Computing-Aware Traffic Steering
Internet-Draft
Intended status: Standards Track
Expires: 9 November 2026

Y. Kehan
China Mobile
C. Li
Huawei Technologies
L. M. Contreras
Telefonica
J. Ros-Giralt
Qualcomm Europe, Inc.
G. Zeng
Huawei Technologies

8 May 2026

CATS Metrics Definition
draft-ietf-cats-metric-definition-07

Abstract

Computing-Aware Traffic Steering (CATS) is a traffic engineering approach that optimizes the steering of traffic to a service instance by considering the dynamic state of computing and network resources. To enable such decisions, CATS components exchange metrics that describe resource conditions affecting service instance selection. This document focuses on compute and communication metrics for CATS and defines a hierarchical abstraction of these metrics to improve interoperability, scalability, and operational simplicity. It does not aim to standardize raw infrastructure (Level 0) metrics; instead, it specifies higher-level representations that can be derived from raw measurements using aggregation and normalization functions.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Computing-Aware Traffic Steering Working Group mailing list (cats@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/cats/.

Source for this draft and an issue tracker can be found at https://github.com/VMatrix1900/draft-cats-metric-definition.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 9 November 2026.

Copyright Notice

Table of Contents

1. Introduction

Service providers are deploying computing capabilities across the network for hosting applications such as distributed AI workloads, AR/VR and driverless vehicles, among others. In these deployments, multiple service instances are replicated across various sites to ensure sufficient capacity for maintaining the required Quality of Experience (QoE) expected by the application. To support the selection of these instances, a framework called Computing-Aware Traffic Steering (CATS) is introduced in [I-D.ietf-cats-framework].

CATS is a traffic engineering approach that optimizes the steering of traffic to a given service instance by considering the dynamic nature of computing and network resources. To achieve this, CATS components require performance metrics for both communication and compute resources. Since these resources are deployed by multiple providers, standardized metrics are essential to ensure interoperability and enable precise traffic steering decisions, thereby optimizing resource utilization and enhancing overall system performance.

There are already well-defined network metrics for traffic steering, such as Traffic Engineering (TE) metrics and IGP metrics (e.g., link delay, link delay variation)[RFC7471], which have been in use in network systems for a long time. In the context of CATS, computing metrics need to be introduced to enable joint TE decisions. [DMTF] defines some fine-grained computing metrics, such as CPU utilization, but directly using these fine-grained computing metrics lacks scalability.

This document does not attempt to standardize low-level fine-grained performance metrics. Instead, it organizes computing and communication metrics into three abstraction levels and defines a metric framework based on aggregation and normalization functions. The framework specifies four categories of Level 1 metrics and a normalized Level 2 metric, balancing metric expressiveness with scalability and ease of use.

2. Conventions and Definitions

This document uses the following terms defined in [I-D.ietf-cats-framework]:

* Computing-Aware Traffic Steering (CATS)

* Service

* Service site

* Service contact instance

* CATS Service Contact Instance ID (CSCI-ID)

* CATS Service Metric Agent (C-SMA)

* CATS Network Metric Agent (C-NMA)

* CATS Path Selector (C-PS)

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. Design Principles

3.1. Three-Level Metrics

As outlined in [I-D.ietf-cats-usecases-requirements], the resource model that defines CATS metrics MUST be scalable, ensuring that its implementation remains within a reasonable and sustainable cost. To that end, a CATS system should select the most appropriate metrics for instance selection, recognizing that different metrics may influence outcomes in distinct ways depending on the specific use case.

Defining metrics requires carefully balancing multiple considerations, including metric diversity, granularity, and rate of change (e.g., update frequency or advertisement churn). An excessive number of metrics, overly fine granularity, or high update frequency can lead to significant signaling overhead, reducing scalability of the metric distribution protocol. In contrast, metrics that are too few, too coarse-grained, or updated too infrequently may fail to provide sufficient information to support effective operational decisions.

Conceptually, it is necessary to define at least two fundamental levels of metrics: one comprising all raw metrics, and the other representing a simplified form---consisting of a single value that encapsulates the overall capability of a service instance.

However, such a definition may reduce implementation flexibility across diverse CATS use cases. Implementers typically seek balanced approaches that carefully manage trade-offs among encoding complexity, accuracy, scalability, and extensibility.

To ensure scalability while providing sufficient detail for effective decision-making, this document provides a definition of metrics that incorporates three levels of abstraction:

* *Level 0: Raw metrics.* These metrics are presented without abstraction, with each metric using its own unit and format as defined by the underlying resource.

* *Level 1: Metrics combined into categories.* These metrics are derived from Level 0 metrics by applying aggregation functions and, optionally, normalization functions to form category-specific metrics, such as computing and communication.

* *Level 2: A single normalized metric.* This metric is computed by aggregating lower-level metrics (Level 0 or Level 1) and applying normalization to produce a single, unitless Level 2 score within a defined range.

3.2. Level 0: Raw Metrics

Level 0 metrics encompass detailed, raw metrics, including but not limited to:

* CPU: Base Frequency, boosted frequency, number of cores, core utilization, memory bandwidth, memory size, memory utilization, power consumption.

* GPU: Frequency, number of render units, memory bandwidth, memory size, memory utilization, core utilization, power consumption.

* NPU: Computing power, utilization, power consumption.

* Communication: Throughput, bandwidth, link utilization, loss, delay, jitter, bytes/packets counters, and other network performance indicators.

* Storage: Available space, read speed, write speed.

* Service-specific metrics: Requests per second, output tokens per second.

Level 0 metrics serve as foundational data. Some of the Level 0 metrics may depend on performance monitoring, some may depend on active state, and some may be static. They provide basic information to support higher-level metrics, as detailed in the following sections.

Level 0 metrics can be encoded and exposed using an Application Programming Interface (API), such as a RESTful API, and can be technology- and implementation-specific. Different resources can have their own metrics, each conveying unique information about their status. These metrics can generally have units, such as bits per second (bps) or floating point instructions per second (flops).

[RFC8911] and [RFC8912] haved defined various network performance metrics and their registries. [DMTF] standardizes a set of computing metrics. These Level 0 raw metrics are not standardized in this document, but can be used as foundational data in CATS to derive higher level metrics.

3.3. Level 1: Metrics Combined in Categories

Level 1 metrics are grouped into four categories: computing, communication, service, and composed, with the possibility of additional categories being defined in future specifications. For each category, a single Level 1 metric is derived through an aggregation function and, when appropriate, further normalized to yield a unitless score reflecting the performance of the underlying resources. The Level 1 categories are described as follows:

* *Computing:* A value derived from aggregating one or more computing-related Level 0 metrics, such as CPU, GPU, and NPU utilization.

* *Communication:* A value derived from aggregating one or more communication-related Level 0 metrics, such as communication throughput.

* *Service:* A value derived from aggregating one or more service-related Level 0 metrics, such as tokens per second and service availability

* *Composed:* A value derived from aggregating a combination of computing, communication, and service metrics.

Refer to Section 4.2.1 and Section 4.2.2 for the definitions and examples of aggregation functions and normalization functions, respectively. Refer to Section 4.3 for the default policies and guidance provided to implementations.

