Computing-Aware Traffic Steering(CATS)
Internet-Draft
Updates: 2024-12-26 (if approved)
Intended status: Informational
Expires: 29 June 2025

T. Fu, Ed.
H. Zhang, Ed.
China Academy of Information and Communications Technology
J. Wang, Ed.
China Mobile

26 December 2024

Problem statements and requirements of Deterministic CATS on the Industrial Internet
draft-ftzhs-cats-industrial-requirement-01

Abstract

The Industrial Internet is a new infrastructure, application mode and industrial ecology with the deep integration among the new information technology, communication technology and the industrial economy.Industrial production tasks are time-sensitive, which put forward high requirements on networks and applications, and need to meet the deterministic requirements in terms of delay, jitter, reliability, etc. Industrial deterministic service refers to a closed loop composed of communication paths and control processes in which two or more applications participate.Industrial management platforms need to unify network forwarding and computing tasks for each deterministic service. This draft illustrates use cases of traffic steering for deterministic service in terms of dynamic computing and networking resource status,together with the requirements and solutions for CATS(Computing-Aware Traffic Steering).

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 June 2025.

Copyright Notice

Table of Contents

1. Introduction

1.1. Backgroud

TBA.

1.2. Requirements Language

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.

2. Definition of Terms

TBA.

3. Problem Statement of Industrial CATS

3.1. Industrial production service

In the Industrial Internet, the performance requirements of production processes are much higher than the Internet. Therefore, the industrial network naturally needs the support of compute-aware traffic steering. Application services are dynamically adapted to industrial scenarios, tasks, and resources. The computation that devices participate in evolves from simple control logic to complex big data decision-making. In the application layer of industrial Internet, deterministic service is the combination of network communication process and calculation process. It refers to a closed loop formed by one or more application communication links and control links. Industrial CATS needs to manipulate the all steps of the industrial production cycle from the service-initiating devices through remote devices with dependencies, such as edge computing or cloud services. In this process, the forwarding and calculation of data must comply with the performance requirements of the service-initiating device, such as delay, jitter, reliability, packet loss rate, etc. TBA.

3.2. Deterministic Industrial Production Service

In smart factories, there are mass production systems, edge computing, industrial clouds, remote communication relationships between various applications, and various services share network queues and computing resources. In order to satisfy the requirements of time-sensitive industrial production services, it is necessary to realize the deterministic management of computing power and network resources based on CATS. The concurrent processing of multiple services must ensure the strict requirements of delay, jitter, sequence, and reliability. At present, it is feasible to use deterministic network and edge computing to ensure multi-service load with millisecond delay through clock synchronization and resource reservation. By strengthening the bidirectional perception of computing power and network, all kinds of resources are uniformly adjusted, and the compromise between computing performance and network communication performance is achieved on various resource-competing devices.TBA.

3.3. Deterministic CATS

Deterministic CATS is adjusting network forwarding configurations according to the computing requirements. Taking the common visual detection scene in industry as an example, industrial robot arm and edge computing are interconnected by deterministic network, which involves the application of industrial robot-arm and image processing application of edge computing. The robot arm periodically collects high-definition images of the parts or products being machined and sends the data to matched edge computing device. Then, edge computing device feeds the results back to the industrial robot-arm after processing. The traditional network can only control the round-trip transmission process. If the edge side blocks the visual detection task due to multiple services, the processing delay of the industrial robot arm will increase. In existing factories, one container or edge computing device is often configured for several tasks, resulting in a waste of entire plant resources. Deterministic CATS manages both the communication process and the calculation process, accurately ensuring the indicators of the entire visual inspection process from the perspective of an industrial robot arm, and then splits the overall indicators into the indicators of each step of the calculation process and the communication process according to the strategy. When multiple deterministic services run concurrently, this scheme can comprehensively schedule resources, do overall multi-objective optimization for multiple services, and appropriately adjust the communication cost and network cost ratio of each deterministic service.TBA.

4. Use Cases

4.1. Computing-Aware Industrial robots

The automatic manufacturing of soft materials has always been a difficult problem in industrial digitization. For the sake of unpredictable deformation of materials, it brings difficulties to the traditional equipments, unless it can recognize real-time states of the deformation for producets. Furthermore, productlines need to accurately perform operations and correct the negative effects of the deformation. The execution of complex assembly tasks in intelligent manufacturing requires the cooperation of multiple robots. With the improvement of industrial intelligence, the industrial robot can replace the manual handling of soft materials. In this case, the two robotic arms plan the operation simultaneously and can judge the status of the flexible material in real time. During the whole folding operation, the offline part updates the neural network periodically to optimize the parameters of the model. The online part periodly recognizes the image, and the robot arm will

continuously feed back the folding effect to the neural network, and judge whether to enter the next cycle on the basis of predicting the operation result and judging the folding effect. The total time is mainly limited by the recognition accuracy and recognition delay. If the computing power resources are enabled through edge computing, and the certainty of network edge communication, self-learning and image recognition is ensured through application-oriented deterministic control, the processing complexity, folding speed and accuracy of flexible objects can be further improved. The new production line of flexible material processing can gradually replace the production line workers, and the intelligence level and cooperation level of the overall certainty are increased through the cloud business collaboration at the network edge. With the continuous optimization of intelligent algorithms, the tightness of operations can be further enhanced to achieve progress from substitution to transcendence.

4.2. Computing-Aware vCloud terminial

Cloud services are control functions migrated from physical devices to Industril cloud, datacenter or edge, dramatically reducing production line costs. Traditional networks do not have the capability of application-oriented certainty assurance, usually directly by the engineer to estimate the processing and calculation time, subtract the estimated time with the control cycle, and finally determine the network certainty requirements, and the calculation part of the certainty is actually inaccurate or overreserved estimates. It will cause a waste of network or computing resources. The deterministic requirements of network cooperation are proposed to reflect the exact requirements from end to end at the application level, so as to realize the deterministic remote closed-loop control between control and execution. Meet the application's high performance requirements such as communication delay, bandwidth, and cloud computing power. Typical scenarios for cloud X services include on-site production line equipment control, robot control, automatic guided vehicle control, 5G PLC, etc. It has the characteristics of multi-network integration, broadening the acquisition channel, industrial equipment reusability, and improving the robustness of production network. PLC logic control has a fixed control cycle, assuming that the application needs to complete an IO data reading, processing and writing operations within a 10ms cycle. Motion control is a precision control business involving robots, servo motors and other equipment, which requires high deterministic capability of delay and jitter, with end-to-end delay to be controlled within 10ms and jitter less than 100us. Machine vision quality inspection is an intelligent quality inspection service involving image processing and analysis. It has high requirements for uplink bandwidth capability, and the network should provide uplink bandwidth greater than 80Mbps. Power differential protection

is a key service related to the safety and stable operation of the power grid. It has high requirements for delay and reliability. The service delay requirement is less than 15ms, and the reliability reaches 5 9s (99.999%).

