

bmwg  
Internet-Draft  
Intended status: Informational  
Expires: 11 December 2025

K. Yao  
P. Liu  
China Mobile  
9 June 2025

Benchmarking Methodology for Computing-aware Traffic Steering  
draft-yl-bmwg-cats-00

Abstract

Computing-aware traffic steering(CATS) is a traffic engineering approach based on the awareness of both computing and network information. This document proposes benchmarking methodologies for CATS.

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 11 December 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. Test Methodology . . . . .	3
3.1. Test Setup . . . . .	3
3.1.1. Test Setup - Centralized Approach . . . . .	3
3.1.2. Test Setup - Distributed Approach . . . . .	5
3.2. Control Plane and Forwarding Plane Support . . . . .	6
3.3. Topology . . . . .	6
3.4. Device Configuration . . . . .	7
4. Reporting Format . . . . .	7
5. Benchmarking Tests . . . . .	8
5.1. CATS Metrics Collection and Distribution . . . . .	8
5.2. Session continuity . . . . .	8
5.3. Latency . . . . .	8
5.4. Sytem Utilization . . . . .	9
6. Security Considerations . . . . .	9
7. IANA Considerations . . . . .	9
8. Acknowledgements . . . . .	9
9. References . . . . .	9
9.1. Normative References . . . . .	9
9.2. Informative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

Computing-aware traffic Steering(CATS) is a traffic engineering approach considering both computing and network metrics, in order to select appropriate service instances. Some of the latency-sensitive, throughput-sensitive applications or compute-intensive applications need CATS to guarantee effective instance selection, which are mentioned in [I-D.ietf-cats-usecases-requirements]. There is also a general CATS framework [I-D.ietf-cats-framework] for implementation guidance. However, considering there are many computing and network metrics that can be selected for traffic steering, as proposed in [I-D.ietf-cats-metric-definition], some benchmarking test methods are required to validate the effectiveness of different CATS metrics. Besides, there are also different deployment approaches, i.e. the distributed approach and the centralized approach, and there are also multiple objectives for instance selection, for example, instance with lowest end-to-end latency or the highest system utilization. The benchmarking methodology proposed in this document is essential for guiding CATS implementation.

## 2. Definition of Terms

This document uses the following terms defined in [I-D.ietf-cats-framework]: CATS: Computing-aware Traffic Steering  
C-PS: CATS path-selection

This document further defines:

CATS Router: Router that supports CATS mechanisms for traffic engineering. ECMP: Equal cost multi-path routing

## 3. Test Methodology

### 3.1. Test Setup

The test setup in general is compliant with [RFC2544]. As is mentioned in the introduction, there are basically two approaches for CATS deployment. The centralized approach and the distributed approach. The difference primarily sits in how CATS metrics are collected and distributed into the network and accordingly, where the CATS path selector(C-PS) is placed to make decisions, as is defined in [I-D.ietf-cats-framework].

#### 3.1.1. Test Setup - Centralized Approach

Figure 1 shows the test setup of the centralized approach to implement CATS. The centralized test setup is similar to the Software Defined Networking(SDN) standalone mode test setup defined in [RFC8456]. The DUT locates at the same place with the SDN controller. In the centralized approach, SDN controller takes both the roles of CATS metrics collection and the decision making for instance selection as well as traffic steering. The SDN controller is connected with application plane via interface 2(I2), and is connected to Edge server manager via interface 4(I4). The Southbound interface(I1) of the SDN controller is connected with the forwarding plane. Service request is sent from application to the SDN controller through I2. CATS metrics are collected from Edge server manager via I4. The traffic steering policies are configured through I1. In the forwarding plane, CATS router 1 serves as the ingress node and is connected with the host which is an application plane emulator. CATS router 2 and CATS router 3 serve as the egress nodes and are connected with two edge servers respectively. Both of the edge servers are connected with edge server manager via I3. I3 is an internal interface for CATS metrics collection within edge sites.

