Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026

K. Yao
China Mobile
G. Zeng
Huawei

2 March 2026

GRASP Extensions for CATS Metrics Distribution
draft-yao-anima-grasp-cats-metrics-distribution-00

Abstract

Computing-Aware Traffic Steering (CATS) requires distribution of computing metrics across the network to enable efficient traffic steering decisions. The Generic Autonomic Signaling Protocol (GRASP) provides a distributed approach for autonomic node discovery, state synchronization, and parameter negotiation in Autonomic Networking (AN). This document defines extensions to the GRASP protocol to support the distribution of CATS metrics, by specifing the GRASP Objective definition for CATS metrics, structured encodings, distribution mechanisms tailored for dynamic and distributed network scenarios such as edge computing.

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 3 September 2026.

Copyright Notice

Table of Contents

1. Introduction

The Computing-Aware Traffic Steering (CATS) framework [I-D.ietf-cats-framework] aims to optimize traffic steering to service instances by jointly considering dynamic computing resources and network states. A key enabler for CATS is the standardized distribution of CATS metrics, which are defined in [I-D.ietf-cats-metric-definition]. [I-D.ietf-cats-framework] defines a distributed model where CATS metrics need to be disseminated distributedly across. The distributed model is adaptive to dynamic network scales.

The Generic Autonomic Signaling Protocol (GRASP) [RFC8990] is the core signaling protocol of the Autonomic Networking Integrated Model and Architecture (ANIMA) [RFC8993], providing native support for autonomic node discovery, peer-to-peer state synchronization, and hop-by-hop flooding without manual configuration. GRASP is deployed over the Autonomic Control Plane (ACP) [RFC8994], which offers secure, hop-by-hop authenticated and encrypted communication. This provides trusted metric distribution. While GRASP supports generic signaling, it lacks a standardized structure for encoding and distributing CATS metrics.

This document extends GRASP to address the above gap by: ** Defining a globally unique GRASP Objective for CATS metrics distribution;

** Specifying a Concise Binary Object Representation (CBOR) [RFC8949]-encoded structured data format for encapsulating CATS L0/L1/L2 metrics, aligned with the CATS metrics framework [I-D.ietf-cats-metric-definition];

** Standardizing two GRASP-based CATS metrics distribution mechanisms: active flooding (for proactive metric dissemination) and on-demand synchronization (for reactive metric retrieval);

** Defining message formats, processing behaviors, and cache management rules for GRASP nodes handling CATS metrics.

The extensions defined in this document are compatible with existing GRASP protocol semantics [RFC8990] and GRASP information distribution extensions [I-D.ietf-anima-grasp-distribution], and can be deployed in distributed, dynamic network scenarios such as edge computing, and intelligent transportation.

2. Definition of Terms

This document reuses the terms defined in [RFC8990], [RFC8993], [I-D.ietf-cats-framework], and [I-D.ietf-cats-metric-definition].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when, and only when, they appear in all capitals, as shown here.

3. GRASP Objective Definition for CATS Metrics

3.1. Objective Name

The GRASP Objective name for CATS metrics distribution is a unique URI-formatted string in local zone to avoid namespace conflicts, defined as:

local-zone:cats-metric

This name MUST be used by all ASAs when transmitting or receiving CATS metrics via GRASP. The namespace "local-zone" is reserved for CATS GRASP Objective within the specific area, and "cats-metric" identifies the objective as CATS metrics distribution.

3.2. Objective Data Structure

The GRASP Objective value for "local-zone:cats-metric" is a CBOR map that encapsulates CATS metrics data and all associated metadata, aligned with the CATS metrics framework [I-D.ietf-cats-metric-definition]. The map is composed of mandatory fields that are REQUIRED for all metric levels and optional fields that are conditionally present based on the metric level or deployment needs.

All field names and values follow CBOR type specifications in [RFC8949]. CATS metrics distribution GRASP Objective contains the following fields, with each field having "filed name", "CBOR type", "mandatory/Optional", "Description", "Applicable Metric Level", and an example for illustration.