4.3. Computing-Aware Multi-Application collaboration

With the increasing degree of networking and digitalization of industrial enterprises, limited by the 1-3 layer network certainty, the combination of machine learning, big data and other technologies with the production line is mainly direct deployment and simple interconnection, and the role of advanced algorithms in resource optimization allocation in the life cycle of industrial production is still not fully reflected. If the network with high certainty accuracy is equipped with IT services that try their best to deal with IT, it is not enough. There is a risk that some links will time out, making it difficult to accurately guarantee the complex production tasks of multi-equipment collaboration. Application oriented deterministic technology can solve this problem through multi-application collaborative deterministic global scheduling. Take the intelligent processing line in Figure 2 as an example. The edge equipment carrying IT technology runs a large number of new algorithms and models, and introduces new computing resources into the workshop network architecture through edge cloud facilities. In order to ensure the certainty of the entire production business, it is necessary to globally manage a series of applications and network transmission according to deterministic constraints to ensure that the overall service quality meets the needs of users. The deterministic global scheduling of multi-application collaboration needs to support the information interconnection between various systems at the data application level of the whole network, and regulate the real-time and reliability capabilities of all devices, so as to strictly meet the deterministic requirements of various applications. The certainty between applications actually includes logic, computing, network transmission and other links, and it is difficult to achieve the overall certainty guarantee by using only one level of certainty guarantee technology. In the future, the certainty of the application layer should be arranged from the network, calculation, logic and other aspects to achieve the overall certainty of the overall high service quality assurance. The certainty of multi-application collaboration can be comprehensively controlled for complex business, which is an important way to realize unmanned production line. Automatic application association analysis helps you connect service planning and device configuration, saving the cost of service configuration and change.

4.4. Industrial Digital Twins

Digital workshop is an information workshop designed and constructed by applying lean production, lean logistics, visual management, standardized management, green manufacturing and other advanced production control theories and methods. The application of a deterministic digital twin on the end side provides a comprehensive understanding of the details of the production environment situation. The generation of massive data is often conducive to the realization of big data decisions, but different types of data are often distributed in several independent information systems or in different parts of the production process. Through industrial Internet sensors and other data acquisition devices, valuable business data is continuously collected in different forms and on an unprecedented scale, and then uploaded to digital twin applications, and real-time graphical display and predictive analysis results can be effectively responded to emergency situations. The digital twin business composed of several links needs to guarantee deterministic parameters at the application level to ensure accurate and reliable projection of the whole plant in the information space. The utility of end-to-side data integration mainly comes from the massive application data collected, the diversity of application data and the accuracy of the collected data. The traditional data integration generally uses the industrial Ethernet private network to collect information, and then converges and processes it to the upper data center through the controller. The whole application to application link is the conversion of multiple system protocols and the interaction between multiple layers of applications. Among them, the industrial system detection, control, implementation of high real-time, industrial production site data has a large volume, and has real-time demand, some scenes real-time requirements within 10ms. At the same time, data collected from endpoints can introduce failure information, duplicate information, and other types of problems. The certainty in traditional data integration scenarios is usually guaranteed by the network layer, but the certainty between applications actually includes multiple links such as sequential processing, computing, and network transmission, and it is difficult to rely on a single network level guarantee capability to support the application-to-application data integration requirements.

4.5. Customized Production Lines

The new model of Customers to Production Lines (C2PL) applies the new technology of artificial intelligence to the layout of factory production activities, and pulls customized orders and flexible production with intelligent large models as the core. Support users to customize their favorite products on the official website, and freely play in terms of appearance, material, size, etc. Users

complete the customization, ordering and payment of high-end products on the Internet side, and the joint business arrangement of IT system and OT system is completed by industrial management software. Automatic formation of a temporary overall certainty, disassembly into multiple processing links and execution. C2PL realizes unmanned production line scheduling and can be widely applied to products with simple structure to improve user participation and demand collection efficiency. The existing ChatGPT can support voice, link, picture and other forms of data. After the system gives the effect diagram, the product form can be improved interactively. In this scenario, AI connects IT and OT workflows, and application-oriented determinism requires global quality of service definition, control, monitoring and evaluation. Support a variety of demand expression forms, voice, pictures, links and other needs can be understood by the system and present the sample effect; Through AI large model engine, order feature interpretation, deterministic production process creation, resource scheduling, etc. Ensure the industrial Internet certainty under the condition of product diversification; The overall determinism is disassembled into a set of processing steps of IT design link and OT production link through AI large model engine. Implement OT infrastructure to provide unified defined manufacturing services to the control system. After realizing standardized production links, application-oriented deterministic control can predict, adjust and monitor deterministic control results more accurately. Allocate temporary resources in the factory resources to complete the production task to ensure that the processing process can be completed in real time.

5. Requirements

5.1. Requirements for Deterministic CATS Service

a. The industial system needs a specific controller to unify scheduling of network resources and computing resources for deterministic services; b. It needs to establish independent network integration diagram for each deterministic service to accurately reflect "application-network" correlation; c. The performance indicators of deterministic services need to be converged within expected boundaries, such as the overall service completion delay, overall jitter, bandwidth, packet loss rate, etc. d. Industrial equipments, such as OT devices and IT devices, need information model to uniformly define the deterministic parameters of deterministic devices; e. All deterministic devices need to support and enable deterministic application and network deterministic control protocols; f. The deterministic management and control of deterministic computing tasks need to be supported, and the integrated scheduling policy of the computing network should be split

into the resource allocation of deterministic devices and delivered to the target devices through the deterministic southbound interface. g. The industial system requires deterministic execution of computing tasks . After receiving the deterministic index parameters of the deterministic computing tasks being executed, the deterministic device ensures the execution speed and output result quality of the computing tasks by means of scheduling priority, elastic allocation of computing resources, isolation of computing units, etc. h. The deterministic controller needs to support deterministic control of deterministic computing tasks, convert the integrated scheduling strategy of the computing network into resource allocation for deterministic devices, and issue it to the target device through southbound interfaces. i. It is necessary for deterministic controller to support the deterministic execution of deterministic computing tasks. After receiving the deterministic indicator parameters of the ongoing deterministic computing task, deterministic devices ensure the execution speed and output quality of the computing task through scheduling priorities, elastic allocation of computing resources, isolation of computing units, and other methods. j. The service needs to provide users with synchronization functions between applications, support the coordination of time sequence between applications, and ensure consistency in the entire deterministic system for applications.k.TBA.