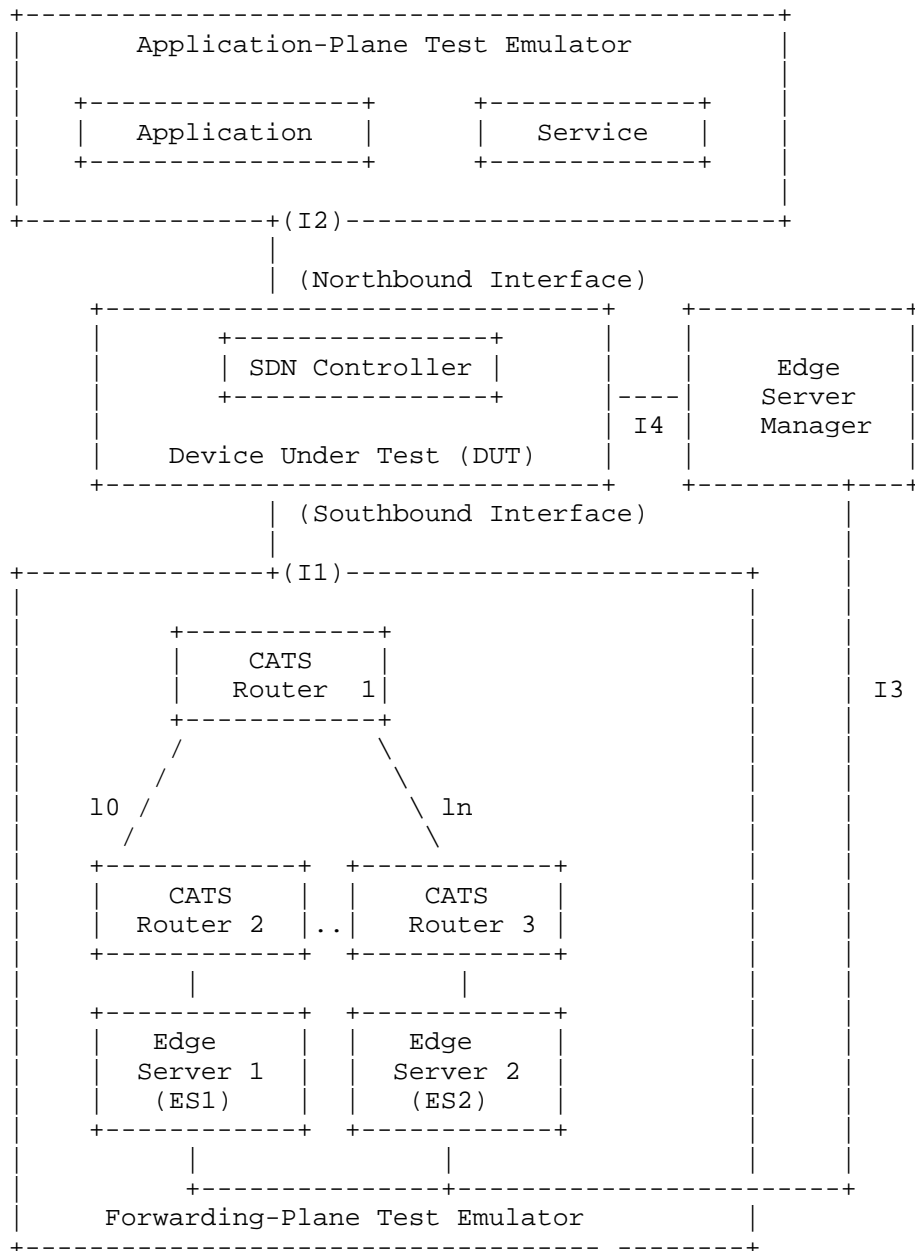


Figure 1: Centralized Test Setup

### 3.1.2. Test Setup - Distributed Approach

Figure 2 shows the test setup of the distributed approach to implement CATS. In the distributed test setup, The DUT is the group of CATS routers, since the decision maker is the CATS ingress node, namely CATS router 1. CATS egress nodes, CATS router 2 and 3, take the role of collecting CATS metrics from edge servers and distribute these metrics towards other CATS routers. Service emulators from application plane is connected with the control-plane and forwarding-plane test emulator through the interface 1.