Level 1 metrics allow to focus solely on the metric categories and their simple values, thereby avoiding the need to process solution-specific Level 0 metrics.

3.4. Level 2: A Single Normalized Metric

The Level 2 metric is a single, normalized score derived from lower-level metrics (Level 0 and/or Level 1) through the application of aggregation and normalization functions. Different implementations may apply different functions to characterize the overall performance of the underlying computing and communication resources. By consolidating multiple lower-level metrics into a single score, the Level 2 metric significantly reduces the complexity associated with metric collection and distribution. Section 4.3 further describes default policies for implementations.

Figure 1 provides a summary of the logical relationships between metrics across the three levels of abstraction.

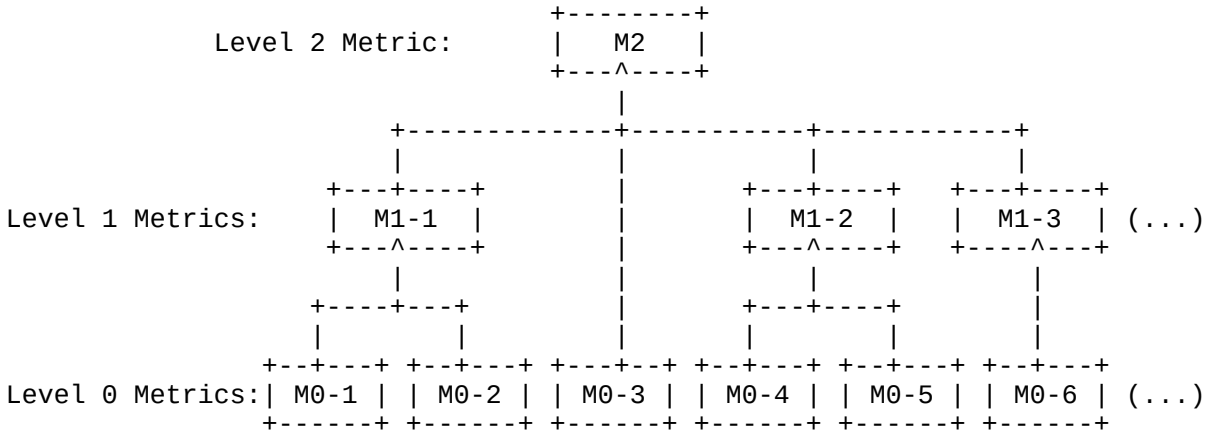

Figure 1: Logic of CATS Metrics in levels

4. CATS Metrics Framework and Specification

The CATS metrics framework defines how metrics are encoded and transmitted over the network. The representation should be flexible enough to accommodate various types of metrics along with their respective units and precision levels, yet simple enough to enable easy implementation and deployment across heterogeneous edge environments.

The design of CATS metrics framework has the following principles:

* Semantic granularity and extensibility: It adopts a layered metric abstraction.

* Metric source: It follows [RFC9439] by introducing a 'Source' field to distinguish metric context.

* Interoperability and flexibility: It allows implementation-specific aggregation and normalization functions, and adds default policies to ensure consistent cross-vendor interpretation.

4.1. CATS Metric Fields

Each CATS metric is expressed as a structured set of fields, with each field describing a specific property of the metric. The following definition introduces the fields used in the CATS metric representations.

* *Metric_Type*: This field specifies the category or kind of CATS metric being reported, such as computational resources, storage capacity, or network bandwidth. It acts as a label that enables network devices to identify the purpose of the metric.

* *Level*: This field specifies the level at which the metric is measured. It is used to categorize the metric based on its granularity and scope. There are only three valid metric levels defined in Section 3.1. This field can take two values: 1 for Level 1 and 2 for Level 2.

* *Format*: This field indicates the data encoding format of the metric, such as uint, ieee_754_float.

* *Length*: This field indicates the size of the value field measured in octets (bytes). It specifies how many bytes are used to store the value of the metric. The length field is important for memory allocation and data handling, ensuring that the value is stored and retrieved correctly.

* *Unit*: This field defines the measurement units for the metric, such as hertz (Hz) for frequency, bytes (B) for data size, or bits per seconds (bps) for data transfer rate. It is usually associated with the metric to provide context for the value.

* *Source*: This field describes the origin of the information used to obtain the metric. This field is optional. It may include one or more of the following non-mutually exclusive values:

  - 'nominal'. Similar to [RFC9439], "a 'nominal' metric indicates that the metric value is statically configured by the underlying devices. For example, bandwidth can indicate the maximum transmission rate of the involved device.

- ’estimation’. The ’estimation’ source indicates that the metric value is computed through an estimation process.

- ’directly measured’. This source indicates that the metric is obtained directly from the underlying device and it is not estimated.

- ’normalization’. The ’normalization’ source indicates that the metric value is normalized. This type of metrics does not have units. This document specifies that the normalized value range for each metric is 0 to 10, where 0 indicates the poorest compute/composed capability, and 10 indicates the optimal compute/composed capability.

- ’aggregation’. This source indicates that the metric value is obtained by using an aggregation function.

Nominal metrics have inherent physical meanings and specific units without any additional processing. Aggregated metrics may or may not have physical meanings, but they retain their significance relative to the directly measured metrics. Normalized metrics, on the other hand, might have physical meanings but lack units.

* *Statistics*: This field provides additional details about the metrics, particularly if there is any pre-computation performed on the metrics before they are collected. This field is optional. It is useful for services that require specific statistics for service instance selection. The ’Statistics’ field must be used together with the ’Measurement_Window’ parameter to indicate the sampling time interval. There are four kinds of statistics:

  - ’max’. The maximum value of the data collected over the intervals.

  - ’min’. The minimum value of the data collected over the intervals.

  - ’mean’. The average value of the data collected over the intervals.

  - ’cur’. The current value of the data collected.

* *Value*: This field represents the actual numerical value of the metric being measured. It provides the specific data point for the metric in question.

The value assignment and encoding rules for these fields are specified in Section Section 4.4.

4.2. Aggregation and Normalization Functions

In the context of CATS metric processing, aggregation and normalization are two fundamental operations that transform raw and derived metrics into forms suitable for decision-making and comparison across heterogeneous systems.

4.2.1. Aggregation

Aggregation functions combine multiple values into a single representative value. Aggregation functions can be applied at all metric levels. This document supports the spatial aggregation and temporal aggregation that are defined in [RFC5835], and further defines cross-category aggregation which can aggregate metrics from different types into a single value. The following are aggregation examples supported by CATS:

* Spatial or temporal aggregation of multiple metrics of the same type to produce a derived metric. In this case, because the input metrics are homogeneous, the resulting metric may retain the same units as the inputs. For example, CPU utilization measurements (expressed in percentage) collected from multiple service instances (spatial aggregation) or averaged over consecutive time intervals (temporal aggregation) can be aggregated to produce a representative CPU utilization metric. Such aggregation concepts are consistent with those described in [RFC5835].