CATS Service Contact Instance ID (CSCI-ID): Text, mandatory. It is a unique identifier of the compute node providing the metric. The applicable metric levels are L0, L1, and L2. An example is an IPv6 address, "2001:db8:1::100"

Metric_Type: Text, mandatory. CATS metric type per [I-D.ietf-cats-metric-definition]. The applicable metric levels are L0, L1, and L2. An example is "compute_norm"

Metric_Level: Text, mandatory. CATS metric level per [I-D.ietf-cats-metric-definition]. The applicable metric levels are L0, L1, and L2. An example is "L1".

Metric_Format: Text, mandatory. Data format of the metric value. The applicable metric levels are L0, L1, and L2. An example is "unsigned integer".

Metric_Value: Number, mandatory. Numeric value of the CATS metric (integer or float per Metric_Format). The applicable metric levels are L0, L1, and L2. An example is 8 (L1 metric), 2.8 (L0 metric--CPU GHz).

Metric_Unit: Text, optional. Unit of the metric. The applicable metric level is L0. An example is "GHz", "TFlops", "us".

Metric_Source: Text, optional. Source of the metric. The applicable metric levels are L0, L1, and L2. An example is "normalization"

Metric_Statistics: Text, optional. Statistical type of the metric. The applicable metric levels are L0, L1, and L2. An example is "max".

Timestamp: Integer, mandatory. Unix timestamp (in seconds) when the metric was collected. The applicable metric levels are L0, L1, and L2. An example is 1719283200.

Expiry: Integer, mandatory. Validity period of the metric (in seconds), and the metric is considered stale after Timestamp + Expiry. The applicable metric levels are L0, L1, and L2. An example is 15.

The following is a CBOR encoding example of the CATS metric distribution GRASP Objective:

CBOR

{

"CSCI-ID": "2001:db8:1::100",

"Metric_Type": "compute_norm",

"Metric_Level": "L1",

"Metric_Format": "unsigned integer",

"Metric_Value": 8,

"Metric_Source": "normalization",

"Timestamp": 1719283200,

"Expiry": 15

}

4. CATS Metrics Distribution Mechanisms

This document defines two core GRASP-based distribution mechanisms for CATS metrics. Both extend existing GRASP message types defined in [RFC8990].

** Active Flooding: Proactive dissemination of CATS metrics to all adjacent ASAs via GRASP M_FLOOD messages, suitable for real-time metric updates in dynamic networks.

** On-Demand Synchronization: Reactive retrieval of CATS metrics via GRASP M_REQ_SYN (request) and M_SYNCH (response) messages, which is suitable for nodes that need targeted metric data for traffic steering decisions, and occassions that don’t require periodical metric update.

4.1. Pre-configuration

Both mechanisms require some pre-configurations.

1. ASA Deployment: compute nodes where CATS service contact instances are located and network nodes MUST deploy ASA that implements the GRASP extensions defined in this document.

2. ACP Establishment: ASAs MUST establish a secure ACP channel with adjacent ASAs for GRASP message signaling.

3. Metric Collection: ASAs at compute nodes MUST collect CATS metrics following [I-D.ietf-cats-metric-definition] and update the metrics at a configurable interval.

4.2. Method 1: Active Flooding

Active flooding uses the GRASP M_FLOOD message type (message type code: 9 [RFC8990]) to proactively distribute CATS metrics from ASAs at compute nodes to all adjacent ASAs. Adjacent ASAs update the metric and re-flood the message to their neighbors, up to a configurable TTL value. This mechanism is suitable for real-time, global dissemination of CATS metrics in dynamic networks with frequent resource state changes.