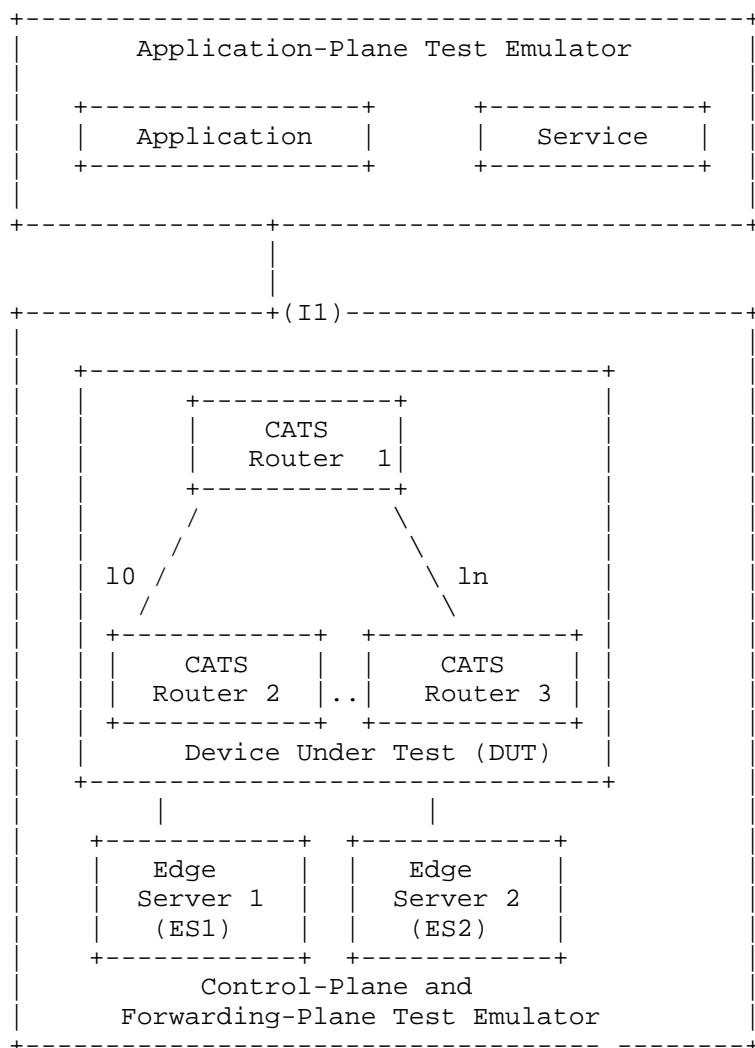


Figure 2: Distributed Test Setup

### 3.2. Control Plane and Forwarding Plane Support

In the centralized approach, Both of the control plane and forwarding plane follow Segment Routing pattern, i.e. SRv6[RFC8986]. The SDN controller configure SRv6 policies based on the awareness of CATS metrics and traffic is steered through SRv6 tunnels built between CATS ingress nodes and CATS egress nodes. The collection of CATS metrics in control plane is through Restful API built between the SDN controller and the edge server manager. In the distributed approach, In terms of the control plane, EBGP[RFC4271] is established between CATS egress nodes and edge servers. And IBGP[RFC4271] is established between CATS egress nodes with CATS ingress nodes. BGP is chosen to distribute CATS metrics in network domain, from edge servers to CATS ingress node. Carrying CATS metrics is implemented through the extension of BGP, following the definition of [I-D.ietf-idr-5g-edge-service-metadata]. Some examples for defining sub-TLVs are like:

- \* Delay sub-TLV: The processing delay within edge sites and the transmission delay in the network.
- \* Site Preference sub-TLV: The priority of edge sites.
- \* Load sub-TLV: The available compute capability of each edge site.

Other sub-TLVs and can be gradually defined according to the CATS metrics agreement defined in [I-D.ietf-cats-metric-definition].

In terms of the forwarding plane, SRv6 tunnels are enabled between CATS ingress nodes with CATS egress nodes. Service flows are routed towards service instances by following anycast IP addresses in both of the approaches.

### 3.3. Topology

In terms of both of the approaches to test CATS performance in laboratory environments, implementors consider only single domain realization, that is all CATS routers are within the same AS. There is no further special requirement for specific topologies.

### 3.4. Device Configuration

Before implementation, there are some pre-configurations need to be settled. Firstly, in both of the approaches, application plane functionalities must be settled. CATS services must be setup in edge servers before the implementation, and hosts that send service requests must also be setup.

Secondly, it comes to the CATS metrics collector setup. In the centralized approach, the CATS metrics collector need to be first setup in the edge server manager. A typical example of the collector can be the monitoring components of Kubernetes. It can periodically collect different levels of CATS metrics. Then the connection between the edge server manager and the SDN controller must be established, one example is to set restful API for CATS metrics publication and subscription. In the distributed approach, the CATS metrics collector need to be setup in each edge site. In this benchmark test, the collector is setup in each edge server which is directly connected with a CATS egress node. Implementors can use plugin software to collect CATS metrics. Then each edge server must set BGP peer with the CATS egress node that's directly connected. In each edge server, a BGP speaker is setup.

Thirdly, The control plane and forwarding plane functions must be pre-configured. In the centralized approach, the SDN controller need to be pre-configured and the interface between the SDN controller and CATS routers must be tested to validate if control plane policies can be correctly downloaded and it metrics from network side can be correctly uploaded. In the distributed approach, the control plane setup is the iBGP connections between CATS routers. For both the approaches. the forwarding plane functions, SRv6 tunnels must be pre-established and tested.

## 4. Reporting Format

The benchmarking test focuses data that can be measured and controllable.

- \* Hardware and software versions of CATS routers, edge servers, and the SDN controller.

- \* Three levels of CATS metrics:

For L0, the benchmarking tests include resource-related metrics like CPU utilization, memory utilization, throughput, delay, and service-related metrics like Queries per second(QPS). For L1 and L2 metrics, the benchmarking tests include all normalized metrics.

## 5. Benchmarking Tests

### 5.1. CATS Metrics Collection and Distribution

- \* Objective: To determine that CATS metrics can be correctly collected and distributed to the DUTs which are the SDN controller in the centralized approach and the CATS ingress node in the distributed approach.