* Aggregation of multiple metrics of different types to produce a higher-level metric that captures combined behavior across resource dimensions. In this case, because the input metrics use different units, the resulting metric cannot retain physical units and must be expressed as a unitless value. For example, CPU capacity (expressed in Hz) and available memory (expressed in bytes) can be combined through aggregation to generate a single computing-time metric that characterizes overall processing capability.

Some common aggregation functions include:

* Mean: Computes the arithmetic mean of a set of input values.

* Minimum / Maximum: Selects the lowest or highest value from a set of input values.

* Weighted average: Computes an average by applying weights to individual values according to their relative importance or priority.

Aggregation functions are not standardized in this document. They are implementation-specific and controlled by operator policies.

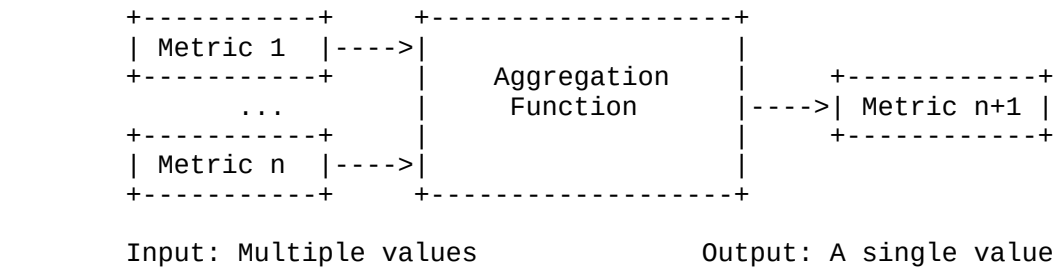

Figure 2: Aggregation function

4.2.2. Normalization

Normalization functions convert a metric value (with or without units) into a unitless normalized score. Normalized metrics facilitate composite scoring and ranking, and can be used to produce Level 1 and Level 2 metrics. The following are normalization examples supported by CATS:

* Normalizing a single Level 0 metric to generate a Level 1 or Level 2 normalized metric;

* Normalizing the output of aggregating multiple Level 0 metrics, to generate a Level 1 normalized metric.

Normalization functions are commonly used to transform metric values into a bounded range (e.g., an integer scale from 0 to 10) using techniques such as sigmoid function and min-max scaling [Min-max-sigmoid]:

* Sigmoid function: Smoothly maps input values to a bounded range.

* Min-max scaling: Rescales values based on known minimum and maximum bounds.

These normalization functions are also not standardized in this document. They are implementation-specific and controlled by operator policies.

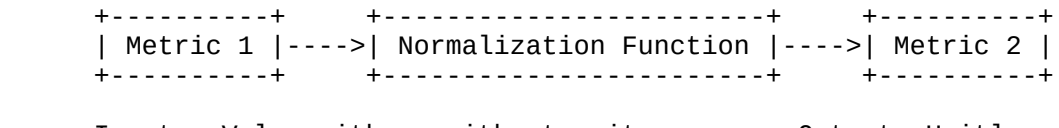

Figure 3: Normalization function

4.3. On the Meaning of Scores in Heterogeneous Metrics Systems

In a system like CATS, where metrics originate from heterogeneous resources---such as compute, communication, and storage---the interpretation of scores requires careful consideration. While normalization functions can convert raw metrics into unitless scores to enable comparison, these scores may not be directly comparable across different implementations. For example, a score of 7 on a scale from 0 to 10 may represent a high-quality resource in one implementation, but only an average one in another.

To achieve consistent cross-vendor behavior, the default normalization policies defined in this document should be followed by all implementations:

* Score directions and semantic mapping: A common 0-10 numeric range should be used for all normalized scores. Unless otherwise specified by the implementation in accompanying documentation, scores in the range 0-3 indicate low capability (not recommended for steering), 4-7 indicate medium capability (steering optional), and 8-10 indicate high capability (priority for steering). This mapping is normative for all CATS Level 1 and Level 2 metrics defined in this document.

* Normalization function baseline: Unless documented otherwise, implementations should use min-max scaling to map the aggregated raw value into the 0-10 range, based on implementation-specific minimum and maximum expected values. Other functions (e.g., sigmoid) are permitted but their parameters must be documented.

* Measurement window: There is no fixed default measurement window. For illustration, a window of 10 seconds is suggested as an example. Implementations can use their chosen window length, but they must indicate the window length as a parameter (i.e., via the Measurement_Window field defined in the registry entries).

4.4. Level Metric Representations

This section defines the representation format and constraints for Level 1 and Level 2 metrics respectively, to ensure consistent encoding and interoperability across implementations.

4.4.1. Level 0 Metrics

Level 0 metrics are raw metrics that are not standardized in this document. See Appendix A for examples of Level 0 metrics developed in the compute and communication industries and other standardization organizations such as the [DMTF].

4.4.2. Level 1 Metrics

Level 1 metrics are derived from Level 0 metrics through the application of aggregation functions and, when appropriate, normalization functions. Depending on how they are formed, Level 1 metrics MAY retain physical units inherited from their inputs or MAY be expressed as unitless values.

Level 1 metrics are organized into semantic categories such as computing, communication, service, and composed metrics. This categorization provides context and meaning to the resulting metrics and enables consistent interpretation across implementations.

The sources of Level 1 metrics is aggregation and normalization.

4.4.2.1. Combined Computing Metrics

The metric type of combined computing metrics is "computing_comb", and its format is unsigned integer. It has no unit. It will occupy an octet. Example:

Fields:

```
Metric_type: computing_comb
Level: Level 1
Format: unsigned integer
Length: one octet
Source: normalization
Value: 5
```

Figure 4: Example of a combined Level 1 computing metric

4.4.2.2. Combined Communication Metrics

The metric type of combined communication metrics is "communication_comb", and its format is unsigned integer. It has no unit. It will occupy an octet. Example:

```
   Fields:
         Metric_type: communication_comb
         Level: Level 1
         Format: unsigned integer
         Length: one octet
         Source: normalization
         Value: 1
```

Figure 5: Example of a combined Level 1 communication metric

4.4.2.3. Combined Service Metrics

The metric type of combined service metrics is "service_comb", and its format is unsigned integer. It has no unit. It will occupy an octet. Example:

```
   Fields:
         Metric_type: service_comb
         Level: Level 1
         Format: unsigned integer
         Length: one octet
         Source: normalization
         Value: 1
```

Figure 6: Example of a combined Level 1 service metric

4.4.2.4. Combined Composed Metrics

The metric type of combined composed metrics is "composed_comb", and its format is unsigned integer. It has no unit. It will occupy an octet. Example:

```
   Fields:
         Metric type: composed_comb
         Level: Level 1
         Format: unsigned integer
         Length: an octet
         Source: normalization
         Value: 8
```

Figure 7: Example of a combined Level 1 composed metric

4.4.3. Level 2 Metrics

A Level 2 metric is a single-value, normalized metric that does not carry any inherent physical unit. While each provider may employ its own internal methods to compute this value, all providers must adhere to the representation guidelines defined in this section to ensure consistency and interoperability of the normalized output.

The Metric Type is "norm_fi". The metric value is encoded as an unsigned integer, carries no unit, and is represented using a single octet. An example is shown below.

