Computing-Aware Traffic Steering
Internet-Draft
Intended status: Standards Track
Expires: 14 November 2026

B. Zhang, Ed.
Pengcheng Laboratory
Y. Dai, Ed.
Sun Yat-sen University
Z. Du, Ed.
China Mobile
C. Miao, Ed.
ZTE Corporation

13 May 2026

Computing Service Metric Definitions and Operation under CATS
draft-zhangb-cats-service-metrics-op-01

Abstract

Computing-Aware Traffic Steering (CATS) optimizes traffic forwarding by considering both computing and networking metrics. While the existing framework and metric drafts provide theoretical models (e.g., L1/L2 normalized metrics), they face significant challenges to achieve direct operational execution in real-world deployments. Normalization methods vary across providers, and aggregated unitless scores often lose critical operational information, making it difficult for routers to make precise decisions.

This document is proposed to fill this gap by providing an executable approach. It defines a set of Computing Service Metrics and their operations under the CATS framework. Instead of transmitting low-level raw hardware metrics, service sites dynamically evaluate and report service-oriented metrics (e.g., Global Available Slots) to the control plane. This enables efficient and precise traffic-steering policies without negating the value of existing normalized metrics.

This Internet-Draft will expire on 14 November 2026.

Table of Contents

1. Introduction 3
2. Terminology 3
3. Motivation and Problem Statement 4
4. Service Information and Metrics Definition 4
4.1. Mandatory Computing Service Information 5
4.1.1. Global Available Slots (GAS) 5
4.1.2. Computing Time 6
4.2. Optional Extension Metrics 6
4.2.1. Cost 6
4.2.2. Reputation 6
4.2.3. Security Label 7
4.2.4. Capability (L1/L2 Compatibility) 7
5. Operation under CATS Framework 7
5.1. Dynamic Metric Reporting 7
5.2. Information Collection and Routing Policies 8
6. Use Case Example 9
6.1. Service Distribution and Table Formation 9
6.2. Service Consumption and Resource Allocation 11
7. Security Considerations 13
8. IANA Considerations 13
9. References 13
9.1. Informative References 13
Authors' Addresses 14

1. Introduction

The Computing-Aware Traffic Steering (CATS) [I-D.ietf-cats-framework-24] architecture aims to steer service traffic to the most suitable service contact instance by evaluating both network state and computing resource availability. To achieve this, CATS Service Metric Agents (C-SMAs) collect computing metrics and advertise them to CATS Path Selectors (C-PSes).

[I-D.ietf-cats-metric-definition-07] introduces a multi-level metric framework (Level 0, Level 1, Level 2) and proposes normalizing heterogeneous computing metrics into unitless scores (e.g., compute_norm). While this establishes a solid theoretical baseline, mapping diverse hardware capabilities (CPUs, GPUs, NPUs) into a single normalized score is highly complex and provider-dependent. In practice, such normalization often obscures the actual service capacity, which cannot reach the required operational effect for fine-grained traffic steering.

To fill the gap between theoretical metric definitions and practical implementation, this document introduces a set of Computing Service Metrics. By decoupling the service capacity from hardware-specific raw metrics, service sites can directly expose actionable metrics that describe their concrete ability to handle specific services.

2. Terminology

This document makes use of the terms defined in [I-D.ietf-cats-framework-24] and [I-D.ietf-cats-metric-definition-07]. In particular, CS-ID and CSCI-ID are used as CATS identifiers. They provide stable service and service-contact-instance references for lookup and forwarding, but are not treated as computing metrics in this document. Additionally, the following terms are used:

* Global Available Slots (GAS): The maximum number of concurrent clients a service site is willing and able to serve for a specific CS-ID at a given time.

* CS-ID (CATS Service ID): An identifier for a service. It is used as a stable lookup key in the C-PS Computing Service Table.

* CSCI-ID (CATS Service Contact Instance ID): An identifier for a service contact instance. In this document, it is interpreted operationally as a locator, such as an IP address and port number, used to establish the data tunnel.

3. Motivation and Problem Statement

The CATS working group has made significant progress in defining how computing metrics should be collected and distributed. In particular, existing works introduce a comprehensive framework that categorizes computing metrics into Raw Metrics (Level 0) and Normalized Metrics (Level 1 and Level 2). However, a critical gap remains: how exactly to use these hardware-centric metrics to effectively steer traffic in operational networks.

