

IDR  
Internet-Draft  
Intended status: Standards Track  
Expires: 5 October 2025

C. Li  
Huawei Technologies  
P. Liu  
China Mobile  
3 April 2025

CATS BGP Extension  
draft-11-idr-cats-bgp-extension-01

## Abstract

CATS (Computing-Aware Traffic Steering) is a traffic engineering approach that takes into account the dynamic nature of computing resources and network state to optimize service-specific traffic forwarding towards a service contact instance. This document defines the control plane BGP extension for CATS (Computing-Aware Traffic Steering).

## 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 5 October 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. Conventions and Definitions . . . . .	3
3. Control Plane Modes . . . . .	4
4. Basic Mode . . . . .	4
4.1. Mandatory Extensions . . . . .	5
4.1.1. Service-Oriented Capability . . . . .	5
4.1.2. Service-Oriented Available Resource . . . . .	6
4.2. Optional Extensions . . . . .	6
4.2.1. Site Preference . . . . .	6
4.2.2. Site Physical Availability Index . . . . .	7
4.2.3. Delay Prediction . . . . .	7
4.2.4. Raw Metadata . . . . .	7
5. Sharing Mode . . . . .	8
5.1. Workflow . . . . .	8
5.2. Extensions . . . . .	9
5.2.1. Option 1: Reusing an Existing Update . . . . .	11
5.2.2. Option 2: Using a Specific Prefix . . . . .	11
5.2.3. Option 3: Using a New NLRI . . . . .	12
6. Deployment Considerations . . . . .	13
7. Manageability Considerations . . . . .	13
8. Security Considerations . . . . .	13
9. IANA Considerations . . . . .	13
10. Normative References . . . . .	13
Appendix A. Acknowledgements . . . . .	14
Contributors . . . . .	14
Authors' Addresses . . . . .	14

## 1. Introduction

When deploying edge clouds, multiple service instances might be deployed in multiple edge sites to provide equivalent function to end users. However, the resource of each edge site is limited due to only limited end users accessing to it locally. When burst traffic is generated by emergent events, the traffic should be balanced to multiple edge sites to provide better services to the end users. In order to provide better traffic steering among edge sites, a framework called CATS (Computing-Aware Traffic Steering)

[I-D.ietf-cats-framework] is proposed.

CATS (Computing-Aware Traffic Steering) [I-D.ietf-cats-framework] is a traffic engineering approach that takes into account the dynamic nature of computing resources and network state to optimize service-specific traffic forwarding. Various relevant metrics are defined in [I-D.ietf-cats-metric-definition], which can be used in CATS.

A general BGP extension of conveying computing metrics are defined in [I-D.ietf-idr-5g-edge-service-metadata]. It proposed a new Path Attribute, called Metadata Path Attribute, and some related sub-TLVs to carry the computing related metadata.

This document defines the BGP extension for CATS control plane. Depending the deployment of CATS, there are two modes for CATS control plane, 1) Basic mode and 2) Sharing mode. For the basic mode, CATS can reuse the BGP extension defined in [I-D.ietf-idr-5g-edge-service-metadata]. For the sharing Mode, this document defines a new mechanism based on the basic extension defined in [I-D.ietf-idr-5g-edge-service-metadata].

## 2. Conventions and Definitions

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.

This document uses terms defined in [I-D.ietf-cats-framework]. Some terms are listed below for clarification.

- \* Computing-Aware Traffic Steering (CATS): An architecture that takes into account the dynamic nature of computing resources and network state to steer service traffic to a service instance. This dynamicity is expressed by means of relevant metrics.
- \* 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.

### 3. Control Plane Modes

Generally, in edge cloud, each service exclusively occupies some computing resources in the edge cloud. Therefore, in the CATS framework, when a service needs to synchronize the computing status of the occupied computing resource to the network, the computing status is transferred in a per-service manner. In other words, each service has its own computing status information release and control plane processing. This mainstream deployment mode is called the basic mode. In this mode, each service exclusively occupies computing resources, causing the price of cloud resources and the CATS scheduling may be high, which is more suitable for large-scale content/service providers such as OTT. However, the price of this deployment mode may be too high for small or medium-sized content/service providers, which prevents a large number of small- and medium-sized content/service providers from choosing CATS services.