Fields:

```
Metric type: norm_fi
Level: Level 2
Format: unsigned integer
Length: an octet
Source: normalization
Value: 1
```

Figure 8: Example of a normalized Level 2 metric

5. Comparison among Metric Levels

Metrics are progressively consolidated from Level 0 to Level 1 and then to Level 2, with each level offering an increasing degree of abstraction to address the diverse requirements of different services. Table 1 provides a comparative overview of the defined metric levels.

Level	Encoding Complexity	Extensibility	Stability	Accuracy
Level 0	High	Low	Low	High
Level 1	Medium	Medium	Medium	Medium
Level 2	Low	High	High	Low

Table 1: Comparison among Metrics Levels

Since Level 0 metrics are raw and service-specific, individual services may define their own metric sets, potentially resulting in hundreds or even thousands of distinct metrics across deployments. This diversity introduces significant complexity in protocol encoding and standardization. Consequently, Level 0 metrics are confined to bespoke implementations tailored to specific service needs, rather than being standardized for broad protocol use. In contrast, Level 1 metrics organize raw data into standardized categories, each consolidated into a single value. This structure makes them more suitable for protocol encoding and standardization. The Level 2 metric takes simplification a step further by consolidating all relevant information into a single normalized value, making them the easiest to encode, transmit, and standardize.

Therefore, from the perspective of encoding complexity, Level 1 and Level 2 metrics are recommended.

When considering extensibility, Level 0 metrics allow new services to define their own custom metrics. However, this flexibility requires corresponding protocol extensions, and the proliferation of metric types can introduce significant overhead, ultimately reducing the protocol’s extensibility. In contrast, Level 1 metrics introduce only a limited set of standardized categories, making protocol extensions more manageable. Level 2 metrics go even further by consolidating all information into a single normalized value, placing the least burden on the protocol.

Therefore, from an extensibility standpoint, Level 1 and Level 2 metrics are recommended.

Regarding stability, Level 0 raw metrics may require frequent protocol extensions as new metrics are introduced, leading to an unstable and evolving protocol format. For this reason, standardizing Level 0 metrics within the protocol is not recommended. In contrast, Level 1 metrics involve only a limited set of predefined categories, and Level 2 metrics rely on a single consolidated value, both of which contribute to a more stable and maintainable protocol design.

Therefore, from a stability standpoint, Level 1 and Level 2 metrics are preferred.

In conclusion, for CATS, Level 2 metrics are recommended due to their simplicity and minimal protocol overhead. If more advanced scheduling capabilities are required, Level 1 metrics offer a balanced approach with manageable complexity. While Level 0 metrics are the most detailed and dynamic, their high overhead makes them unsuitable for direct transmission to network devices and thus not recommended for standard protocol integration.

6. CATS Metric Registry Entries

This section defines the formal registry entries for one CATS Level 2 metric and four Level 1 metrics, intended for registration with IANA. By providing a common template that specifies the metric’s summary, definition, method of measurement, output, and administrative items, this section ensures interoperability among different implementations.

6.1. CATS Level 2 Metric Registry Entry

This section gives an initial Registry Entry for the CATS Level 2 metric.

6.1.1. Summary

This category includes multiple indexes to the Registry Entry: the element ID, Metric Name, URI, Metric Description, Metric Controller, and Metric Version.

6.1.1.1. ID (Identifier)

IANA has allocated the Identifier XXX for the Named Metric Entry in this section. See the next Section for mapping to Names.

6.1.1.2. Name

Norm_Passive_CATS-Level 2_RFCXXXXsecY_Unitless_Singleton

Naming Rule Explanation

* Norm: Metric type (Normalized Score)
* Passive: Measurement method
* CATS-Level 2: Metric level (CATS Metric Framework Level 2)
* RFCXXXXsecY: Specification reference (To-be-assigned RFC number and section number)

* Unitless: Metric has no units

* Singleton: Metric is a single value

6.1.1.3. URI

To-be-assigned.

6.1.1.4. Description

This metric represents a single normalized score used within CATS (Level 2). It is derived by aggregating one or more CATS Level 0 and/or Level 1 metrics, followed by a normalization process that produces a unitless value. The resulting score provides a concise assessment of the overall capability of a service instance, enabling rapid comparison across instances and supporting efficient traffic steering decisions.

6.1.1.5. Change Controller

IETF

6.1.1.6. Version

1.0

6.1.2. Metric Definition

6.1.2.1. Reference Definition

[I-D.ietf-cats-metric-definition] Core referenced sections: Section 3.4 (Level 2 Level Metric Definition), Section 4.2 (Aggregation and Normalization Functions)

6.1.2.2. Fixed Parameters

* Normalization score range: 0-10 (0 indicates the poorest capability, 10 indicates the optimal capability)

* Data precision: non-negative integer

6.1.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous method for implementations.

6.1.3.1. Reference Methods

Raw Metrics collection: Collect Level 0 service and compute raw metrics using platform-specific management protocols or tools (e.g., Prometheus [Prometheus] in Kubernetes). Collect Level 0 network performance raw metrics using existing standardized protocols (e.g., NETCONF [RFC6241], IPFIX [RFC7011]).

Aggregation logic: Refer to Section 4.2.1.

Normalization logic: Refer to Section 4.2.2.

The reference method aggregates and normalizes Level 0 metrics to generate Level 1 metrics in different categories, and further calculates a Level 2 singleton score for ultimate normalization.

6.1.3.2. Packet Stream Generation

N/A

6.1.3.3. Traffic Filtering (Observation) Details

N/A

6.1.3.4. Sampling Distribution

Sampling method: Continuous sampling (e.g., collect Level 0 metrics every 10 seconds)

6.1.3.5. Runtime Parameters and Data Format

CATS Service Contact Instance ID (CSCI-ID): an identifier of CATS service contact instance. According to [I-D.ietf-cats-framework], a unicast IP address can be an example of identifier. (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Service_Instance_IP: Service instance IP address (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Measurement_Window: Metric measurement time window (Units: seconds, milliseconds; Format: uint64; Default: 10 seconds)

6.1.3.6. Roles

C-SMA: Collects Level 0 service and compute raw metrics, and optionally calculates Level 1 metrics according to service-specific strategies.

C-NMA: Collects Level 0 network performance raw metrics, and optionally calculates Level 1 metrics according to service-specific strategies.

C-PS: Aggregate all Level 1 metrics collected from C-NMA and C-SMA to calculate the Level 2 metric. ### Output

This category specifies all details of the output of measurements using the metric.

6.1.3.7. Type

Singleton value

6.1.3.8. Reference Definition

Output format: Refer to [I-D.ietf-cats-metric-definition] Section 4.4.3

Score semantics: 0-3 (Low capability, not recommended for steering), 4-7 (Medium capability, optional for steering), 8-10 (High capability, priority for steering)

6.1.3.9. Metric Units

Unitless

6.1.3.10. Calibration

Calibration method: Conduct benchmark calibration based on standard test sets (fixed workload) to ensure the output score deviation of C-SMA and C-NMA is lower than 0.1 (one abnormal score in every ten test rounds).

