Computing-Aware Traffic Steering
Internet-Draft
Intended status: Informational
Expires: 16 November 2025

H. Shi
C. Li
Huawei Technologies
Z. Du
China Mobile
X. Yi
China Unicom
T. Yang
China Broadcast Mobile Network Company

15 May 2025

Design analysis of methods for distributing the computing metric draft-shi-cats-analysis-of-metric-distribution-04

Abstract

This document analyses different methods for distributing the computing metrics from service instances to the ingress router.

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-method-analysis.

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 16 November 2025.

Table of Contents

1. Introduction

Many modern computing services are deployed in a distributed way. Multiple service instances deployed in multiple sites provide equivalent function to the end user. As described in [I-D.yao-cats-ps-usecases], traffic steering that takes computing resource metrics into account would improve the quality of service. Such computing metrics are defined in [I-D.du-cats-computing-modeling-description]. This document analysis different methods for distributing these metrics.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses terms defined in [I-D.ldbc-cats-framework]. We list them below for clarification.

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

* CATS Service Metric Agent (C-SMA):Responsible for collecting service capabilities and status, and reporting them to a CATS Path Selector (C-PS).

* CATS Path Selector (C-PS): An entity that determines the path toward the appropriate service location and service instances to meet a service demand given the service status and network status information.

3. Requirement of distributing computing metric

The CATS functional components are defined in [I-D.ldbc-cats-framework](see Figure 1, the figure is replicated here for better understanding). C-SMA is responsible for collecting the computing metrics of the service instance and distributing the metrics to the C-PSes. A C-PS then selects a path based on the computing metrics and network metrics.