5.2. Requirements for Deterministic Networks

a. In the case of data transmission with high throughput, it is required to implement multi-channel deterministic transmission through the deterministic transport layer protocol; b. Network is required to support fast session establishment, connection migration (wireless network), elastic congestion control; c. Network controller needs application-oriented subflow management; d. It needs cross-layer configuration and consistency of key parameters such as clock, cycle, data unit and priority; e. Deterministic networks need supporting cross-domain scheduling of data flows, which involves crossing networks at different levels and crossing boundary devices (devices that modify protocols such as gateways); f. Network controller is required to achieve model-based predictability for cross-layer, large-scale and heterogeneous networking, etc. (Predictable function is useful for automated configuration)g. The network needs to have time synchronization function. On the one hand, it provides synchronization support for upper layer application services. On the other hand, through the synchronization of network devices and terminals, it can support time slot scheduling of data traffic and improve the deterministic ability of network transmission. h. In the transmission of high reliability services, the network needs to have the function of multi-path redundant

transmission, which can support the transmission of business data in multiple paths, to ensure that when the determinacy of one link is difficult to meet, other links can still meet the transmission requirements of the business. i.The network can provide deterministic transmission capabilities based on application business requirements, including latency, bandwidth, reliability, etc., to allocate network resources and scheduling strategies for different business traffic needs, achieving on-demand transmission guarantee. j. TBA.

5.3. Requirements for Internal factory computing

a. Computing resources imformation, such as basic information, computing information, load information, task list, etc. are required to upload actively after computing equipments are registered to CATS controller; b. The computing equipment needs to regularly upload real-time information; c. CATS controller needs to support calculation force and valuation function, used to evaluate the scheduling results of own resources; d. CATS controller support billing and query of external computing resources; e. CATS controller needs supporting precise isolation for multi-core hardware, and support the mapping of some certain computing units to a deterministic computing task; f. Resource reservation for a deterministic computing task should be supported on computing devices; g. CATS controller support elastic resource expansion and contraction of containers; h. A deterministic information model for identifying deterministic services needs to be supported. i. Deterministic controller needs to support the cross domain interconnection of computing power and the long-distance lossless data transmission of computing power data. j. Deterministic controller needs to support public network computing power perception and network active perception of computing power network application status; k. Deterministic controller needs to support computing power management and operation: fine-grained data collection capability, hyper visual monitoring capability, automated operation and maintenance capability, and intelligent collaborative optimization; l. Deterministic controller needs to support intelligent scheduling and optimization of computing resources: meet the QoS service guarantee requirements of applications, flexibly optimize networking and computing resource allocation strategies, and achieve deterministic computing resource supply for business; m. Deterministic controller needs to support trusted computing power trading services: trusted real-time dynamic computing power service supply and demand configuration, efficient and reliable data transmission, secure and reliable data protection, fair and equal distribution of benefits; n.TBA.

5.4. Requirements for External Factory Computing

TBA.

5.5. Requirements for Global Management

a. CATS controller support differentiated assurance and control methods for deterministic computing tasks; b. CATS controller support deterministic service management of two-way sensing on the cloud side of the network; c. CATS controller support controlled service migration and service change downtime; d. All deterministic devices need to support deterministic northbound interfaces and deterministic southbound interfaces; e. OPC-UA protocol should be adopted to transmit deterministic requirements, monitoring information, and deterministic configuration among application-oriented deterministic management and control systems, deterministic computing devices, and deterministic network devices.f. CATS controller support the use of OPC-UA protocol to transmit deterministic requirements, monitoring information, and deterministic configurations between application-oriented deterministic control systems, deterministic computing devices, and deterministic network devices.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This document should not affect the security of the Internet.

8. References

8.1. Normative References

[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/info/rfc8174>.

8.2. Informative 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/info/rfc2119>.

[exampleRefMin] Surname, Initials., "Title", 2006.

[exampleRefOrg]
Organization, "Title", 1984, <http://www.example.com/>.

Appendix A. Appendix 1

TBA.

Acknowledgements

TBA.

Contributors

Thanks to all of the contributors.

Authors' Addresses

Fu Tao (editor)
China Academy of Information and Communications Technology
Huayuanbei No.52
beijing
beijing, 100191
China
Email: futao@caict.ac.cn

Zhang Hengsheng (editor)
China Academy of Information and Communications Technology
Huayuanbei No.52
beijing
beijing, 100191
China
Email: zhanghengsheng@caict.ac.cn

Wang Jing (editor)
China Mobile
beijing
beijing, 100191
China
Email: wangjingjc@chinamobile.com

Computing-Aware Traffic Steering
Internet-Draft
Intended status: Informational
Expires: 14 July 2025

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

10 January 2025

CATS Metrics Definition
draft-ietf-cats-metric-definition-00

Abstract

This document defines a set of computing metrics used for Computing-Aware Traffic Steering (CATS).

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 14 July 2025.

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 (C-PS, C-Forwarders, etc.) 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.

Various considerations for metric definition are proposed in [I-D.du-cats-computing-modeling-description], which are useful for defining computing metrics. This document categorizes the relevant compute and network metrics for CATS into three levels based on their complexity and granularity, following the considerations outlined in [I-D.du-cats-computing-modeling-description].

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

* Computing-Aware Traffic Steering (CATS)
* Service
* Service contact instance

3. Definition of Metrics

Introducing a definition of metrics requires balancing the following trade-off: if the metrics are too fine-grained, they become unscalable due to the excessive number of metrics that must be communicated through the metrics distribution protocol. (See [I-D.rcr-opsawg-operational-compute-metrics] for a discussion of metrics distribution protocols.) Conversely, if the metrics are too coarse-grained, they may lack the necessary information to make informed decisions. To ensure scalability while providing sufficient detail for effective decision-making, we propose a definition of metrics that incorporates three levels of abstraction:

* *Level 0 (L0): 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 (L1): Normalized metrics in categories.* These metrics are derived by aggregating L0 metrics into multiple categories, such as network, computing, and storage. Each category is summarized with a single L1 metric by normalizing it into a value within a defined range of scores.

* *Level 2 (L2): Fully normalized metric.* These metrics are derived by aggregating lower level metrics (L0 or L1) into a single L2 metric, which is then normalized into a value within a defined range of scores.

3.1. Level 0: Raw Metrics

Level 0 metrics encompass detailed, raw metrics, including but not limit 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.

* Network: Bandwidth, capacity, throughput, transmit bytes, receive bytes, host bus utilization.

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

* Delay: Time taken to process a request.

L0 metrics can be encoded into an Application Programming Interface (API), such as a RESTful API, and can be solution-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).

Regarding network-related information, the IPPM WG has defined various types of metrics in [performance-metrics]. Additionally, in [RFC9439], the ALTO WG has introduced an extended set of metrics related to packet performance and throughput/bandwidth. For compute metrics, [I-D.rcr-opsawg-operational-compute-metrics] lists a set of cloud resource metrics.