6.1.4. Administrative Items

6.1.4.1. Status

Current

6.1.4.2. Requester

To-be-assgined

6.1.4.3. Revision

1.0

6.1.4.4. Revision Date

2026-01-20

6.1.4.5. Comments and Remarks

None

6.2. CATS Level 1 Metric Registry Entry: Computing

This section gives an initial Registry Entry for the CATS Level 1 metric in the _computing_ category.

6.2.1. Summary

This category includes multiple indexes to the Registry Entry: the element ID, Metric Name, URI, Metric Description, Metric Controller, and Metric Version.

6.2.1.1. ID (Identifier)

IANA has allocated the Identifier XXX for the Named Metric Entry in this section. See the next Section for mapping to Names.

6.2.1.2. Name

Comb_Passive_CATS-Level 1_Computing_RFCXXXXsecY_Unitless_Singleton

Naming Rule Explanation

* Comb: Metric type (Combined Score)
* Passive: Measurement method
* CATS-Level 1: Metric level (CATS Metric Framework Level 1)
* Computing: Metric category (Computing)
* RFCXXXXsecY: Specification reference (To-be-assigned RFC number and section number)
* Unitless: Metric has no units
* Singleton: Metric is a single value for the computing category

6.2.1.3. URI

To-be-assigned.

6.2.1.4. Description

This metric represents a single normalized score for the _computing_ category within CATS (Level 1). It is derived from one or more computing-related Level 0 metrics (e.g., CPU/GPU/NPU utilization, CPU frequency, memory utilization, or other computing resource indicators) by applying an implementation-specific aggregation function over the selected Level 0 computing metrics and then applying a normalization function to produce a unitless score.

The resulting score provides a concise indication of the relative computing capability (or headroom) of a service contact instance for the purpose of instance selection and traffic steering. Higher values indicate better computing capability according to the provider’s normalization strategy.

6.2.1.5. Change Controller

IETF

6.2.1.6. Version

1.0

6.2.2. Metric Definition

6.2.2.1. Reference Definition

[I-D.ietf-cats-metric-definition]

Core referenced sections: Section 3.3 (Level 1 Level Metric Definition), Section 4.2 (Aggregation and Normalization Functions), Section 4.4.2 (Level 1 Metric Representations)

6.2.2.2. Fixed Parameters

* Normalization score range: 0-10 (0 indicates the poorest computing capability, 10 indicates the optimal computing capability)

* Data precision: non-negative integer

* Metric type: "computing_comb"

* Level: Level 1

* Metric units: Unitless

6.2.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous method for implementations.

6.2.3.1. Reference Methods

Raw Metrics collection: Collect computing-related Level 0 raw metrics (e.g., CPU/GPU/NPU, memory, and relevant platform counters) using platform-specific management protocols or tools (e.g., Prometheus [Prometheus] in Kubernetes or equivalent telemetry systems).

Aggregation logic (within computing category): Refer to Section 4.2.1 to combine selected Level 0 computing metrics into a single intermediate value prior to normalization. The selection of Level 0 computing metrics and any weights used are implementation-specific.

Normalization logic: Refer to Section 4.2.2 to map the aggregated (or directly selected) computing value into the fixed score range.

The reference method aggregates and normalizes Level 0 computing metrics to generate a single Level 1 computing score ("computing_comb").

6.2.3.2. Packet Stream Generation

N/A

6.2.3.3. Traffic Filtering (Observation) Details

N/A

6.2.3.4. Sampling Distribution

Sampling method: Continuous sampling (e.g., collect underlying Level 0 computing metrics every 10 seconds)

6.2.3.5. Runtime Parameters and Data Format

CATS Service Contact Instance ID (CSCI-ID): an identifier of CATS service contact instance. According to [I-D.ietf-cats-framework], a unicast IP address can be an example of identifier. (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Service_Instance_IP: Service instance IP address (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Measurement_Window: Metric measurement time window (Units: seconds, milliseconds; Format: uint64; Default: 10 seconds)

6.2.3.6. Roles

C-SMA: Collects Level 0 compute raw metrics and calculates the Level 1 compute normalized score ("computing_comb") according to service/provider-specific aggregation and normalization strategies.

C-NMA: Not required for this metric.

6.2.4. Output

This category specifies all details of the output of measurements using the metric.

6.2.4.1. Type

Singleton value

6.2.4.2. Reference Definition

Output format: Refer to [I-D.ietf-cats-metric-definition] Section 4.4.2

Score semantics: 0-3 (Low compute capability, not recommended for steering), 4-7 (Medium compute capability, optional for steering), 8-10 (High compute capability, priority for steering)

6.2.4.3. Metric Units

Unitless

6.2.4.4. Calibration

Calibration method: Conduct benchmark calibration based on representative compute workloads (fixed test workload profiles) to align the mapping from Level 0 computing metrics to the Level 1 score, such that score deviation across measurement agents within the same administrative domain is minimized (e.g., less than 0.1 over repeated test rounds).

6.2.5. Administrative Items

6.2.5.1. Status

Current

6.2.5.2. Requester

To-be-assgined

6.2.5.3. Revision

1.0

6.2.5.4. Revision Date

2026-01-20

6.2.5.5. Comments and Remarks

None

6.3. CATS Level 1 Metric Registry Entry: Communication

This section gives an initial Registry Entry for the CATS Level 1 metric in the _communication_ category.

6.3.1. Summary

This category includes multiple indexes to the Registry Entry: the element ID, Metric Name, URI, Metric Description, Metric Controller, and Metric Version.

6.3.1.1. ID (Identifier)

IANA has allocated the Identifier XXX for the Named Metric Entry in this section. See the next Section for mapping to Names.

6.3.1.2. Name

Comb_Passive_CATS-Level 1_Communication_RFCXXXXsecY_Unitless_Singleton

Naming Rule Explanation

* Comb: Metric type (Combined Score)

* Passive: Measurement method

* CATS-Level 1: Metric level (CATS Metric Framework Level 1)

* Communication: Metric category (Communication)

* RFCXXXXsecY: Specification reference (To-be-assigned RFC number and section number)

* Unitless: Metric has no units

* Singleton: Metric is a single value for the communication category

6.3.1.3. URI

To-be-assigned.

6.3.1.4. Description

This metric represents a single normalized score for the _communication_ category within CATS (Level 1). It is derived from one or more communication-related Level 0 metrics (e.g., throughput, bandwidth, link utilization, loss, delay, jitter, bytes/packets counters, and other network performance indicators) by applying an implementation-specific aggregation function over the selected Level 0 communication metrics and then applying a normalization function to produce a unitless score.

The resulting score provides a concise indication of the relative communication capability (or headroom) associated with reaching a service contact instance for the purpose of instance selection and traffic steering. Higher values indicate better communication capability according to the provider’s normalization strategy.