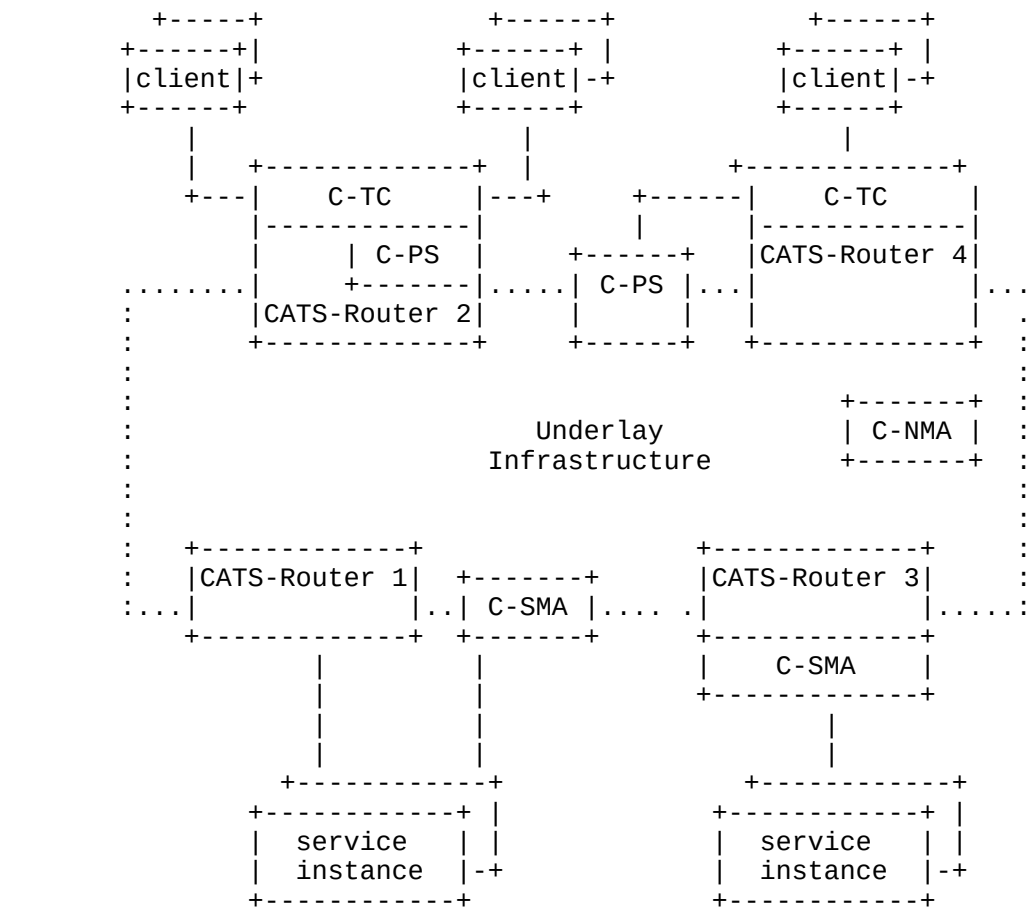

Figure 1: CATS Functional Components

4. Overhead of Metric Distribution

In the context of metric distribution for CATS, whether in a distributed or centralized architecture, one of the key considerations is the overhead involved in transferring metrics between the producers (C-SMA) and consumers (C-PS). This overhead can be defined by the following equation:

Metric Distribution Overhead = Number of Producers × Number of Consumers × Distribution Frequency × Metric Size

The number of producers and consumers is dictated by the scope of the metric distribution. Not all ingress routers need to be aware of the computing metrics from all sites; typically, it is sufficient to distribute metrics only for nearby or relevant sites to optimize scheduling. Additionally, the distribution frequency can be optimized by sending metric updates only when there are significant changes in the metrics, reducing unnecessary transmissions and minimizing overhead while maintaining the accuracy of the scheduling process.

5. Choice 1: Centralized versus Dencentralized

5.1. Option 1: Centralized C-SMA + Centralized C-PS

The computing metrics can be collected internally with a hosting infrastructure by a centralized monitor of the hosting infrastructure. Various tools such as Prometheus can serve this purpose. The monitor can pass the metrics to a network controller, which behaves as a C-PS. Then, the network controller calculates the optimal path and distribute the paths to CATS ingress routers. When a service request arrives at the CATS ingress router, it just steers the request to the path. The network controller distributed the metric update to the C-PS using south-bound protocol.

5.2. Option 2: Centralized C-SMA + Distributed C-PS

Similar to option 1, the network controller does not calculate the path. It just passes the computing metrics received from the cloud monitor to the C-PS embedded in a CATS ingress router. The C-PS at each CATS ingress router will proceed with path computation locally.

5.3. Option 3: Distributed C-SMA + Centralized C-PS

The C-SMA can be deployed in a distributed way. For example, C-SMA running at each site collects the computing metrics of the service instances running in a site. Then, it reports the metrics to a network controller, which behaved as a C-PS. The network controller calculates the best path for a service and distribute the path to a CATS ingress router.

5.4. Option 4: Distributed C-SMA + Distributed C-PS

Similar to option 3, each C-SMA collects the computing metrics of each site. Then it needs to distribute the metric to C-PS at each ingress router. It can do so directly or through a network controller.

5.5. Comparaison

	Option 1	Option 2	Option 3	Option 4
Protocol	None	Southbound	Southbound	Southbound or Eastbound
CATS router requirement	Low	High	Low	High
Network controller requirement	High	Low	High	Low

Table 1: Comparison between different option

6. Choice 2: Push versus Pull

There are two primary modes of the metric distribution: push and pull modes. The push mode operates on the principle of immediate dissemination of computing metrics as soon as they are refreshed. This approach boasts the advantage of timeliness, ensuring that the latest metrics are always available at the cost of frequent updates. The frequency of these updates directly correlates with the rate at which the computing metrics are refreshed.

Conversely, the pull mode adopts a more reactive strategy, where the latest computing metrics are fetched only upon receiving a specific request for them. This means that the distribution frequency of computing metrics hinges on the demand of such data, determined by the frequency of incoming service request from each ingress.

Irrespective of the chosen mode, various optimization techniques can be employed to regulate the frequency of metric distribution effectively. For instance, in the push mode, setting thresholds can mitigate the rate of updates by preventing the dispatch of new computing metrics unless there is a significant change in the metrics. This approach reduces unnecessary network traffic and computational overhead but at the potential cost of not always having the most up-to-date information.

In the pull mode, caching the returned computating metric for a predetermined duration offers a similar optimization. This method allows for the reuse of previously fetched data, delaying the need for subsequent requests until the cache expires. While this reduces

the load, it introduces a delay in acquiring the latest computing metrics, possibly affecting decision-making processes that rely on the most current data.