3.2. Level 1: Normalized Metrics in Categories

L1 metrics are organized into distinct categories, such as computing, networking, storage, and delay. Each L0 metric is classified into one of these categories. Within each category, a single L1 metric is computed using an _aggregation function_ and normalized to a unitless score that represents the performance of the underlying resources according to that category. Potential categories include:

* *Computing:* A normalized value derived from computing-related L0 metrics, such as CPU, GPU, and NPU metrics.

* *Networking:* A normalized value derived from network-related L0 metrics.

* *Storage:* A normalized value derived from storage-related L0 metrics.

* *Delay:* A normalized value derived from computing, networking, and storage metrics, reflecting the end-to-end processing delay of a request.

Editor note: detailed categories can be updated according to the CATS WG discussion.

The L0 metrics, such as those defined in [performance-metrics], [RFC9439], and [I-D.rcr-opsawg-operational-compute-metrics], can be categorized into the aforementioned categories. Each category will employ its own aggregation function (e.g., weighted summary) to generate the normalized value. This approach allows the protocol to focus solely on the metric categories and their normalized values, thereby avoiding the need to process solution-specific detailed metrics.

3.3. Level 2: Fully Normalized Metric.

The L2 metric is a single score value derived from the lower level metrics (L0 or L1) using an aggregation function. Different implementations may employ different aggregation functions to characterize the overall performance of the underlying compute and communication resources. The definition of the L2 metric simplifies the complexity of collecting and distributing numerous lower-level metrics by consolidating them into a single, unified score.

TODO: Some implementations may support configuration of Ingress CATS-Forwarders with the metric normalizing method so that it can decode the affection from the L1 or L0 metrics.

Figure 1 shows the logic of metrics in Level 0, Level 1, and Level 2.

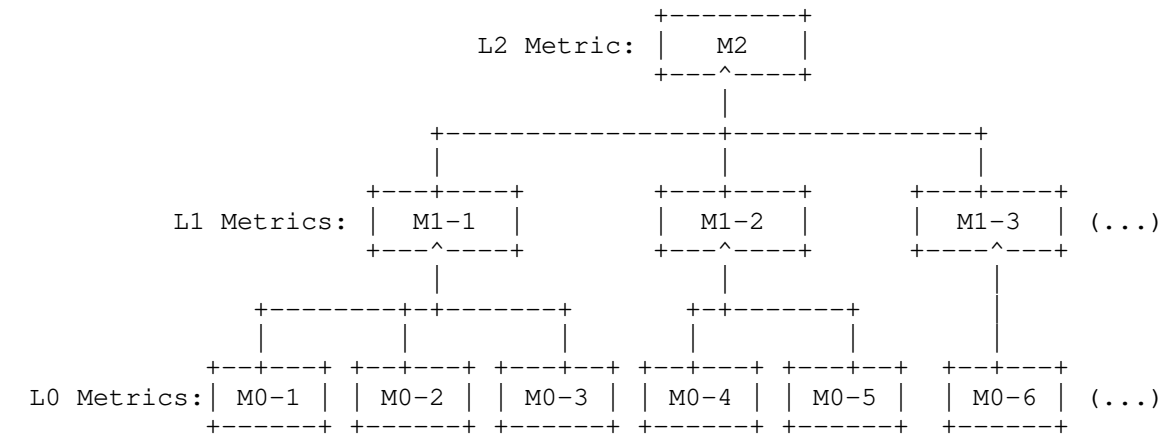


Figure 1: Logic of CATS Metrics in levels

4. Representation of Metrics

This section includes the detailed representation of metrics. [RFC9439] gives a good way to show the representation of some network metrics which is used for network capabilities exposure to applications. This document further describes the representation of CATS metrics.

Basically, in each metric level and for each metric, there will be some common fields for representation, including metric type, unit, and precision. Metric type is a label for network devices to recognize what the metric is. "unit" and "precision" are usually associated with the metric. How many bits a metric occupies in protocols is also required.

Beyond these basic representations, the source of the metrics must also be declared, since there are multiple levels of metrics and their sources are different. As defined in [RFC9439], there are three cost-sources, nominal, sla, and estimation. This document further divide the estimation type into three sub-types, direct measurement, aggregation, and normalization, since different levels of metrics require different sources to acquire CATS metrics. Directly measured metrics have physical meanings and units without any processing. Aggregated metrics can be either physically meaningful or not, and they maintain their meanings compared to the directly measured metrics. Normalized metrics can have physical meanings or not, but they do not have units, and they are just numbers that used for routing decision making.

To be more fine-grained, this document refers to the definition of [RFC9439] on the metrics statistics.

4.1. Level 0 Metric Representation

Raw metrics have exact physical meanings and units. They are directly measured from the underlying computing resources providers. Lots of definition on this level of metrics have been defined in IT industry and other standardizations[DMTF], and this document only show some examples for different categories of metrics for reference.

4.1.1. Compute Raw Metrics

The metric type of compute resources are named as â\200\234compute_type: CPUâ\200\235
or â\200\234compute_type: GPUâ\200\235. Their frequency unit is GHZ, the compute
capabilities unit is FLOPS. Format should support integer and FP8.
It will occupy 4 octets. Example:

Basic fields:
- Metric type: â\200\234compute type_CPUâ\200\235
- Format: integer, FP8
- Bits occupation: 4 octets

Special fields:
- Frequency unit: GHZ
- Compute capabilities unit: FLOPs

Source:
- Direct measurement

Statistics:
- Mean

Figure 2: An Example for Compute Raw Metrics

4.1.2. Storage Raw Metrics

The metric type of storage resources like SSD are named as â\200\234storage_type: SSDâ\200\235. The storage space unit is megaBytes(MBs). Format is integer. It will occupy 2 octets. The unit of read or write speed is denoted as MB per second. Example:

```
Basic fields:
      Metric type: â\200\234storage type_SSDâ\200\235
      Format: integer
      Unit: GB
      Bits occupation: 2 octets
Source:
      nominal
Statistics:
      cur
```

Figure 3: An Example for Storage Raw Metrics

4.1.3. Network Raw Metrics

The metric type of network resources like bandwidth are named as "network_type: Bandwidthâ\200\235. The unit is gigabits per second(Gb/s). Format is integer. It will occupy 2 octets. The unit of TXBytes and RXBytes is denoted as MB per second. Example:

```
Basic fields:
      Metric type: â\200\234network type_Bandwidthâ\200\235
      Format: integer
      Unit: Gb/s
      Bits occupation: 2 octets
Source:
      nominal
Statistics:
      cur
```

Figure 4: An Example for Network Raw Metrics

4.1.4. Delay Raw Metrics

Delay is a kind of synthesized metric which is influenced by computing, storage access, and network transmission. It is named as â\200\234delay_rawâ\200\235. Format should support integer and FP8. Its unit is microsecond. It will occupy 4 octets. Example:

```
   Basic fields:
         Metric type: â\200\234delay_rawâ\200\235
         Format: integer, FP8
         Unit: Microsecond(us)
         Bits occupation: 4 octets
   Source:
         aggregation
   Statistics:
         max
```

Figure 5: An Example for Delay Raw Metrics

4.1.5. Considerations on the Sources of Metrics and the Statistics

The sources of L0 metrics can be nominal, directly measured, or aggregated. Nominal L0 metrics are provided initially by resource providers. Dynamic L0 metrics are measured and updated during service stage. L0 metrics also support aggregation, in case that there are multiple service instances.