Message Format:

The M_FLOOD message for CATS metrics is a CBOR array per [RFC8990], with the payload extended to include the CATS metrics Objective and its structured data. The message format is defined as:

CBOR

[

9, ; GRASP message type: M_FLOOD (fixed = 9)

generate_msg_id(), ; Message ID: globally unique (timestamp + CSCI-ID hash)

local-ipv6-cbor, ; Sender’s IPv6 address (CBOR byte string)

```
   15000, ; TTL: flood scope (ms, Default: 15000)

   [ ; GRASP Objective List (single Objective for CATS)

  "local-zone:cats-metric",   ; Objective Name (fixed)

  5,           ; Objective Flags: F_SYNCH_bits (0x5, per {{RFC8990}})

  3,           ; Loop Count: retry count (fixed = 3 for CATS)

  cbor-serialize(cats-metric)  ; CATS Metric Data after CBOR serialization

   ],

   [] ; Locator List: empty (flooding does not require locators)

   ]
```

Basic Workflow for M_Flood

The basic workflow of M_Flood message for actively flooding CATS metrics Objective is defined in Figure 1. There are the following processing steps:

1. Message Reception: ASA receives a M_FLOOD message with the CATS Objective, validates the message format (CBOR array, correct message type, Objective name) and ACP security (authenticated/ encrypted). Discard invalid or unauthenticated messages.

2. TTL Check: Decrement the TTL by the local processing delay (in ms). If the resulting TTL is equal to or less than 0, discard the message and end flooding.

3. Staleness Check: Extract the "Timestamp" and "Expiry" fields from the CATS metric data, and calculate the validity window. If the current time > Timestamp + Expiry, discard the stale metric.

4. Cache Check & Update: a. Check the local CATS metrics cache for an entry with the same "CSCI_ID" and "Metric_Type".

b. If no entry exists: add the metric to the cache, mark with the validity window, and proceed to re-flood.

c. If an entry exists: compare the "Timestamp" of the received metric with the cached one. If the received "Timestamp" is newer: update the cache with the new metric; if older: discard the received metric (no re-flood).

1. Re-Flood: Re-transmit the M_FLOOD message (with the decremented TTL) to all adjacent ASAs (excluding the sender) via the ACP channel.