This document does not negate the value of L1/L2 normalized metrics; rather, it identifies that relying solely on the normalization of raw hardware metrics poses operational challenges during routing execution:

1. The Implementation Gap (HOW to normalize?): In a real-world multi-vendor network, computing resources are highly heterogeneous. It is extremely difficult to establish a unified mathematical model that fairly normalizes a GPU’s capacity and a CPU’s capacity into the same 0-10 score.

2. The Information Loss Gap (WHY disseminate raw features?): Normalizing diverse hardware capabilities into a single unitless score results in the loss of actionable information. A normalized compute score of "7" cannot explicitly guarantee a client’s <10 ms delay requirement.

3. The Routing Mechanism Gap (WHO uses this data?): Routers (C-PS) do not need to know whether a service is backed by a CPU or a GPU. They only care about routing parameters: "Is there capacity?", "How long will it take?", and "Where is the destination?".

To bridge this gap, CATS requires a Service-Oriented Abstraction. We explicitly divide the required service information into Mandatory Computing Service Metrics and Optional Extension Metrics to support executable traffic-steering policies.

4. Service Information and Metrics Definition

This section defines the service information used by CATS control-plane components. Some fields are identifiers or locators, while others are service-oriented metrics. Metric examples follow the structural guidelines specified in Section 4 of [I-D.ietf-cats-metric-definition-07]. Encoding details are intentionally left as TBD until the working group has a stable representation to reference.

4.1. Mandatory Computing Service Information

These fields are essential for the C-PS to make fundamental traffic steering decisions. CS-ID and CSCI-ID are identifiers, while GAS and Computing Time are service-oriented metrics.

4.1.1. Global Available Slots (GAS)

GAS is the core contribution of this metric framework. It represents the maximum number of concurrent clients a service site can serve for a specific CS-ID.

Crucially, GAS acts as a direct abstraction layer (a "lid") over the complex and fluctuating raw computing metrics (CPU, GPU, Memory, Storage) and status metrics (load and health). Instead of exposing highly dynamic raw metrics to the network, the service site absorbs these variations internally. The site initially provides a GAS value based on its fixed resource allocation.

As the number of concurrent users increases, the GAS value naturally decreases. Furthermore, the site monitoring system dynamically reduces the GAS value upon detecting abnormal status metrics, such as:

* Load changes: Sudden increase in internal resources occupied by local users or tasks.

* Health changes: Sudden performance drop, possibly due to a cyber attack.

* Reachability: The site crashes or becomes unresponsive.

Note: The C-SMA proactively reports significant adjustments to the control plane according to local policy, thresholds, or aggregation intervals. Small per-session changes do not necessarily need to be reported immediately. When GAS drops to 0, it means the instance cannot allocate any more resources, and no new requests will be steered to it.

Basic fields:
  Metric Type: gas
  Level: L0/TBD
  Value: 500
  Source: estimation

Encoding details, including field length and wire format, are TBD.

4.1.2. Computing Time

The time required for the site to perform one service request. The service site dynamically adjusts this metric based on real-time load.

Basic fields:
- Metric Type: comp_time
- Level: L0/TBD
- Unit: ms
- Value: 5
- Source: estimation

Encoding details, including field length and wire format, are TBD.

4.2. Optional Extension Metrics

To accommodate advanced traffic-steering scenarios and maintain backward compatibility, the following optional fields are defined.

4.2.1. Cost

Self-defined by the service site to apply administrative or economic billing policies.

Basic fields:
- Metric Type: cost
- Level: L0/TBD
- Value: 100
- Source: nominal

Encoding details, including field length and wire format, are TBD.

4.2.2. Reputation

A dynamic quality score based on user feedback. Upon completion, if a user experiences long delays or inaccurate results, feedback is returned to the C-PS *along with the resource release message*.

Basic fields:
- Metric Type: reputation
- Level: L0/TBD
- Value: 8
- Source: estimation (user feedback)

Encoding details, including field length and wire format, are TBD.

4.2.3. Security Label

The Security Label reflects the security status of a service site. A higher score indicates a more secure site.

Score range: 0-10 (0 indicates the poorest security; 10 indicates optimal security).

Basic fields:
- Metric Type: security_label
- Level: L0/TBD
- Value: 9
- Source: estimation

Encoding details, including field length and wire format, are TBD.

4.2.4. Capability (L1/L2 Compatibility)

To maintain compatibility with L1/L2 normalized metrics, this optional field represents the overall computing and storage capability allocated by the site. It can correspond to a Level 1 or Level 2 overall capability score when such a normalized value is available.