Both push and pull models, despite their inherent differences, share a common challenge: striking a balance between the accuracy of the distributed computating metrics and the overhead associated with their distribution. Optimizing the distribution frequency through techniques such as threshold setting or caching can help mitigate these challenges. However, it's important to acknowledge that these optimizations may compromise the precision of scheduling tasks based on these metrics, as the very latest information may not always be available. This trade-off necessitates a careful consideration of the specific requirements and constraints of the computational environment to determine the most suitable approach.

7. Choice 3: Aggregation of metric update messages

Another crucial aspect to consider in the distribution of computing metrics is the potential for aggregating updates. Specifically, in distributed C-SMA scenarios, where an Egress point connects to multiple sites, it's feasible to consolidate updates from these sites into a single message. This aggregation strategy significantly reduces the number of individual update messages required, streamlining the process of disseminating computing metric.

Aggregation can be particularly beneficial in reducing network congestion and optimizing the efficiency of information distribution. By bundling updates, we not only minimize the frequency of messages but also the associated overheads, such as header information and protocol handling costs. This approach is not limited to distributed environments but is equally applicable in centralized C-SMA scenarios.

In centralized C-SMA scenarios, a controller responsible for managing computing metric updates to ingress nodes can employ a similar aggregation technique. By consolidating updates for multiple sites into a single message, the system can significantly decrease the overhead associated with update protocol messages.

While aggregating computing metrics offers substantial benefits in terms of reducing network traffic and optimizing message efficiency, it's important to address a specific challenge associated with this approach: the potential delay in message timeliness due to the waiting period required for aggregation. In scenarios where computing metrics from multiple nodes are consolidated into a single update message, the updates from individual nodes might not arrive simultaneously. This discrepancy can lead to situations where updates must wait for one another before they can be aggregated and sent out.

This waiting period introduces a delay in the dissemination of computing metrics, which, while beneficial for reducing the volume of update messages and network overhead, can inadvertently affect the system's responsiveness. The delay in updates might not align well with the dynamic needs of computing resource management, where timely information is crucial for making informed decisions about resource allocation and load balancing.

Therefore, while the aggregation of updates is an effective strategy for enhancing the efficiency of computing metrics distribution, it necessitates a careful consideration of its impact on the system's ability to respond to changes in computing needs promptly. Balancing the benefits of reduced message frequency and overhead with the potential delays introduced by aggregation requires a nuanced approach. This might involve implementing mechanisms to minimize waiting times, such as setting maximum wait times for aggregation or dynamically adjusting aggregation strategies based on the current load and the arrival patterns of updates.

8. Security Considerations

TBD

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

10.2. Informative References

[I-D.du-cats-computing-modeling-description]
Du, Z., Yao, K., Li, C., Huang, D., and Z. Fu, "Computing Information Description in Computing-Aware Traffic Steering", Work in Progress, Internet-Draft, draft-du-cats-computing-modeling-description-03, 6 July 2024, <https://datatracker.ietf.org/doc/html/draft-du-cats-computing-modeling-description-03>.

[I-D.ldbc-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-ldbc-cats-framework-06, 8 February 2024, <https://datatracker.ietf.org/doc/html/draft-ldbc-cats-framework-06>.

[I-D.yao-cats-ps-usecases]
Yao, K., Trossen, D., Boucadair, M., Contreras, L. M., Shi, H., Li, Y., and S. Zhang, "Computing-Aware Traffic Steering (CATS) Problem Statement, Use Cases, and Requirements", Work in Progress, Internet-Draft, draft-yao-cats-ps-usecases-03, 30 June 2023, <https://datatracker.ietf.org/doc/html/draft-yao-cats-ps-usecases-03>.

Acknowledgments

The author would like to thank Xia Chen, Guofeng Qian, Haibo Wang for their help.

Authors’ Addresses

Hang Shi
Huawei Technologies
China
Email: shihang9@huawei.com

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

Zongpeng Du
China Mobile
Email: duzongpeng@foxmail.com

Xinxin Yi
China Unicom
Email: yixx3@chinaunicom.cn

Tianle Yang
China Broadcast Mobile Network Company
China
Email: yangtianle@10099.com.cn