In order to allow more small- and medium-sized content/service providers to choose edge computing and CATS scheduling services, this document proposes a new mode, called sharing mode. In sharing mode, multiple services can use the same shared resource and share the CATS scheduling service in the BGP control plane, reducing the price of purchasing edge cloud services and CATS scheduling services. This deployment mode also enables carriers to obtain more service providers.

The detailed BGP extension of basic and sharing modes will be described in the following sections.

### 4. Basic Mode

In the basic mode, each service is deployed independently, including occupying independent computing resources in edge site. The computing resources update is triggered by the service itself and independent with each other, for example, by using an independent BGP update message. In this mode, the computing metrics update and the CATS steering decision is implemented in a service-oriented way.

In order to support 5G services in edge cloud, a general mechanism with BGP extensions is defined in [I-D.ietf-idr-5g-edge-service-metadata]. CATS framework basic mode can reuse the extensions defined in [I-D.ietf-idr-5g-edge-service-metadata]. However, in real implementation, only the basic extensions instead of all the extensions defined are required to be implemented. This document will specify the mandatory part and optional part of extensions in CATS implementation to support easier implementation and interoperability test.

#### 4.1. Mandatory Extensions

This section specifies the mandatory extensions for CATS in implementation.

In CATS, the fundamental features are reporting the computing metrics from the edge site to the network devices, such as the egress router of CATS overlay path, which is connected to the edge site. Therefore the basic and mandatory features are reporting the service related capability and available resource to the network devices. When the egress CATS forwarder [I-D.ietf-cats-framework] receive the computing metrics, it can reuse the Service-Oriented Capability and Service-Oriented Available Resource sub-TLVs to distribute the metrics to the ingress CATS forwarder.

##### 4.1.1. Service-Oriented Capability

Capability information describes the total resources that can be used by a service, which is important in selecting the best service contact instance, so the distribution of this information is mandatory.

This document reuses the Service-Oriented Capability sub-TLV in Metadata Path Attribute [I-D.ietf-idr-5g-edge-service-metadata] to distribute the capability information from an egress CATS forwarder to an ingress CATS forwarder. The format of Service-Oriented Capability sub-TLV is shown below for reference.

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| ServiceOriented Cap Sub-Type | Length | Res | MT |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               SO-CapValue                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 1: Service-Oriented Capability sub-TLV

The received capability information of a service is encoded by the egress CATS forwarder into the SO-CapValue field, and the sub-TLV will be encoded into the Metadata Path Attribute. The whole mechanism of encoding and processing the capability reuses the mechanism defined in [I-D.ietf-idr-5g-edge-service-metadata].

#### 4.1.2. Service-Oriented Available Resource

Service-Oriented Available Resource sub-TLV

[I-D.ietf-idr-5g-edge-service-metadata] is defined for distributing the real-time available resource of a service. The real-time available resource is vital for some low-latency service/applications which need the dynamic resource status to achieve on-time load balancing. Therefore, the distribution of available resource for a service is mandatory in CATS implementation.

This document reuses the Service-Oriented Available Resource sub-TLV in Metadata Path Attribute [I-D.ietf-idr-5g-edge-service-metadata] to distribute the available resource information from an egress CATS forwarder to an ingress CATS forwarder. The format of Service-Oriented Available Resource sub-TLV is shown below for reference.

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|ServiceOriented Avail Sub-Type |   Length   |P| Res |  MT  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SO-AvailRes                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 2: Service-Oriented Available Resource sub-TLV

The received available resource information of a service is encoded by the egress CATS forwarder into the SO-AvailRes field, and the sub-TLV will be encoded into the Metadata Path Attribute. The whole mechanism of encoding and processing the capability reuses the mechanism defined in [I-D.ietf-idr-5g-edge-service-metadata].

The above two sub-TLVs and the Metadata Path Attribute are required in CATS implementation.

#### 4.2. Optional Extensions

In order to support some enhanced features, the following optional extension can be implemented when implementing CATS.

##### 4.2.1. Site Preference

Site preference indicates the preference of each site when comparing service contact instances from different site. The preference value may be generated from several factors, such as the price of energy, renting fee of DC rack, etc. However, this is a site-level selection instead of service-level. Except the same site-level conditions, such as energy price, other service specific factors may be more important when selecting a better service contact instance from

sites.