The statistics of L0 metrics will follow the definition of Section 3.2 of [RFC9439].

4.2. Level 1 Metric Representation

Normalized metrics in categories have physical meanings but they do not have unit. They are numbers after some ways of abstraction, but they can represent their type, in case that in some use cases, some specific types of metrics require more attention.

4.2.1. Normalized Compute Metrics

The metric type of normalized compute metrics is â\200\234compute_normâ\200\235, and its format is integer. It has no unit. It will occupy an octet. Example:

```
   Basic fields:
         Metric type: â\200\234compute_normâ\200\235
         Format: integer
         Bits occupation: an octet
         Score: 1
   Source:
         normalization
```

Figure 6: An Example for Normalized Compute Metrics

4.2.2. Normalized Storage Metrics

The metric type of normalized compute metrics is â\200\234storage_normâ\200\235, and its format is integer. It has no unit. It will occupy a octet.
Example:

```
Basic fields:
      Metric type: â\200\234storage_normâ\200\235
      Format: integer
      Bits occupation: an octet
      Score: 1
Source:
      normalization
```

Figure 7: An Example for Normalized Storage Metrics

4.2.3. Normalized Network Metrics

The metric type of normalized compute metrics is â\200\234network_normâ\200\235, and its format is integer. It has no unit. It will occupy a octet.
Example:

```
Basic fields:
      Metric type: â\200\234network_normâ\200\235
      Format: integer
      Bits occupation: an octet
      Score: 1
Source:
      normalization
```

Figure 8: An Example for Normalized Network Metrics

4.2.4. Normalized Delay

The metric type of normalized compute metrics is â\200\234delay_normâ\200\235, and its format is integer. It has no unit. It will occupy a octet.
Example:

```
Basic fields:
      Metric type: â\200\234delay_normâ\200\235
      Format: integer
      Bits occupation: an octet
      Score: 1
Source:
      normalization
```

Figure 9: An Example for Normalized Delay Metrics

4.2.5. Considerations on the Sources of Metrics and the Statistics

The sources of L1 metrics is normalized. Based on L0 metrics, service providers design their own algorithms to normalize metrics. For example, assigning different cost values to each raw metric and do summation. L1 metric do not need further statistical values.

4.3. Level 2 Metric Representation

A fully normalized metric is a single value which does not have any physical meaning or unit. Each provider may have its own methods to derive the value, but all providers must follow the definition in this section to represent the fully normalized value.

Metric type is â\200\234norm_fiâ\200\235. The format of the value is non-negative integer. It has no unit. It will occupy a octet. Example:

```
Basic fields:
      Metric type: â\200\234norm_fiâ\200\235
      Format: non-negative integer
      Bits occupation: an octet
      Score: 1
Source:
      normalization
```

Figure 10: An Example for Fully Normalized Metric

The fully normalized value also supports aggregation when there are multiple service instances providing these fully normalized values. When providing fully normalized values, service instances do not need to do further statistics.

5. Comparison of three layers of metric

From L0 to L1 to L2, the computing metric is consolidated. Different level of abstraction can meet the requirements from different services. Table 1 shows the comparison among metric levels.

Level	Encoding Complexity	Extensibility	Stability	Accuracy
Level 0	Complicated	Bad	Bad	Good
Level 1	Medium	Medium	Medium	Medium
Level 2	Simple	Good	Good	Medium

Table 1: Comparison among Metrics Levels

Since Level 0 metrics are raw metrics, therefore, different services may have their own metrics, resulting in hundreds or thousands of metrics in total, this brings huge complexity in protocol encoding and standardization. Therefore, this kind of metrics are always used in customized IT systems case by case. In Level 1 metrics, metrics are categorized into several categories and each category is normalized into a value, therefore they can be encoded into the protocol and standardized. Regarding the Level 2 metrics, all the metrics are normalized into one single metric, it is easier to be encoded in protocol and standardized. Therefore, from the encoding complexity aspect, Level 2 and Level 1 metrics are suggested.

Similarly, when considering extensibility, new services can define their own new L0 metrics, which requires protocol to be extended as needed. Too many metrics type can create a lot of overhead to the protocol resulting in a bad extensibility of the protocol. Level 1 introduce only several metrics categories, which is acceptable for protocol extension. Level 2 metric only need one single metric, so it brings least burden to the protocol. Therefore, from the extensibility aspect, Level 2 and Level 1 metrics are suggested.

Regarding Stability, new Level 0 raw metrics may require new extension in protocol, which brings unstable format for protocol, therefore, this document does not recommend to standardize Level 0 metrics in protocol. Level 1 metrics request only few categories, and Level 2 Metric only introduce one metric to the protocol, so they are preferred from the stability aspect.

In conclusion, for computing-aware traffic steering, it is recommended to use the L2 metric due to its simplicity. If advanced scheduling is needed, L1 metric can be used. L2 metrics are the most comprehensive and dynamic, therefore transferring them to network devices is discouraged due to their high overhead.

Editor notes: this draft can be updated according to the discussion of metric definition in CATS WG.

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. References

8.1. 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-04, 17 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-04>.

[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>.

8.2. Informative References

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

[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.rcr-opsawg-operational-compute-metrics]
Randriamasy, S., Contreras, L. M., Ros-Giralt, J., and R. Schott, "Joint Exposure of Network and Compute Information for Infrastructure-Aware Service Deployment", Work in Progress, Internet-Draft, draft-rcr-opsawg-operational-compute-metrics-08, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-rcr-opsawg-operational-compute-metrics-08>.

[performance-metrics]
"performance-metrics", n.d., <https://www.iana.org/assignments/performance-metrics/performance-metrics.xhtml>.

[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>.

Authors' Addresses

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

Hang Shi
Huawei Technologies
China
Email: shihang9@huawei.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

cats
Internet-Draft
Intended status: Informational
Expires: 4 January 2025

K. Yao
China Mobile
L. M. Contreras
Telefonica
H. Shi
Huawei Technologies
S. Zhang
China Unicom
Q. An
Alibaba Group

3 July 2024

Computing-Aware Traffic Steering (CATS) Problem Statement, Use Cases, and Requirements
draft-ietf-cats-usecases-requirements-04

Abstract

Distributed computing is a tool that service providers can use to achieve better service response time and optimized energy consumption. In such a distributed computing environment, providing services by utilizing computing resources hosted in various computing facilities aids support of services such as computationally intensive and delay sensitive services. Ideally, compute services are balanced across servers and network resources to enable higher throughput and lower response times. To achieve this, the choice of server and network resources should consider metrics that are oriented towards compute capabilities and resources instead of simply dispatching the service requests in a static way or optimizing solely on connectivity metrics. The process of selecting servers or service instance locations, and of directing traffic to them on chosen network resources is called "Computing-Aware Traffic Steering" (CATS).