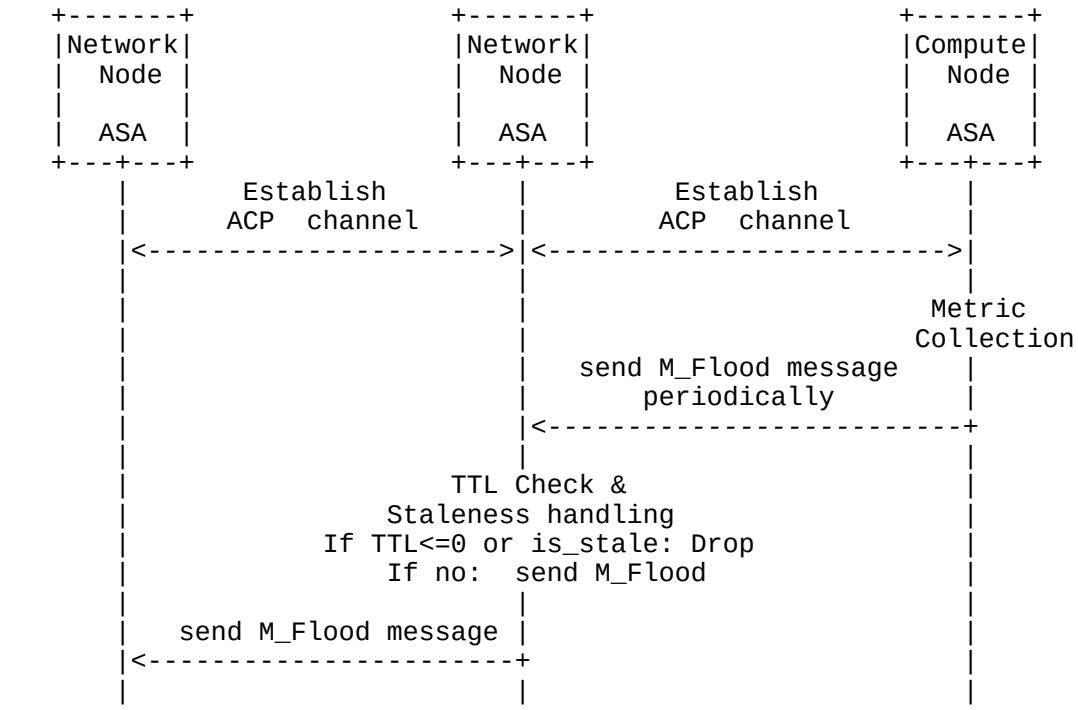

Figure 1: Active Flooding

4.3. Method 2: On-Demand Synchronization

On-demand synchronization uses two GRASP message types to enable reactive retrieval of CATS metrics. M_REQ_SYN (synchronization request, message type code: 4 [RFC8990]) and M_SYNCH (synchronization response, message type code: 8 [RFC8990]). A metric consumer ASA (e.g., CATS Ingress Node) sends a M_REQ_SYN message to a specific compute node ASA to request its current CATS metrics. The compute node ASA responds with a M_SYNCH message containing the latest metric data which is collected just after request time. This mechanism is suitable for nodes that need targeted, real-time CATS metrics for traffic steering decisions and no need for global flooding.

Message Format:

Both M_REQ_SYN and M_SYNCH messages are CBOR arrays per [RFC8990], with the payload extended to include the CATS Objective. The request and response are message ID-bound: the M_SYNCH message reuses the M_REQ_SYN message ID to ensure correlation.

The message format of M_REQ_SYN is defined as:

CBOR

[

4, ; GRASP message type: M_REQ_SYN (fixed = 4)

generate_msg_id(), ; Request Message ID: globally unique (for response correlation)

[; GRASP Objective List

"local-zone:cats-metric", ; Objective Name (fixed)

5, ; Objective Flags: F_SYNCH_bits (0x5, per {{RFC8990}})

3, ; Loop Count: retry count (fixed = 3 for CATS)

0 ; ’0’ is an initial value that means CATS metrics request

]

]

The message format of M_SYNCH is defined as:

CBOR

[8, ; GRASP message type: M_SYNCH (fixed = 8)

req_msg_id(), ; Response Message ID: REUSE the request’s Message ID (correlation)

[; GRASP Objective List

"local-zone:cats-metric", ; Objective Name (fixed)

5, ; Objective Flags: F_SYNCH_bits (0x5, per {{RFC8990}})

3, ; Loop Count: match the request’s Loop Count

cbor_serialize(cats-metric) ; CATS Metric Data

]

]

Basic Workflow for M_REQ_SYN and M_SYNCH

Consumer ASA (Requestor) Processing Steps:

** Request Generation: Consumer ASA generates an M_REQ_SYN message with a unique message ID and the target "CSCI-ID" and "Metric_Type". It then unicasts the message to the compute node ASA via the ACP channel.

** Response Reception: Waits for an M_SYNCH response with the same message ID (configurable timeout, Default: 5s). If no response is received within the timeout, retry the request up to the Loop Count (3) times.

** Response Validation: Validates the M_SYNCH message format, Objective name, and metric freshness (Timestamp + Expiry >= current time). Discard invalid or stale responses.

** Metric Usage: Extracts the CATS metric data from the M_SYNCH payload and uses it for CATS traffic steering decisions.

Compute Node ASA (Responder) Processing Steps:

** Request Reception: Receives an M_REQ_SYN message; validates the message format, Objective name, ACP security, and target CSCI-ID. Discard invalid or unauthenticated requests.

** Real-Time Metric Collection: Collects the latest local CATS metric data for the requested Metric_Type (or all metrics) per [I-D.ietf-cats-metric-definition]. It then sets the "Timestamp" to the collection time and "Expiry" to 15s (Default value).

** Response Generation: Generates an M_SYNCH message with the same message ID as the request; encodes the collected metric data into the CATS Objective CBOR map.