6.3.1.5. Change Controller

IETF

6.3.1.6. Version

1.0

6.3.2. Metric Definition

6.3.2.1. Reference Definition

[I-D.ietf-cats-metric-definition]

Core referenced sections: Section 3.3 (Level 1 Level Metric Definition), Section 4.2 (Aggregation and Normalization Functions), Section 4.4.2 (Level 1 Metric Representations)

6.3.2.2. Fixed Parameters

* Normalization score range: 0-10 (0 indicates the poorest communication capability, 10 indicates the optimal communication capability)

* Data precision: non-negative integer

* Metric type: "communication_comb"

* Level: Level 1

* Metric units: Unitless

6.3.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous method for implementations.

6.3.3.1. Reference Methods

Raw Metrics collection: Collect communication-related Level 0 raw metrics using existing standardized protocols and telemetry systems (e.g., NETCONF [RFC6241], IPFIX [RFC7011]), and/or using network performance metric definitions and registries such as [RFC8911], [RFC8912], and [RFC9439] where applicable.

Aggregation logic (within communication category): Refer to [I-D.ietf-cats-metric-definition] Section 4.2.1 (e.g., Weighted Average Aggregation) to combine selected Level 0 communication metrics into a single intermediate value prior to normalization. The selection of Level 0 communication metrics and any weights used are implementation-specific.

Normalization logic: Refer to [I-D.ietf-cats-metric-definition] Section 4.2.2 (e.g., Sigmoid Normalization or Min-max scaling) to map the aggregated (or directly selected) communication value into the fixed score range.

The reference method aggregates and normalizes Level 0 communication metrics to generate a single Level 1 communication score ("communication_comb"). No cross-category aggregation is performed for this metric (i.e., it does not incorporate compute or service metrics).

6.3.3.2. Packet Stream Generation

N/A

6.3.3.3. Traffic Filtering (Observation) Details

N/A

6.3.3.4. Sampling Distribution

Sampling method: Continuous sampling (e.g., collect underlying Level 0 communication metrics every 10 seconds)

6.3.3.5. Runtime Parameters and Data Format

CATS Service Contact Instance ID (CSCI-ID): an identifier of CATS service contact instance. According to [I-D.ietf-cats-framework], a unicast IP address can be an example of identifier. (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Service_Instance_IP: Service instance IP address (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Measurement_Window: Metric measurement time window (Units: seconds, milliseconds; Format: uint64; Default: 10 seconds)

6.3.3.6. Roles

C-NMA: Collects Level 0 communication raw metrics and calculates the Level 1 communication normalized score ("communication_comb") according to provider-specific aggregation and normalization strategies.

C-SMA: Not required for this metric.

6.3.4. Output

This category specifies all details of the output of measurements using the metric.

6.3.4.1. Type

Singleton value

6.3.4.2. Reference Definition

Output format: Refer to [I-D.ietf-cats-metric-definition] Section 4.4.2

Score semantics: 0-3 (Low communication capability, not recommended for steering), 4-7 (Medium communication capability, optional for steering), 8-10 (High communication capability, priority for steering)

6.3.4.3. Metric Units

Unitless

6.3.4.4. Calibration

Calibration method: Conduct benchmark calibration based on representative network test profiles (e.g., fixed traffic mixes and path conditions) to align the mapping from Level 0 communication metrics to the Level 1 score, such that score deviation across measurement agents within the same administrative domain is minimized (e.g., less than 0.1 over repeated test rounds).

6.3.5. Administrative Items

6.3.5.1. Status

Current

6.3.5.2. Requester

To-be-assgined

6.3.5.3. Revision

1.0

6.3.5.4. Revision Date

2026-01-20

6.3.5.5. Comments and Remarks

None

6.4. CATS Level 1 Metric Registry Entry: Service

This section gives an initial Registry Entry for the CATS Level 1 metric in the _service_ category.

6.4.1. Summary

This category includes multiple indexes to the Registry Entry: the element ID, Metric Name, URI, Metric Description, Metric Controller, and Metric Version.

6.4.1.1. ID (Identifier)

IANA has allocated the Identifier XXX for the Named Metric Entry in this section. See the next Section for mapping to Names.

6.4.1.2. Name

Comb_Passive_CATS-Level 1_Service_RFCXXXXsecY_Unitless_Singleton

Naming Rule Explanation

* Comb: Metric type (Combined Score)
* Passive: Measurement method
* CATS-Level 1: Metric level (CATS Metric Framework Level 1)
* Service: Metric category (Service)
* RFCXXXXsecY: Specification reference (To-be-assigned RFC number and section number)
* Unitless: Metric has no units
* Singleton: Metric is a single value for the service category

6.4.1.3. URI

To-be-assigned.

6.4.1.4. Description

This metric represents a single normalized score for the _service_ category within CATS (Level 1). It is derived from one or more service-related Level 0 metrics that characterize the health and performance of the service instance itself (e.g., service availability, request success rate, admission/overload indicators, tokens per second and/or requests per second, application-level queue depth, and other service KPIs) by applying an implementation-specific aggregation function over the selected Level 0 service metrics and then applying a normalization function to produce a unitless score.

The resulting score provides a concise indication of the relative service capability (or headroom) of a service contact instance for the purpose of instance selection and traffic steering. Higher values indicate better service capability according to the provider’s normalization strategy.

6.4.1.5. Change Controller

IETF

6.4.1.6. Version

1.0

6.4.2. Metric Definition

6.4.2.1. Reference Definition

[I-D.ietf-cats-metric-definition]

Core referenced sections: Section 3.3 (Level 1 Level Metric Definition), Section 4.2 (Aggregation and Normalization Functions), Section 4.4.2 (Level 1 Metric Representations)

6.4.2.2. Fixed Parameters

* Normalization score range: 0-10 (0 indicates the poorest service capability, 10 indicates the optimal service capability)
* Data precision: non-negative integer
* Metric type: "service_comb"
* Level: Level 1
* Metric units: Unitless

6.4.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous method for implementations.

6.4.3.1. Reference Methods

Raw Metrics collection: Collect service-related Level 0 raw metrics from the service runtime and service management plane using platform-specific telemetry systems (e.g., Prometheus [Prometheus] in Kubernetes or equivalent monitoring/observability tools). These metrics are service-dependent and may include availability/health status, success/error rates, overload or admission control signals, and throughput indicators (e.g., tokens per second for AI inference services), among others.

Aggregation logic (within service category): Refer to [I-D.ietf-cats-metric-definition] Section 4.2.1 (e.g., Weighted Average Aggregation) to combine selected Level 0 service metrics into a single intermediate value prior to normalization. The selection of Level 0 service metrics, any weights used, and any gating logic (e.g., forcing the score to a low value when the instance is unhealthy) are implementation-specific.

Normalization logic: Refer to [I-D.ietf-cats-metric-definition] Section 4.2.2 (e.g., Sigmoid Normalization or Min-max scaling) to map the aggregated (or directly selected) service value into the fixed score range.

The reference method aggregates and normalizes Level 0 service metrics to generate a single Level 1 service score ("service_comb"). No cross-category aggregation is performed for this metric (i.e., it does not incorporate compute or communication metrics).