This document provides the problem statement and the typical scenarios for CATS, which shows the necessity of considering more factors when steering traffic to the appropriate computing resource to best meet the customer’s expectations and deliver the requested service.

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 4 January 2025.

Table of Contents

1. Introduction

Network and computing convergence has been evolving in the Internet for considerable time. With Content Delivery Networks (CDNs) 'frontloading' access to many services, over-the-top service provisioning has become a driving force for many services, such as video, storage and many others. Network operators have extended their capabilities by complementing their network infrastructure by developing CDN capabilities, particularly in edge sites. In addition, more computing resource are deployed at these edge sites as well.

The reason of the fast development of this converged network/compute infrastructure is user demand. On the one hand, users want the best experience, e.g., expressed in low latency and high reliability, for new emerging applications such as high-definition video, Augmented Reality(AR)/Virtual Reality(VR), live broadcast and so on. On the other hand, users want stable experience when moving to different areas.

Generally, edge computing aims to provide better response times and transfer rates compared to cloud computing, by moving the computing towards the edge of a network. There are millions of home gateways, thousands of base stations, and hundreds of central offices in a city that could serve as compute-capable nodes to deliver a service. Note that not all of these nodes would be considered as edge nodes in some views of the network, but they can all provide computing resources to enable a service.

That brings about the key problem of deploying and scheduling traffic to the most suitable computing resource in order to meet the users' service demand.

Service providers often have their own service sites, many of which have been enhanced to support computing services. A service instance deployed at a single site might not provide sufficient capacity to fully guarantee the quality of service required by a customer. Especially at peak hours, computing resources at a single site can not handle all the incoming service requests, leading to longer response times or even dropping of requests experienced by clients. Moreover, increasing the computing resources hosted at each location to the potential maximum capacity is neither feasible nor economically viable in many cases. Offloading computation intensive processing to the user devices is neither acceptable, since it would place huge pressure on local resources such as the battery and incur some data privacy issues if the needed data for computation is not provided locally.

Instead, the same service can be deployed at multiple sites for better availability and scalability. Furthermore, it is desirable to balance the load across all service instances to improve throughput. For this, traffic needs to be steered to the 'best' service instance according to information that may include current computing load, where the notion of 'best' may highly depend on the application demands.

This document describes sample usage scenarios that drive CATS requirements and will help to identify candidate solution architectures and solutions.

2. Definition of Terms

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

Service identifier: An identifier representing a service, which the clients use to access it.

Network Edge: The network edge is an architectural demarcation point used to identify physical locations where the corporate network connects to third-party networks.

Edge Computing: Edge computing is a computing pattern that moves computing infrastructures, i.e, servers, away from centralized data centers and instead places it close to the end users for low latency communication.

Relations with network edge: edge computing infrastructures connect to corporate network through a network edge entry/exit point.

Even though this document is not a protocol specification, it makes use of upper case key words to define requirements unambiguously.

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. Problem Statement

3.1. Multi-deployment of Edge Sites and Service

Since edge computing aims at a closer computing service based on the shorter network path, there will be more than one edge site with the same application in the city/province/state, a number of representative cities have deployed multi-edge sites and the typical applications, and there are more edge sites to be deployed in the future. Before deploying edge sites, there are some factors that need to be considered, such as:

* The existing infrastructure capacities, which could be updated to edge sites, e.g. operators' machine room.

* The amount and frequency of computing resource that is needed.

* The network resource status linked to computing resource.

To improve the effectiveness of service deployment, the problem of how to choose the optimal edge node on which to deploy services needs to be solved. [I-D.contreras-alto-service-edge] introduces considerations for how to deploy applications or functions to the edge, such as the type of instance, optional storage extension, optional hardware acceleration characteristics, and the compute flavor of CPU/GPU, etc. More network and service factors may also be considered, such as:

* Network and computing resource topology: The overall consideration of network access, connectivity, path protection or redundancy, and the location and overall distribution of computing resources in the network, and the relative position within the network topology.

* Location: The number of users, the differentiation of service types, and the number of connections requested by users, etc. For edge nodes located in populous area with a large number of users and service requests, service duplication could be deployed more than in other areas.

* Capacity of multiple edge nodes: Not only the capacity of a single node, but also the total number of requests that can be processed by the resource pool composed of multiple nodes.

* Service category: For example, whether the service is a multi-user interaction, such as video conferencing, or games, or whether it just resource acquisition, such as video viewing. ALTO [RFC7285] can help to obtain one or more of the above pieces of information, so as to provide suggestions or formulate principles and strategies for service deployment.

This information could be collected periodically, and could record the total consumption of computing resources, or the total number of sessions accessed. This would indicate whether additional service instances need to be deployed. Unlike the scheduling of service requests, service deployment should follow the principle of proximity to place new service instances near to customer sites that will request them. If the resources are insufficient to support new instances, the operator can be informed to increase the hardware resources.

In general, the choice of where to locate service instances and when to create new ones in order to provide the right levels of resource to support user demands is important in building a network that supports computing services. However, those aspects are out of scope for CATS and are left for consideration in another document.

3.2. Traffic Steering among Edges Sites and Service Instances

This section describes how existing edge computing systems do not provide all of the support needed for real-time or near-real-time services, and how it is necessary to steer traffic to different sites considering mobility of people, different time slots, events, server loads, network capabilities, and some other factors which might not be directly measured, i.e., properties of edge sites(e.g., geographical location), etc.

In edge computing, the computing resources and network resources are considered when deploying edge sites and services. Traffic is steered to an edge site that is "closest" or to one of a few "close" sites using load-balancing. But the "closest" site is not always the "best" as the status of computing resources and of the network may vary as follows:

* Closest site may not have enough resource, the load may dynamically change.

* Closest site may not have related resource, heterogeneous hardware in different sites.

* The network path to the closest site might not provide the necessary network characteristics, such as low latency or high throughput.

To address these issues some enhancements are needed to steer traffic to sites that can support the requested services.

We assume that clients access one or more services with an objective to meet a desired user experience. Each participating service may be realized at one or more places in the network (called, service instances). Such service instances are instantiated and deployed as part of the overall service deployment process, e.g., using existing orchestration frameworks, within so-called edge sites, which in turn are reachable through a network infrastructure via an edge router.