Basic fields:
- Metric Type: site_cap
- Level: L1/L2/TBD
- Value: 7
- Source: normalization

Encoding details, including field length and wire format, are TBD.

5. Operation under CATS Framework

5.1. Dynamic Metric Reporting

Service sites proactively monitor their internal instances. When capacity drops, available slots cross a configured threshold, or latency spikes, the site reports updated metrics to the local C-SMA. The C-SMA may aggregate small per-session changes and synchronize only significant updates with the control plane.

Choosing appropriate protocols for conveying CATS metrics is important. For distributed systems, existing routing protocols such as BGP extensions[RFC4760] and GRASP [RFC8990] may serve as a baseline. However, considering that the CATS working group focuses on single-domain models, centralized approaches are highly suitable. In an SDN context [RFC7149] [RFC7426], the metric agent acts as an

application that uses a RESTful API via the northbound interface to report CATS metrics directly to the centralized C-PS (or SDN controller) for centralized decision-making.

5.2. Information Collection and Routing Policies

To ensure separation of concerns between computing resources and network states, the C-PS maintains two distinct data structures. Network metrics do not need to be normalized or redefined in CATS; they rely on existing network mechanisms.

* Computing Service Table: Formed by the C-SMA. It gathers service identifiers and service-oriented metrics (GAS, Computing Time, etc.) and is indexed by CS-ID.

* Network Service Table (e.g., TEDB in SDN controller): Formed by the C-NMA. C-NMA leverages existing techniques (e.g.,[RFC7471], [RFC8570], and [RFC8571]) to generate it. The Network Service Table contains network topology and link information (including delay, jitter, bandwidth, and availability). As an example, when the C-PS is implemented using an SDN Controller, the Network Service Table corresponds to the TEDB in the SDN control plane.

When a user request arrives, the C-PS combines the Computing Service Table and the Network Service Table (e.g., TEDB in SDN controller). The routing policy works by first querying the Computing Service Table to find candidate CSCI-IDs that meet the service requirements. Then, it queries the Network Service Table to determine the path and delay from the user's Ingress CATS-Forwarder to the candidate Egress CATS-Forwarder. The service node may be outside the ingress domain, so this document does not require measuring delay directly from the ingress to the service node. Finally, the C-PS selects the optimal CSCI-ID by minimizing the total service time, i.e., computing time plus ingress-to-egress network delay.

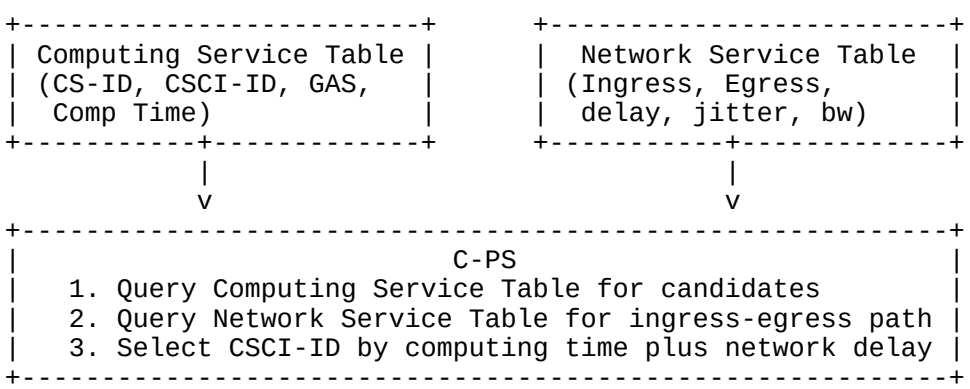

Figure 1: Service Selection Process Combining Computing and Network Metrics

6. Use Case Example

To illustrate the integrated routing logic of Service Metrics, consider a scenario where the C-PS combines both computing and network information. In this example, the C-PS filters candidates based on CS-ID and Computing Time, and combines that with the network delay between the Ingress and Egress CATS-Forwarders.