In addition, the factors such as energy price can be added as a factor in generating the normalized metric value of Service-Oriented Capability and Available Resource. Therefore, Site Preference is an optional metric in CATS implementation, and the extension and processing refers to the Site Preference Index Sub-TLV defined in [I-D.ietf-idr-5g-edge-service-metadata].

#### 4.2.2. Site Physical Availability Index

Similar to Site Preference, Site Physical Availability Index is a site-level metric, which is useful in batch processing of BGP updates. This is an enhancement of the CATS implementation, which can be implemented optionally.

#### 4.2.3. Delay Prediction

The end-2-end delay information is vital for time-sensitive service, such as 5G uRLLC services. However, it is not easy to detect the real-time delay in deployment. As per [I-D.ietf-idr-5g-edge-service-metadata], Service Delay Prediction Index sub-TLV is defined for encoding the delay information. It includes two types of delay formats, one is for delay prediction index, another one is the NTP format of raw delay information. CATS implementation can reuse this mechanism.

Distributing real-time delay information may bring too heavy burden to the BGP control plane, therefore this document defines it as an optional feature in CATS implementation. Furthermore, the delay information can be predicted by some raw metrics, such as the capability of GPU/CPU, which can be considered as factors when generating the normalized Service-Oriented Capability and Service-Oriented Available Resource metrics.

#### 4.2.4. Raw Metadata

Raw metadata is too complicated in standardization and implementation, therefore, this document does not recommend to implement the mechanism of Raw Metadata defined in [I-D.ietf-idr-5g-edge-service-metadata]. An implementation might choose to implement it.

In the basic mode, all the extensions and mechanisms can reuse the extensions and mechanisms defined in [I-D.ietf-idr-5g-edge-service-metadata], therefore, this document will not explain the details of processing any sub-TLVs.

## 5. Sharing Mode

In the Basic mode, each service will occupy its own computing resources in the edge site. But the price of edge clouds might be too high to small- and medium-sized service providers, which may block them to deploy server closer to end users. Sharing Mode is a mode that allows different service providers to share the same computing resources when deploying services in edge sites. Since the computing resources are shared among different services, the price of the computing resources for each service can be reduced.

When multiple services share the same computing resources, all the services will be influenced by the status of the shared computing resources. For example, in the Basic mode, when the available computing resources is running out, a C-SMA (CATS Service Metric Agent) can collect the metrics per-service and distribute them to the egress CATS forwarder. The egress CATS forwarder will redistribute the metrics in a per-service manner to ingress CATS forwarders.

If there are 10 services are sharing the shared computing resources, then the C-SMA should collect 10 times for these 10 services respectively only because of one single metrics update of the shared computing resource. Furthermore, the egress CATS forwarder may generate 10 BGP update messages to distribute the update of the metrics to the ingress CATS forwarder though all the metrics of in the Metadata Path Attribute are the same.

When different service's routes are using the same Path attributes, including the same Metadata Path Attribute, the 10 BGP update messages can be packed in to a single update message. However, when different service routes use different Path Attributes, a single update of the shared computing resources can trigger more than 1 update messages.

Sharing mode provides a new mechanism to address this problem by turning service-oriented metrics update into resource-oriented metrics update or service-group-oriented metric update.

### 5.1. Workflow

The Sharing mode mainly includes two phases of updating the computing resource metrics.

- \* Association establishment
- \* Resource-oriented/Service-group-oriented metrics update



In the first phases, a resource ID that identifies the shared computing resource, or a service group ID that identifies a group of service using a specific shared computing resource, needs to be added into the Service-Oriented Capability sub-TLV and Service-Oriented Available Resource sub-TLV. When the ingress CATS forwarder receives a BGP update message, it can learn the service ID (which is the prefix in the NLRI), and the resource ID or service group ID, and then build an association relation between them.

In the second phase, the egress CATS forwarder may choose a possible method described below to distribute the metrics of the shared computing resources directly instead of sending multiple service oriented independent BGP update. When the ingress CATS forwarder receives the resource metric update, it will trigger all the processing of the associated services. In this way, only one BGP update message is needed for a single update for a shared resource, which can reduce N-1 (N is the number of the services that share the same shared computing resources) BGP update messages in the best case.