When a client issues a service request for a required service, the request is steered to one of the available service instances. Each service instance may act as a client towards another service, thereby seeing its own outbound traffic steered to a suitable service instance of the request service and so on, achieving service composition and chaining as a result.

The aforementioned selection of one of candidate service instances is done using traffic steering methods, where the steering decision may take into account pre-planned policies (assignment of certain clients to certain service instances), realize shortest-path to the 'closest' service instance, or utilize more complex and possibly dynamic metric information, such as load of service instances, latency experienced or similar, for a more dynamic selection of a suitable service instance.

It is important to note that clients may move. This means that the service instance that was "best" at one moment might no longer be best when a new service request is issued. This creates a (physical) dynamicity that will need to be catered for in addition to the changes in server and network load.

Figure 1 shows a common way to deploy edge sites in the metro. There is an edge data center for metro area which has high computing resource and provides the service to more User Equipments(UEs) at the working time. Because more office buildings are in the metro area. And there are also some remote edge sites which have limited computing resource and provide the service to the UEs closed to them.

Applications to meet service demands could be deployed in both the edge data center in metro area and the remote edge sites. In this case, the service request and the resource are matched well. Some potential traffic steering may be needed just for special service request or some small scheduling demand.

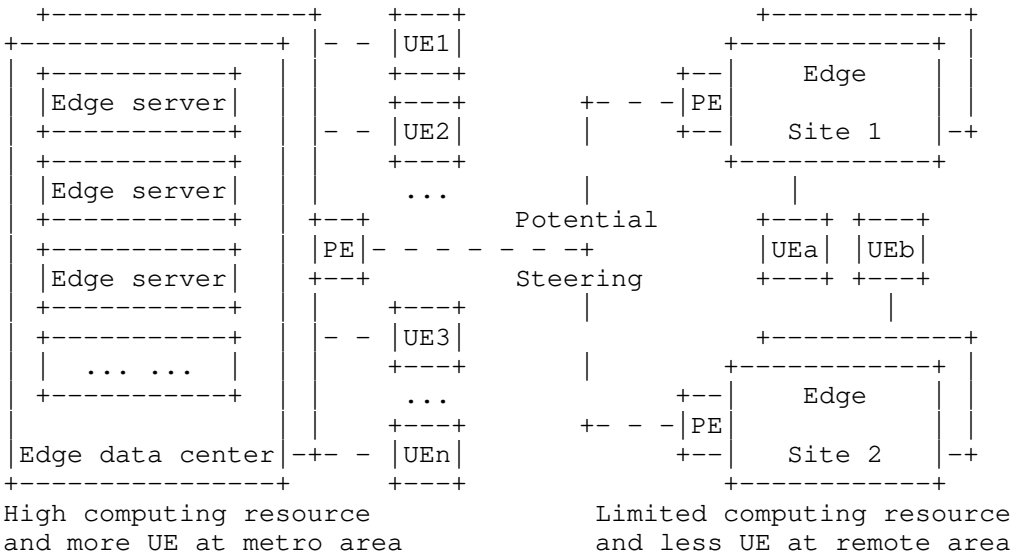


Figure 1: Common Deployment of Edge Sites

Figure 2 shows that during non-working hours, for example at weekend or daily night, more UEs move to the remote area that are close to their house or for some weekend events. So there will be more service request at remote but with limited computing resource, while the rich computing resource might not be used with less UE in the metro area. It is possible for many people to request services at the remote area, but with the limited computing resource, moreover, as the people move from the metro area to the remote area, the edge sites that serve common services will also change, so it may be necessary to steer some traffic back to the metro data center.

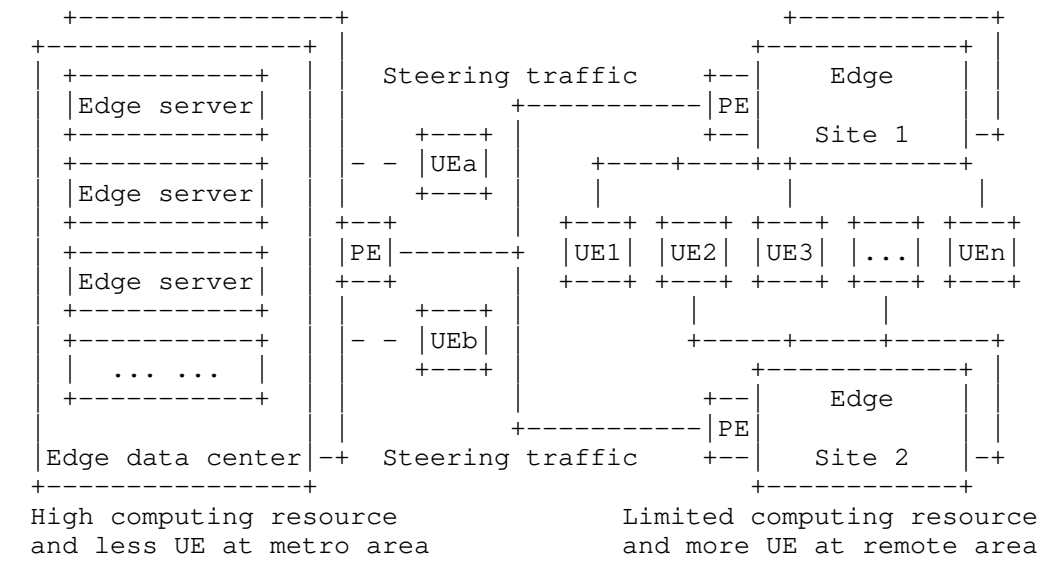


Figure 2: Steering Traffic among Edge Sites

There will also be the common variable of network and computing resources, for someone who is not moving but get a poor latency sometime. Because of other UEs moving, a large number of request for temporary events such as vocal concert, shopping festival and so on, and there will also be the normal change of the network and computing resource status. So for some fixed UEs, it is also expected to steer the traffic to appropriate sites dynamicity.

Those problems indicate that traffic needs to be steered among different edge sites, because of the mobility of the UE and the common variable of network and computing resources. Moreover, some use cases in the following section require both low latency and high computing resource usage or specific computing hardware capabilities (such as local GPU); hence joint optimization of network and computing resource is needed to guarantee the Quality of Experience (QoE).

4. Use Cases

This section presents a non-exhaustive set of use cases which would benefit from the dynamic selection of service instances and the steering of traffic to those service instances.

4.1. Computing-Aware AR or VR

Cloud VR/AR services are used in some exhibitions, scenic spots, and celebration ceremonies. In the future, they might be used in more applications, such as industrial internet, medical industry, and meta verse.

Cloud VR/AR introduces the concept of cloud computing to the rendering of audiovisual assets in such applications. Here, the edge cloud helps encode/decode and render content. The end device usually only uploads posture or control information to the edge and then VR/ AR contents are rendered in the edge cloud. The video and audio outputs generated from the edge cloud are encoded, compressed, and transmitted back to the end device or further transmitted to central data center via high bandwidth networks.