6.1. Service Distribution and Table Formation

Multiple service sites deploy various instances (e.g., AR1, AR2, LLM1). The C-SMA at each site pushes its local service information to the C-PS in the message format (CS-ID, CSCI-ID, Computing Time, GAS, [Optional Metrics]), forming the following unified Computing Service Table:

Figure 2 illustrates the metric dissemination process across the network:

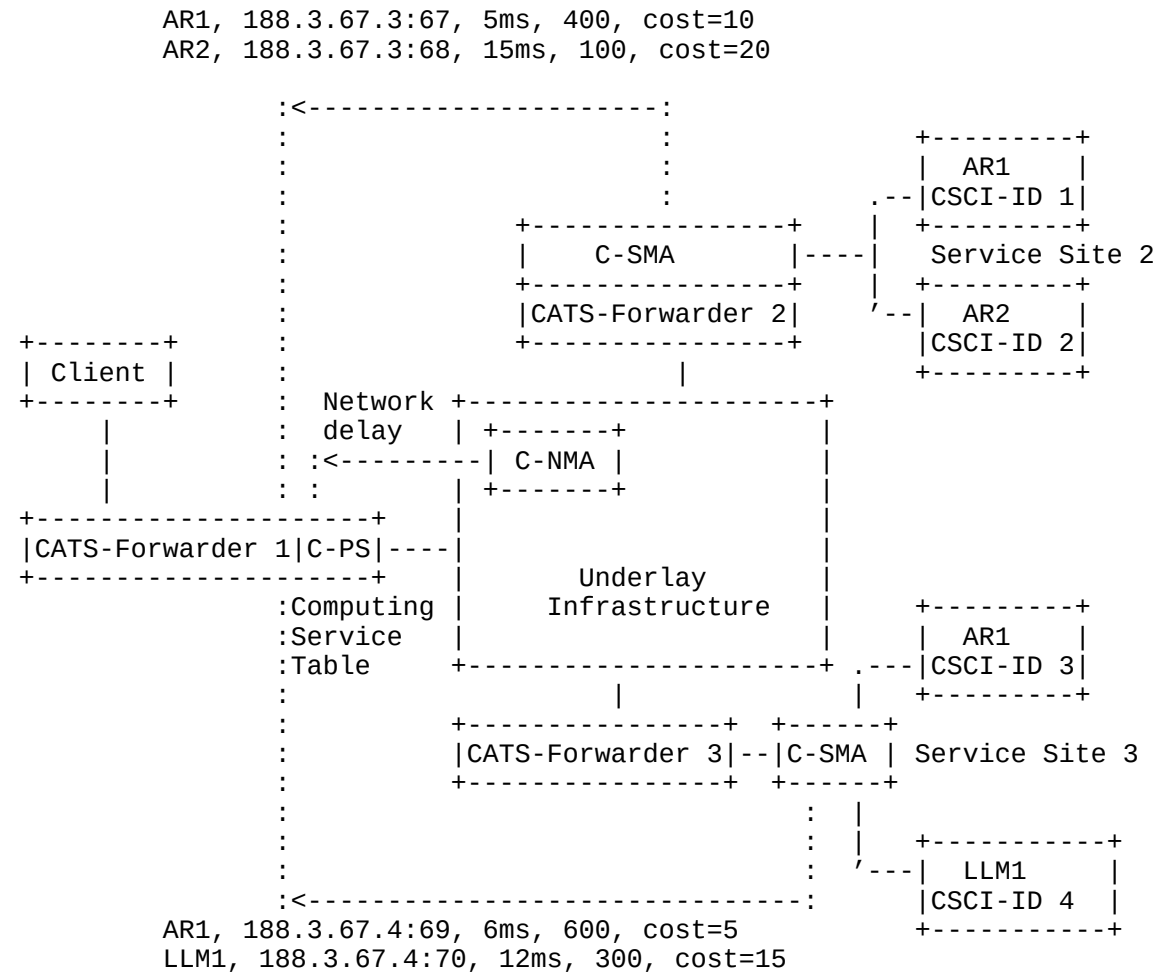

Figure 2: An Example of Service Metric Dissemination under CATS

The C-PS stores the C-SMA reports in the Computing Service Table:

CS-ID	CSCI-ID (IP:Port)	GAS	Comp Time(ms)	Cost (Optional)
AR1	188.3.67.3:67	400	5	10
AR2	188.3.67.3:68	100	15	20
AR1	188.3.67.4:69	600	6	5
LLM1	188.3.67.4:70	300	12	15

Table 1

The C-PS also maintains the Network Service Table (e.g., TEDB in an SDN controller) for network path information. An example subset relevant to the Ingress CATS-Forwarder is shown below:

Ingress CATS-Forwarder	Egress CATS-Forwarder	Network Delay (ms)
10.0.0.1	188.3.67.3	8
10.0.0.1	188.3.67.4	6

Table 2

Network delay information is maintained in the Network Service Table rather than being merged into the Computing Service Table. In this example, network delay refers to the Ingress-to-Egress delay.