** Response Transmission: Unicasts the M_SYNCH message to the consumer ASA via the ACP channel.

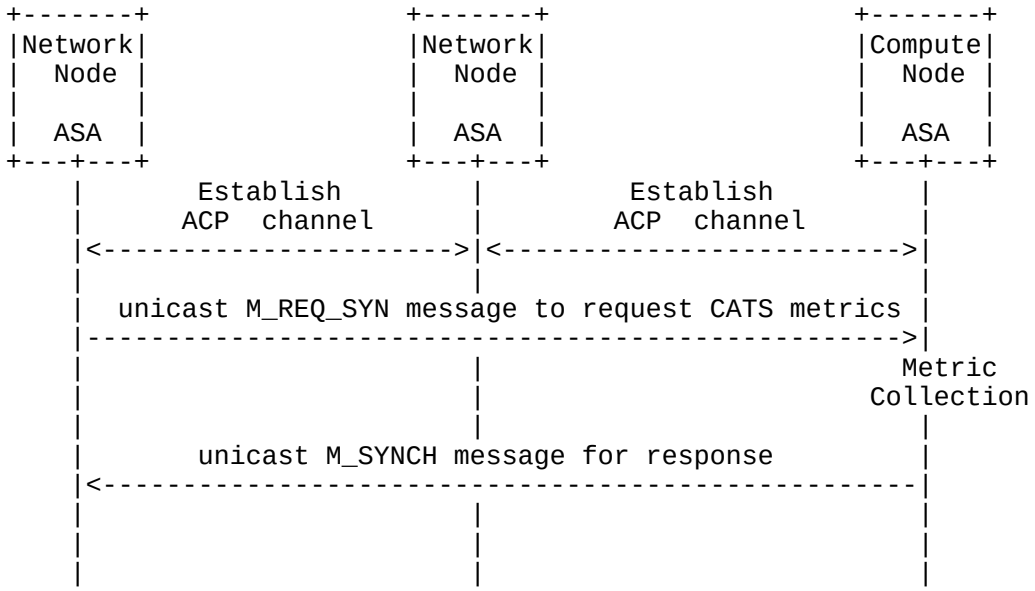

Figure 2: On-Demand Synchronization

5. Security Considerations

All CATS metrics distribution messages defined in this document are transmitted over the ACP channel [RFC8994], which provides mandatory hop-by-hop authentication, encryption, and integrity protection. No additional security mechanisms are required for the metrics messages themselves, as the ACP addresses the following core security concerns:

** Authentication: All ASAs are authenticated via LDevID certificates that are provisioned by Bootstrapping Remote Secure Key Infrastructure (BRSKI) [RFC8995]. Unauthenticated ASAs cannot join the ACP or receive/transmit GRASP messages.

** Confidentiality: GRASP messages that include CATS metrics are encrypted hop-by-hop via the ACP, which limits eavesdropping on metric data.

** Integrity: ACP provides message integrity protection. Tampered CATS metrics messages are detected and discarded by ASAs.

Some additional security considerations:

** Flooding Scope Limitation: ASAs SHOULD use the TTL field to limit the flooding scope of CATS metrics. This reduces the attack surface for denial-of-service (DoS) attacks via excessive flooding.

** Cache Eviction: ASAs SHOULD enforce strict cache eviction rules to prevent cache exhaustion attacks via spoofed CATS metrics messages.

6. IANA Considerations

This document requests IANA to make the registrations of the GRASP Objective for CATS metrics distribution. Details to-be-added.

7. Acknowledgements

8. Informative References

[I-D.ietf-anima-grasp-distribution]
Jiang, S., Liu, B., Xiao, X., Hecker, A., and X. Zheng, "Information Distribution over GRASP", Work in Progress, Internet-Draft, draft-ietf-anima-grasp-distribution-12, 11 December 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-grasp-distribution-12>.

[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-20, 26 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-20>.

[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-05, 2 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-metric-definition-05>.

[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>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

[RFC8990] Bormann, C., Carpenter, B., Ed., and B. Liu, Ed., "GeneRic Autonomic Signaling Protocol (GRASP)", RFC 8990, DOI 10.17487/RFC8990, May 2021, <https://www.rfc-editor.org/rfc/rfc8990>.

[RFC8993] Behringer, M., Ed., Carpenter, B., Eckert, T., Ciavaglia, L., and J. Nobre, "A Reference Model for Autonomic Networking", RFC 8993, DOI 10.17487/RFC8993, May 2021, <https://www.rfc-editor.org/rfc/rfc8993>.

[RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <https://www.rfc-editor.org/rfc/rfc8994>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

Authors' Addresses

Kehan Yao
China Mobile
Email: yaokehan@chinamobile.com

Guanming Zeng
Huawei
Email: zengguanming@huawei.com