- \* Procedure:

In the centralized approach, the edge server manager periodically grasp CATS metrics from every edge server that can provide CATS service. Then it passes the information to the SDN controller through publish-subscription methods. Implementors then should log into the SDN controller to check if it can receive the CATS metrics from the edge server manager. In the distributed approach, the collectors within each edge server periodically grasp the CATS metrics of the edge server. Then it distributes the metrics to the CATS egress node it directly connected. Then Each CATS egress node further distributes the metrics to the CATS ingress node. Implementors then log into the CATS ingress node to check if metrics from all edge servers have been received.

### 5.2. Session continuity

- \* Objective: To determine that traffic can be correctly steered to the selected service instances and TCP sessions are maintained for specific service flows.
- \* Procedure: Enable several hosts to send service requests. In distributed approach, log into the CATS ingress node to check the forwarding table that route entries have been created for service instances. Implementors can see that a specific packet which hits the session table, is matched to a target service instance. Then manually increasing the load of the target edge server. From the host side, one can see that service is going normally, while in the interface of the CATS router, one can see that the previous session table aging successfully which means CATS has steer the service traffic to another service instance. In the centralized approach, implementors log into the management interface of the SDN controller and can check routes and sessions.

### 5.3. Latency

- \* Objective: To determine that CATS works properly under the pre-defined test condition and prove its effectiveness in service end-to-end latency guarantee.



- \* Procedure: Pre-define the CATS metrics distribution time to be  $T_1$  seconds. Enable a host to send service requests. In distributed approach, log into the CATS ingress node to check if route entries have been successfully created. Suppose the current selected edge server is ES1. Then manually increase the load of ES1, and check the CATS ingress node again. The selected instance has been changed to ES2. CATS works properly. Then print the logs of the CATS ingress router to check the time it update the route entries. The time difference  $\Delta T$  between when the new route entry first appears and when the previous route entry last appears should equals to  $T_1$ . Then check if service SLA can be satisfied. In the centralized approach, implementors log into the management interface of the SDN controller and can check routes and sessions.

#### 5.4. Sytem Utilization

- \* Objective: To determine that CATS can have better load balancing effect at server side than simple network load balancing mechanism, for example, ECMP.
- \* Procedure: Enable several hosts to send service requests and enable ECMP at network side. Then measure the bias of the CPU utilization among different edge servers in time duration  $\Delta T_2$ . Stop services. Then enable the same number of service requests and enable CATS at network side(the distributed approach and the centralized approach are tested separately.). Measure the bias of the CPU utilization among the same edge servers in time duration  $\Delta T_2$ . Compare the bias value from two test setup.

#### 6. Security Considerations

The benchmarking characterization described in this document is constrained to a controlled environment (as a laboratory) and includes controlled stimuli. The network under benchmarking MUST NOT be connected to production networks. Beyond these, there are no specific security considerations within the scope of this document.

#### 7. IANA Considerations

This document has no IANA actions.

#### 8. Acknowledgements

#### 9. References

##### 9.1. Normative References

- [RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <<https://www.rfc-editor.org/rfc/rfc2544>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/rfc/rfc4271>>.
- [RFC8456] Bhuvaneshwaran, V., Basil, A., Tassinari, M., Manral, V., and S. Banks, "Benchmarking Methodology for Software-Defined Networking (SDN) Controller Performance", RFC 8456, DOI 10.17487/RFC8456, October 2018, <<https://www.rfc-editor.org/rfc/rfc8456>>.
- [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/rfc/rfc8986>>.

## 9.2. Informative References

- [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-07, 30 April 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-07>>.
- [I-D.ietf-cats-metric-definition]  
Yao, K., Shi, H., Li, C., Contreras, L. M., and J. Ros-Giralt, "CATS Metrics Definition", Work in Progress, Internet-Draft, draft-ietf-cats-metric-definition-02, 3 March 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-cats-metric-definition-02>>.
- [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-idr-5g-edge-service-metadata]

Dunbar, L., Majumdar, K., Li, C., Mishra, G. S., and Z.  
Du, "BGP Extension for 5G Edge Service Metadata", Work in  
Progress, Internet-Draft, draft-ietf-idr-5g-edge-service-  
metadata-29, 28 April 2025,  
<[https://datatracker.ietf.org/doc/html/draft-ietf-idr-5g-  
edge-service-metadata-29](https://datatracker.ietf.org/doc/html/draft-ietf-idr-5g-edge-service-metadata-29)>.

#### Authors' Addresses

Kehan Yao  
China Mobile  
Email: [yaokehan@chinamobile.com](mailto:yaokehan@chinamobile.com)

Peng Liu  
China Mobile  
Email: [liupengyjy@chinamobile.com](mailto:liupengyjy@chinamobile.com)