6.4.3.2. Packet Stream Generation

N/A

6.4.3.3. Traffic Filtering (Observation) Details

N/A

6.4.3.4. Sampling Distribution

Sampling method: Continuous sampling (e.g., collect underlying Level 0 service metrics every 10 seconds)

6.4.3.5. Runtime Parameters and Data Format

CATS Service Contact Instance ID (CSCI-ID): an identifier of CATS service contact instance. According to [I-D.ietf-cats-framework], a unicast IP address can be an example of identifier. (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Service_Instance_IP: Service instance IP address (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Measurement_Window: Metric measurement time window (Units: seconds, milliseconds; Format: uint64; Default: 10 seconds)

6.4.3.6. Roles

Service contact instace: Collects Level 0 service raw metrics and calculates the Level 1 service normalized score ("service_comb") according to service/provider-specific aggregation and normalization strategies.

C-NMA: Not required for this metric.

6.4.4. Output

This category specifies all details of the output of measurements using the metric.

6.4.4.1. Type

Singleton value

6.4.4.2. Reference Definition

Output format: Refer to [I-D.ietf-cats-metric-definition] Section 4.4.2

Score semantics: 0-3 (Low service capability, not recommended for steering), 4-7 (Medium service capability, optional for steering), 8-10 (High service capability, priority for steering)

6.4.4.3. Metric Units

Unitless

6.4.4.4. Calibration

Calibration method: Conduct benchmark calibration based on representative service workload profiles (fixed request mixes and known-good baselines) to align the mapping from Level 0 service metrics to the Level 1 score, such that score deviation across measurement agents within the same administrative domain is minimized (e.g., less than 0.1 over repeated test rounds). Calibration MAY include failure/overload scenarios (e.g., simulated dependency failures or saturation) to ensure score behavior is consistent with operational intent.

6.4.5. Administrative Items

6.4.5.1. Status

Current

6.4.5.2. Requester

To-be-assigned

6.4.5.3. Revision

1.0

6.4.5.4. Revision Date

2026-01-20

6.4.5.5. Comments and Remarks

None

6.5. CATS Level 1 Metric Registry Entry: Composed

This section gives an initial Registry Entry for the CATS Level 1 metric in the _composed_ category.

6.5.1. Summary

This category includes multiple indexes to the Registry Entry: the element ID, Metric Name, URI, Metric Description, Metric Controller, and Metric Version.

6.5.1.1. ID (Identifier)

IANA has allocated the Identifier XXX for the Named Metric Entry in this section. See the next Section for mapping to Names.

6.5.1.2. Name

Comb_Passive_CATS-Level 1_Composed_RFCXXXXsecY_Unitless_Singleton

Naming Rule Explanation

* Comb: Metric type (Combined Score)
* Passive: Measurement method
* CATS-Level 1: Metric level (CATS Metric Framework Level 1)
* Composed: Metric category (Composed)
* RFCXXXXsecY: Specification reference (To-be-assigned RFC number and section number)
* Unitless: Metric has no units
* Singleton: Metric is a single value for the composed category

6.5.1.3. URI

To-be-assigned.

6.5.1.4. Description

This metric represents a single normalized score for the _composed_ category within CATS (Level 1). A composed metric is derived by combining multiple lower-level metrics that may span different categories (e.g., compute, communication, and service) and/or multiple components along the request path.

Typical examples of composed metrics include (but are not limited to) end-to-end delay, application-level response time, or other synthesized indicators that are computed as a function of multiple contributing factors (e.g., the sum of compute processing delay and network transmission delay along the selected path).

The composed Level 1 score is obtained by applying an implementation-specific aggregation function over the selected contributing Level 0 metrics (and/or previously computed Level 1 category metrics), followed by a normalization function that yields a unitless score. Higher values indicate better composed capability according to the provider’s normalization strategy.

6.5.1.5. Change Controller

IETF

6.5.1.6. Version

1.0

6.5.2. Metric Definition

6.5.2.1. Reference Definition

[I-D.ietf-cats-metric-definition]

Core referenced sections: Section 3.3 (Level 1 Level Metric Definition), Section 4.2 (Aggregation and Normalization Functions), Section 4.4.2 (Level 1 Metric Representations)

6.5.2.2. Fixed Parameters

* Normalization score range: 0-10 (0 indicates the poorest composed capability, 10 indicates the optimal composed capability)

* Data precision: non-negative integer

* Metric type: "composed_comb"

* Level: Level 1

* Metric units: Unitless

6.5.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous method for implementations.

6.5.3.1. Reference Methods

Raw Metrics collection: Collect contributing Level 0 raw metrics from the relevant sources across categories. For example, compute- and service-related Level 0 metrics may be collected by a C-SMA using platform-specific telemetry systems (e.g., Prometheus [Prometheus]), while communication-related Level 0 metrics may be collected by a C-NMA using network telemetry and protocols (e.g., NETCONF [RFC6241], IPFIX [RFC7011]), and/or using network performance metric definitions and registries such as [RFC8911], [RFC8912], and [RFC9439] where applicable.

Aggregation logic (within composed category): Refer to [I-D.ietf-cats-metric-definition] Section 4.2.1 (e.g., Weighted Average Aggregation) to combine selected contributing metrics into a single intermediate value prior to normalization. The aggregation function MAY combine Level 0 metrics directly, and/or MAY take as input one or more Level 1 category metrics (e.g., "computing_comb" and "communication_comb"). The selection of contributing metrics, any weights used, and the composition model (e.g., sum of delays, bottleneck/maximum, or weighted utility) are implementation-specific.

Normalization logic: Refer to [I-D.ietf-cats-metric-definition] Section 4.2.2 (e.g., Sigmoid Normalization or Min-max scaling) to map the aggregated composed value into the fixed score range.

The reference method aggregates and normalizes the selected contributing metrics to generate a single Level 1 composed score ("composed_comb").

6.5.3.2. Packet Stream Generation

N/A

6.5.3.3. Traffic Filtering (Observation) Details

N/A

6.5.3.4. Sampling Distribution

Sampling method: Continuous sampling (e.g., collect underlying contributing metrics every 10 seconds)

6.5.3.5. Runtime Parameters and Data Format

CATS Service Contact Instance ID (CSCI-ID): an identifier of CATS service contact instance. According to [I-D.ietf-cats-framework], a unicast IP address can be an example of identifier. (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Service_Instance_IP: Service instance IP address (format: ipv4-address-no-zone or ipv6-address-no-zone, complying with [RFC9911])

Measurement_Window: Metric measurement time window (Units: seconds, milliseconds; Format: uint64; Default: 10 seconds)

6.5.3.6. Roles

C-SMA: Collects Level 0 service and compute raw metrics that may contribute to the composed score, and MAY calculate the Level 1 composed score ("composed_comb") when it has access to the required inputs.

C-NMA: Collects Level 0 communication raw metrics that may contribute to the composed score, and MAY calculate the Level 1 composed score ("composed_comb") when it has access to the required inputs.

CATS Controller (or other CATS component): MAY compute the Level 1 composed score when the contributing metrics originate from multiple agents and are combined at a common computation point.

6.5.4. Output

This category specifies all details of the output of measurements using the metric.