6.2. Service Consumption and Resource Allocation

A client requests the AR1 service with a requirement for the shortest total service time (computing time plus Ingress-to-Egress network delay). Cost is omitted in this example to focus on the combined metric.

1. Match CS-ID (Control Plane): The C-PS scans the Computing Service Table and isolates entries matching CS-ID = AR1. Candidates: 188.3.67.3:67 (Comp Time = 5ms) and 188.3.67.4:69 (Comp Time = 6ms).

2. Query Network Service Table for Network Delay: The C-PS queries the Network Service Table (e.g., TEDB in an SDN controller) using the user's Ingress CATS-Forwarder and each candidate Egress CATS-Forwarder: path to the egress attached to 188.3.67.3 has a network delay of 8 ms; path to the egress attached to 188.3.67.4 has a network delay of 6 ms.

3. Calculate Total Service Time: The C-PS sums computing time and Ingress-to-Egress network delay for each candidate: candidate 188.3.67.3:67 has 5 ms + 8 ms = 13 ms; candidate 188.3.67.4:69 has 6 ms + 6 ms = 12 ms. The C-PS selects 188.3.67.4:69 because it offers the shortest total service time (12 ms), even though its computing time is slightly higher than candidate 67.

4. Allocate (Minus-one operation): The selected service site or its C-SMA updates the local GAS counter for 188.3.67.4:69 from 600 to 599. The C-SMA does not need to report every single decrement to the C-PS when the change is operationally insignificant; it may report aggregated or threshold-based updates.

5. Return Contact IP (Control Plane): The C-PS sends the selected contact IP (188.3.67.4:69) to the Ingress CATS-Forwarder (CATS-Forwarder 1). The Ingress CATS-Forwarder then forwards this information to the client. The internal metrics (computing time, network delay, cost, etc.) are shielded from the user.

6. Data Plane Establishment: The client sends the concrete service data to the Ingress CATS-Forwarder (CATS-Forwarder 1). The Ingress CATS-Forwarder encapsulates the packets and forwards them over the CATS-computed path to the selected Egress CATS-Forwarder (e.g., CATS-Forwarder 2 or 3). The Egress CATS-Forwarder decapsulates the packets and sends them to the target service site. After processing, the service site returns the response to the same Egress CATS-Forwarder, which forwards it back to the Ingress CATS-Forwarder and then to the client.

7. Release (Plus-one operation): Once the service session is completed, the selected service site or its C-SMA increments the local GAS counter back to 600. The updated value is synchronized to the C-PS according to the same local reporting policy, threshold, or aggregation interval.

7. Security Considerations

The dynamic reporting of Service Metrics introduces potential attack vectors. Authentication mechanisms between service sites and C-SMAs MUST be enforced. The Security Label (Section 4.2.3) can be utilized by the C-PS to prevent routing sensitive traffic to compromised sites.

8. IANA Considerations

This document has no IANA actions at this time.

9. References

9.1. Informative References

[I-D.ietf-cats-framework-24]
Li, C., Du, Z., Boucadair, M., Contreras, L. M., and J. Drake, "A Framework for Computing-Aware Traffic Steering (CATS)", April 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-24>.

[I-D.ietf-cats-metric-definition-07]
Kehan, Y., Li, C., Contreras, L. M., Ros-Giralt, J., and G. Zeng, "CATS Metrics Definition", 8 May 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-metric-definition-07>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC8990] Bormann, C., Carpenter, B., Ed., and B. Liu, Ed., "GeneRic Autonomic Signaling Protocol (GRASP)", March 2021, <https://www.rfc-editor.org/info/rfc8990>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", March 2019, <https://www.rfc-editor.org/info/rfc8570>.

[RFC8571] Ginsberg, L., Ed., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", March 2019, <https://www.rfc-editor.org/info/rfc8571>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", March 2014, <https://www.rfc-editor.org/info/rfc7149>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Salim, J. H., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", January 2015, <https://www.rfc-editor.org/info/rfc7426>.

Authors' Addresses

Bin Zhang (editor)
Pengcheng Laboratory
Email: zhangb@pcl.ac.cn

Yina Dai (editor)
Sun Yat-sen University
Email: daiyn5@mail2.sysu.edu.cn

Zongpeng Du (editor)
China Mobile
Email: duzongpeng@chinamobile.com

Chuanyang Miao (editor)
ZTE Corporation
Email: miao.chuanyang@zte.com.cn