5.2. Extensions

This section describes the extensions of Sharing mode.

As mentioned above, in the association establishment phase, a resource ID that identifies the shared computing resource, or a service group ID that identifies a group of service using a specific shared computing resource, needs to be added into the Service-Oriented Capability sub-TLV and Service-Oriented Available Resource sub-TLV. The updated format of Service-Oriented Capability sub-TLV is shown below.

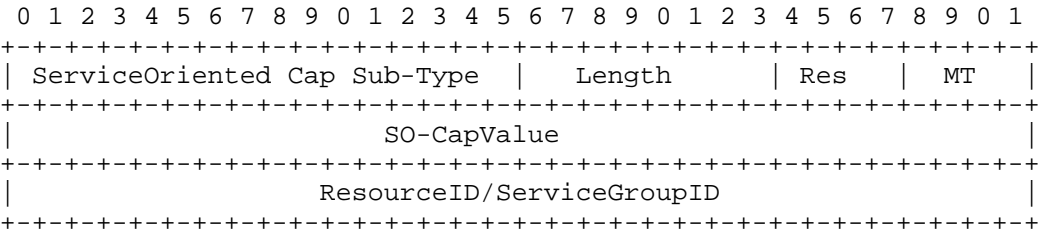


Figure 3: Service-Oriented Capability sub-TLV

Where,

- \* ResourceID (4 bytes): an ID that identifies the shared computing resource consumed by this service. 0 is reserved, meaning no specific shared resource is identified, and it MUST be ignored in processing.
- \* or ServiceGroupID (4 bytes): an ID that identifies a group of service that the service belongs to. All the service within this group use a specific shared computing resource. 0 is reserved, meaning this service is not sharing any resource with other services, therefore no service group is associated with it, so it MUST be ignored in processing.

Other fields are not changed in the sub-TLV.

Similarly, the updated format of Service-Oriented Available Resource sub-TLV is shown below.

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|ServiceOriented Avail Sub-Type | Length |P| Res | MT |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               SO-AvailRes                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               ResourceID/ServiceGroupID                   |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 4: Service-Oriented Available Resource sub-TLV

Where,

- \* ResourceID (4 bytes): an ID that identifies the shared computing resource consumed by this service. 0 is reserved, meaning no specific shared resource is identified, and it MUST be ignored in processing.
- \* or ServiceGroupID (4 bytes): an ID that identifies a group of service that the service belongs to. All the service within this group use a specific shared computing resource. 0 is reserved, meaning this service is not sharing any resource with other services, there for no service group is associated with it, so it MUST be ignored in processing.

Other fields are not changed in the sub-TLV.

When receiving the above sub-TLVs on the ingress CATS forwarder, if the association relation is not existed, the ingress can learn the association between the service ID (the prefix value in the NLIR) and the shared resource, which is identified by the resource ID or the

service group ID. And then, the ingress CATS forwarder will refresh the metrics of the shared computing resource, which will trigger all the associated services to process this metrics update event.

After the association relation is established, the later metrics update can use the following potential mechanisms to achieve less BGP update messages, to release the processing burden of BGP speakers.

Editor note: Authors will update the draft to choose the best solution according to the WG conclusion.

#### 5.2.1. Option 1: Reusing an Existing Update

When the first metrics update is finished for a service, the ingress CATS forwarder learned the association relation, and register the shared computing resource update event for this service. When multiple services are sharing the same resources, the ingress CATS forwarder will add these services into a group identified by the resource ID or service group ID.

For the following updates, as long as the ingress CATS forwarder receives an update message that contain a resource ID or service group ID, then the ingress updates the status of that shared resource or the associated service group using the shared resource, and trigger CATS processing for all the associated services.

In this way, there is no need to send per-service BGP update message to the ingress. Instead, choosing only one service update with a resource ID/service group ID can trigger all the updates for services sharing the shared resources.

Editor's note: In order to make the logic clearer, a flag can be added into Service-Oriented Capability and Service-Oriented Available Resource sub-TLVs, to identify the action of association establishment and batch update. The implementation is similar as the I-flag in Site Physical Availability Index Sub-TLV [I-D.ietf-idr-5g-edge-service-metadata]. This can be added in the future revision after WG discussion.

