be few and simple, so as to avoid exposing too much information of the service points.
2. **Scalability:** The computing metrics in CATS SHOULD be evolveable for the future extensions.
3. **Interoperability:** The computing metrics in CATS SHOULD be vendor-independent, and OS-independent.

4. Stability: The computing metrics in CATS SHOULD NOT incur too much overhead in protocol design, and it can be stabilized to be used.
5. Accuracy: The computing metrics in CATS SHOULD be effective for path selection decision making, and the accuracy SHOULD be guaranteed.

## 6. Computing Metric Considerations in CATS

Various metrics can be considered in CATS, and perhaps different services would need different metrics. However, we can start with simple cases.

In CATS, a straightforward intent is to minimal the total delay in the network domain and the computing domain. Thus, we can have a start point for the metric designation in CATS considering only the delay information. In this case, the decision point can collect the network delay and the computing delay, and make a decision about the optimal service point accordingly. The advantage of this method is that it is simple and easy to start; meanwhile, the network metric and the computing metric have the same unit of measure. The network delay can be the latency between the Ingress node and Egress node in the network. The computing delay can be generated by the server, which is the delay that server in the service point to process the CATS service. It means "the estimate of the duration of my processing of request", and it is usually an average value for the service request. The optimization objective of traffic steering in this scenario is the minimal total delay for the client.

Another metric that can be considered is the service capability, which is the ability that the server in the service point to process the CATS service. For example, one server can support 100 simultaneous sessions and another can support 10,000 simultaneous sessions. The value can be generated by the server when deploying the service instance. The metric can work alone. In this scenario, the decision point can do a Load Balance job according to the service capability. For example, the decision process can be load balancing after pruning the service points with poor network latency metrics. Also, the metric can work with the computing delay metric. For example, in this scenario, we can prune the service points with poor total latency metrics before the load balancing.

In future, we can also consider other metrics, which may be more dynamic. Besides, for some other optimization objectives, we can consider other metrics, even metrics about energy consumption.

However, in this cases, the decision point needs to consider more dimensions of metrics. A suggestion is that we should firstly make sure the service point is available, which means the service point can still accept more sessions, and then select an optimal target service point according to the optimization objective.

## 7. Default Policy Discussion On Decision Point

In this section, for the convenience of description, we assume that the decision point is on the Ingress, and the Egress would gather the computing metrics of the corresponding service points and notify the Ingress. After receiving the notification about the computing metrics, the Ingress can generate one or more routing policies accordingly for forwarding the packets of CATS traffic from the user. In this document, we suggest that the routing policies should include a default one at least. The packets of CATS traffic received by the Ingress should have an anycast IP as the destination address, which will trigger the routing policy pre-configured.

To enable the basic cooperation in CATS, we need one or a set of default computing metrics to be notified into the network. All the CATS Ingresses need to understand the default metrics and trigger the same or similar operations inside the router. It is to say that Ingress should enable some default routing policy after receiving the default metrics. The detailed procedures inside the Ingresses are vendor-specific, for example, which default policy is enabled for a specific service on the router.

By comparison, other metrics would be optionally, although perhaps they can obtain a better or more preferred LB result than the default ones. If the Ingress receives the additional metrics and can understand them, it can use the optional metrics to update the forwarding policy for the routes of the anycast IP.

There are two kinds of forwarding treatments on the Ingress. Although they are implementations inside the equipment, we give a general description about them here, because they are related to the default metric selection.

The first one is that the Ingress will deploy several routes for the anycast IP, but among them only one is active, and others are for backup and are set to inactive. The second one is that the Ingress can have multiple active routes for the anycast IP, and each route has a dedicated weight, so that a load balancing can be done within the Ingress.

The advantage of the first one is that it can select a best service instance for the client according to the network and computing status. However, its disadvantage is that the Ingress will forward all the traffic of the new clients to a single service point before the policy is updated, which will potentially cause the service point to become busy. For the second one, it may achieve a better LB result.

If the first one is used for a specific service, the routing policy on the Ingress may be the minimum total delay first. In this case, the Ingress needs to collect the network delay and the computing delay, and add them together to select the best Egress. If the second one is used for a specific service, the routing policy on the Ingress may be load balancing based on service capability.

An initial proposal of the default metrics for the default policies is that we can always send the two metrics mentioned in the last paragraph, i.e., the computing delay and the service capability. At least one of them should be valid. The bits of the computing delay or the service capability are set to all "zero" will be considered invalid, and other values are considered valid. Meanwhile, the bits of the computing delay or the service capability are set to all "one" stands for the service point is temporary busy, and the Ingress should not send new clients to that service point.

Alternatively, we can also add another simple metric to indicate the busy or not status. However, this metric is relatively more dynamic than the former two. If the Ingress receives the busy indicator from an Egress, it can change the routing policy accordingly. For example, it can change the path to the related Egress into an inactive one, or change the weight of the path to the related Egress into zero. After the change, the path to the related Egress will not be used for forwarding the CATS traffic from any new clients, i.e., it is deleted from the routing policies temporarily. After a certain time, which can be pre-configured, or receiving a new busy or not indication, the Ingress would update the status or the weight of the path to the related Egress. Thus, the path to the related Egress will be added into the routing policies again.

## 8. Security Considerations

TBD.

## 9. IANA Considerations

TBD.

## 10. Acknowledgements

The author would like to thank Adrian Farrel, Joel Halpern, Tony Li, Thomas Fossati, Dirk Trossen, Linda Dunbar for their valuable suggestions to this document.

## 11. Contributors

The following people have substantially contributed to this document:

Yuxia Fu  
China Mobile  
fuyuxia@chinamobile.com

Jing Wang  
China Mobile  
wangjingjc@chinamobile.com

Peng Liu  
China Mobile  
liupengyjj@chinamobile.com

Wenjing Li  
Beijing University of Posts and Telecommunications  
wjli@bupt.edu.cn

Lanlan Rui  
Beijing University of Posts and Telecommunications  
llrui@bupt.edu.cn

## 12. Informative References

### [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-06, 14 February 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-cats-usecases-requirements-06>>.

### [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>>.

## Authors' Addresses

Zongpeng Du  
China Mobile  
No.32 XuanWuMen West Street  
Beijing  
100053  
China  
Email: [duzongpeng@foxmail.com](mailto:duzongpeng@foxmail.com)

Kehan Yao  
China Mobile  
No.32 XuanWuMen West Street  
Beijing  
100053  
China  
Email: [yaokehan@chinamobile.com](mailto:yaokehan@chinamobile.com)

Guangping Huang  
ZTE  
Email: [huang.guangping@zte.com.cn](mailto:huang.guangping@zte.com.cn)

Zhihua Fu  
New H3C Technologies  
Email: [fuzhihua@h3c.com](mailto:fuzhihua@h3c.com)