6.5.4.1. Type

Singleton value

6.5.4.2. Reference Definition

Output format: Refer to [I-D.ietf-cats-metric-definition] Section 4.4.2

Score semantics: 0-3 (Low composed capability, not recommended for steering), 4-7 (Medium composed capability, optional for steering), 8-10 (High composed capability, priority for steering)

6.5.4.3. Metric Units

Unitless

6.5.4.4. Calibration

Calibration method: Conduct benchmark calibration based on representative end-to-end test profiles (fixed request mixes and controlled network/compute conditions) to align the mapping from contributing metrics to the Level 1 composed score. The calibration goal is to minimize score deviation across measurement agents and computation points within the same administrative domain (e.g., less than 0.1 over repeated test rounds). Calibration MAY include failure and saturation scenarios (e.g., compute overload, network congestion, and dependency failures) to ensure the composed score behavior is consistent with operational intent.

6.5.5. Administrative Items

6.5.5.1. Status

Current

6.5.5.2. Requester

To-be-assigned

6.5.5.3. Revision

1.0

6.5.5.4. Revision Date

2026-01-20

6.5.5.5. Comments and Remarks

None

7. Security Considerations

The CATS metrics defined in this document are dynamic and potentially sensitive. To prevent stability attacks (e.g., rapid metric churn), implementations MUST support aggregation, dampening, and threshold-triggered updates. To protect against disclosure or tampering, metric collection and distribution MUST use encryption, integrity protection, and authentication among C-SMA, C-NMA, and receivers. C-SMAs MUST authenticate the service instances they report on. False

reporting SHOULD be mitigated via secondary validation.

8. IANA Considerations

This document defines several CATS metric registry entries. IANA is requested to create a new registry titled "CATS Metrics" under a new "Computing-Aware Traffic Steering (CATS)" heading.

The initial entries for this registry are defined in Section 6 as follows:

Section 6.1: CATS L2 Metric Registry Entry

Section 6.2: CATS L1 Metric Registry Entry: Computing

Section 6.3: CATS L1 Metric Registry Entry: Communication

Section 6.4: CATS L1 Metric Registry Entry: Service

Section 6.5: CATS L1 Metric Registry Entry: Composed

For each entry, IANA is requested to assign a unique Identifier (defined in each subsection) from the registry’s assignment pool.

All metric entries have the following common attributes: Name, URI, Description, Change Controller (IETF), and Version. The naming convention and structure follow the definitions in each respective subsection of Section 6.

9. References

9.1. 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-24, 2 April 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-24>.

[I-D.ietf-cats-metric-definition]
Yao, K., Li, C., Contreras, L. M., Ros-Giralt, J., and G. Zeng, "CATS Metrics Definition", Work in Progress, Internet-Draft, draft-ietf-cats-metric-definition-06, 1 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-metric-definition-06>.

[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>.

[RFC5835] Morton, A., Ed. and S. Van den Berghe, Ed., "Framework for Metric Composition", RFC 5835, DOI 10.17487/RFC5835, April 2010, <https://www.rfc-editor.org/rfc/rfc5835>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/rfc/rfc6241>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/rfc/rfc7011>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/rfc/rfc7471>.

[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>.

[RFC8911] Bagnulo, M., Claise, B., Eardley, P., Morton, A., and A. Akhter, "Registry for Performance Metrics", RFC 8911, DOI 10.17487/RFC8911, November 2021, <https://www.rfc-editor.org/rfc/rfc8911>.

[RFC8912] Morton, A., Bagnulo, M., Eardley, P., and K. D'Souza, "Initial Performance Metrics Registry Entries", RFC 8912, DOI 10.17487/RFC8912, November 2021, <https://www.rfc-editor.org/rfc/rfc8912>.

[RFC9439] Wu, Q., Yang, Y., Lee, Y., Dhody, D., Randriamasy, S., and L. Contreras, "Application-Layer Traffic Optimization (ALTO) Performance Cost Metrics", RFC 9439, DOI 10.17487/RFC9439, August 2023, <https://www.rfc-editor.org/rfc/rfc9439>.

[RFC9911] Sch旦nw辰lder, J., Ed., "Common YANG Data Types", RFC 9911, DOI 10.17487/RFC9911, December 2025, <https://www.rfc-editor.org/rfc/rfc9911>.

9.2. Informative References

[DMTF] "DMTF", 1998, <https://www.dmtf.org/>.

[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>.

[Min-max-sigmoid]
"Data Mining: Concepts and Techniques (Fourth Edition)", 2023, <https://doi.org/10.1016/C2013-0-18660-6>.

[performance-metrics]
"performance-metrics", 19 March 2020, <https://www.iana.org/assignments/performance-metrics/performance-metrics.xhtml>.

[Prometheus]
"Prometheus", 2012, <https://prometheus.io/>.

Appendix A. Level 0 Metric Examples

Several definitions have been developed within the compute and communication industries, as well as through various standardization efforts---such as those by the [DMTF]---that can serve as Level 0 metrics. This section provides illustrative examples.

A.1. Compute Raw Metrics

This section uses CPU frequency as an example to illustrate the representation of raw computing metrics. The metric type is labeled as compute_CPU_frequency, with the unit specified in GHz. The format supports floating-point values. The corresponding metric fields are defined as follows:

Fields:

Metric_Type: compute_CPU_frequency
Level: Level 0
Format: floating point
Length: four octets
Unit: GHz
Source: nominal
Value: 2.2

Figure 9: An Example for Compute Raw Metrics

A.2. Communication Raw Metrics

This section takes the total transmitted bytes (TxBytes) as an example to show the representation of communication raw metrics. TxBytes are named as "communication type_TxBytes". The unit is Mega Bytes (MB). Format is unsigned integer. It will occupy 4 octets. The source of the metric is "Directly measured" and the statistics is "mean". Example:

Fields:

```
Metric_Type: "communication type_TXBytes"
Level: Level 0
Format: unsigned integer
Length: four octets
Unit: MB
Source: Directly measured
Statistics: mean
Value: 100
```

Figure 10: An Example for Communication Raw Metrics

A.3. Delay Raw Metrics

Delay is a kind of synthesized metric which is influenced by computing, storage access, and network transmission. Usually delay refers to the overal processing duration between the arrival time of a specific service request and the departure time of the corresponding service response. It is named as "delay_raw". The format supports floating point. Its unit is microseconds, and it occupies 4 octets. For example:

Fields:

```
Metric_Type: "delay_raw"
Level: Level 0
Format: floating point
Length: four octets
Unit: microsecond
Source: aggregation
Statistics: max
Value: 231.5
```

Figure 11: An Example for Delay Raw Metrics

Contributors

Mohamed Boucadair
Orange
Email: mohamed.boucadair@orange.com

Zongpeng Du
China Mobile
Email: duzongpeng@chinamobile.com

Hang Shi
Huawei
Email: shihang9@huawei.com

Authors' Addresses

Kehan Yao
China Mobile
China
Email: yaokehan@chinamobile.com

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

L. M. Contreras
Telefonica
Email: luismiguel.contrerasmurillo@telefonica.com

Jordi Ros-Giralt
Qualcomm Europe, Inc.
Email: jros@qti.qualcomm.com

Guanming Zeng
Huawei Technologies
China
Email: zengguanming@huawei.com