#### 5.2.2. Option 2: Using a Specific Prefix

Another solution is to use a specific configured or well-known prefix value to indicate that this update is for a shared resource in the Sharing mode instead of for a specific service/prefix.

This specific prefix value can be configured in the CATS domain by operators, or this document can apply for a specific purpose of prefix for this.

Editor's note: However, this is complicated than the option one, so editor will recommend the option 1.

### 5.2.3. Option 3: Using a New NLRI

The above options try to reuse the existing solution, and NLRI, which is easier in implementation. However, in the Sharing mode, the computing resource metrics update has become a resource-oriented or service-group-oriented solution, therefore, a resource-oriented or service-group-oriented NLRI might help the implementation from separating the computing metrics and network metrics.

A potential NLRI format is shown below.

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+
|length (1 Byte)|
+-----+
|           SiteID (2 Bytes)           |
+-----+
|           ResourceID/ServiceGroupID           |
+-----+
```

Figure 5: NLRI for Resouce/Service Group

Where,

- \* length (1 byte): one byte length, indicates the length of the NLRI.
- \* SiteID (2 bytes): a 2 bytes SiteID identifying the ID of a site. Typically, a site can be an edge cloud site. the value comes from the Site-ID field in Site Physical Availability Index Sub-TLV [I-D.ietf-idr-5g-edge-service-metadata].
- \* ResourceID (4 bytes): an ID that identifies the shared computing resource consumed by this service. 0 is reserved, meaning no specific shared resource is identified, and it MUST be ignored in processing.
- \* or ServiceGroupID (4 bytes): an ID that identifies a group of service that the service belongs to. All the service within this group use a specific shared computing resource. 0 is reserved, meaning this service is not sharing any resource with other services, there for no service group is associated with it, so it MUST be ignored in processing.

The AFI is recommended to be 1 or 2 meaning that it can be applied for service using IPv4 and IPv6 prefix, while the SAFI is TBD. However, it is possible to allocate a new AFI for this new type of NLRI.

The Path attributes of this NLRI reuse the Metadata Path Attribute [I-D.ietf-idr-5g-edge-service-metadata]. When distributing a metrics update of a computing resource, the related sub-TLVs are included in the Metadata Path Attribute accordingly, such as the Service-Oriented Capability sub-TLV. Since the ResourceID/ServiceGroupID has been in the NLRI, the ResourceID/ServiceGroupID SHOULD be ignored when encoding into the sub-TLVs, and MUST be ignored in receiving.

Because the update frequency of the computing metrics might be significantly higher than the existing metrics, it is reasonable to consider to put all computing metrics into a new NLRI, so that the filtering and route damping can be implemented in NLRI level.

Editor's Note: However, standardizing, implementing and deploying a new NLRI is a long-term work, which might be hard to be successful in the industry. Authors would like to hear more voices on these three options.

## 6. Deployment Considerations

TBD

## 7. Manageability Considerations

TBD

## 8. Security Considerations

This section will be updated in the future revision.

## 9. IANA Considerations

This section will be updated according to the progress of the sharing mode.

## 10. Normative 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-05, 10 February 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-05>>.

[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-26, 15 January 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-5g-edge-service-metadata-26>>.

[I-D.ysl-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-ysl-cats-metric-definition-03, 21 November 2024, <<https://datatracker.ietf.org/doc/html/draft-ysl-cats-metric-definition-03>>.

[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/rfc/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/rfc/rfc8174>>.

## Appendix A. Acknowledgements

The authors would like to thank Zongpeng Du, Huijuan Yao, Kehan Yao, Guofeng Qian, Haibo Wang, Xia Chen, Jianwei Mao, Zhenbin Li, Xinyue Zhang, Weier Li, and Linda Dunbar for their comments and suggestions.

## Contributors

Hang Shi  
Huawei Technologies  
Email: shihang9@huawei.com

## Authors' Addresses

Cheng Li  
Huawei Technologies  
China  
Email: c.l@huawei.com

Peng Liu  
China Mobile  
China  
Email: liupengyjy@chinamobile.com