Edge sites may use CPU or GPU for encode/decode. GPU usually has better performance but CPU is simpler and more straightforward to use as well as possibly more widespread in deployment. Available remaining resources determines if a service instance can be started. The instance's CPU, GPU and memory utilization has a high impact on the processing delay on encoding, decoding and rendering. At the same time, the network path quality to the edge site is a key for user experience of quality of audio/ video and input command response times.

A Cloud VR service, such as a mobile gaming service, brings challenging requirements to both network and computing so that the edge node to serve a service request has to be carefully selected to make sure it has sufficient computing resource and good network path. For example, for an entry-level cloud VR (panoramic 8K 2D video) with 110-degree Field of View (FOV) transmission, the typical network requirements are bandwidth 40Mbps, 20ms for motion-to-photon latency, packet loss rate is 2.4E-5; the typical computing requirements are 8K H.265 real-time decoding, 2K H.264 real-time encoding. We can further divide the 20ms latency budget into:

(i) sensor sampling delay(client), which is considered imperceptible by users is less than 1.5ms including an extra 0.5ms for digitalization and end device processing.

(ii) display refresh delay(client), which take 7.9ms based on the 144Hz display refreshing rate and 1ms extra delay to light up.

(iii) image/frame rendering delay(server), which could be reduced to 5.5ms.

(iv) round trip network delay(network), which should be bounded to 20-1.5-5.5-7.9 = 5.1ms.

So the budgets for server(computing) delay and network delay are almost equivalent, which make sense to consider both of the delay for computing and network. And it can't meet the total delay requirements or find the best choice by either optimize the network or computing resource.

Based on the analysis, here are some further assumption as Figure 3 shows, the client could request any service instance among 3 edge sites. The delay of client could be same, and the differences of different edge sites and corresponding network path has different delays:

* Edge site 1: The computing delay=4ms based on a light load, and the corresponding network delay=9ms based on a heavy traffic.

* Edge site 2: The computing delay=10ms based on a heavy load, and the corresponding network delay=4ms based on a light traffic.

* Edge site 3: The edge site 3's computing delay=5ms based on a normal load, and the corresponding network delay=5ms based on a normal traffic.

In this case, we can't get a optimal network and computing total delay if choose the resource only based on either of computing or network status:

* If choosing the edge site based on the best computing delay it will be the edge site 1, the E2E delay=22.4ms.

* If choosing the edge site based on the best network delay it will be the edge site 2, the E2E delay=23.4ms.

* If choosing the edge site based on both of the status it will be the edge site 3, the E2E delay=19.4ms.

So, the best choice to ensure the E2E delay is edge site 3, which is 19.4ms and is less than 20ms. The differences of the E2E delay is only 3~4ms among the three, but some of them will meet the application demand while some doesn't.

The conclusion is that it requires to dynamically steer traffic to the appropriate edge to meet the E2E delay requirements considering both network and computing resource status. Moreover, the computing resources have a big difference in different edges, and the "closest site" may be good for latency but lacks GPU support and should therefore not be chosen.

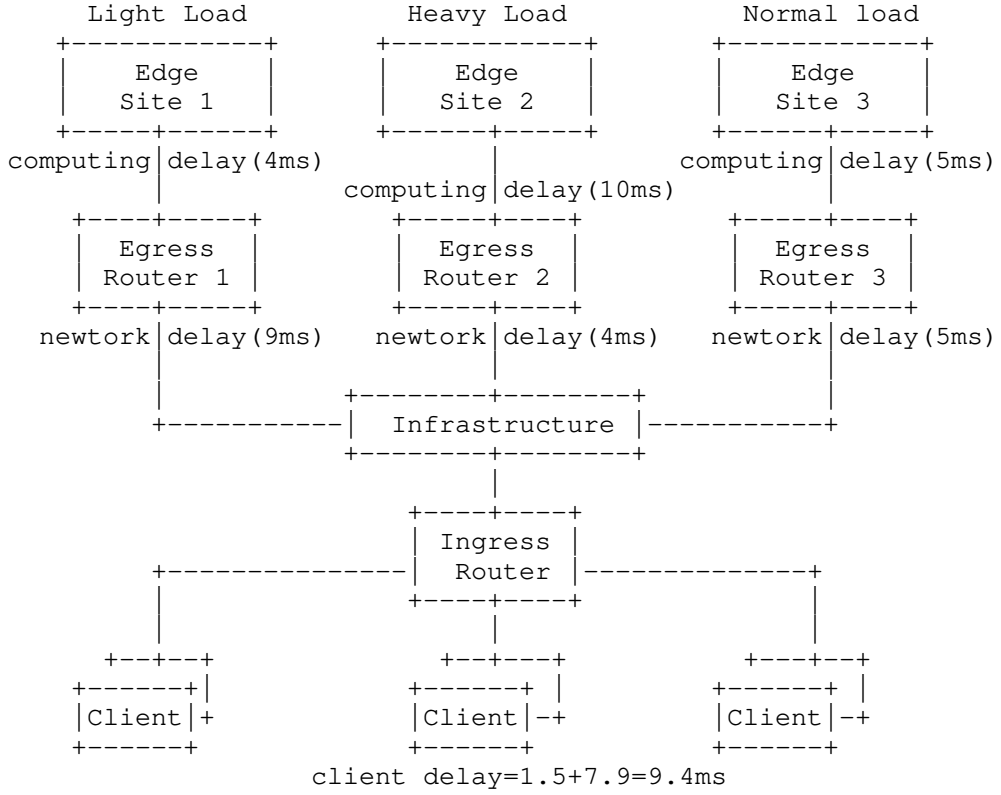


Figure 3: Computing-Aware AR or VR

Furthermore, specific techniques may be employed to divide the overall rendering into base assets that are common across a number of clients participating in the service, while the client-specific input data is being utilized to render additional assets. When being delivered to the client, those two assets are being combined into the overall content being consumed by the client. The requirements for sending the client input data as well as the requests for the base assets may be different in terms of which service instances may serve the request, where base assets may be served from any nearby service instance (since those base assets may be served without requiring cross-request state being maintained), while the client-specific

input data is being processed by a stateful service instance that changes, if at all, only slowly over time due to the stickiness of the service that is being created by the client-specific data. Other splits of rendering and input tasks can be found in[TR22.874] for further reading.

When it comes to the service instances themselves, those may be instantiated on-demand, e.g., driven by network or client demand metrics, while resources may also be released, e.g., after an idle timeout, to free up resources for other services. Depending on the utilized node technologies, the lifetime of such "function as a service" may range from many minutes down to millisecond scale. Therefore, computing resources across participating edges exhibit a distributed (in terms of locations) as well as dynamic (in terms of resource availability) nature. In order to achieve a satisfying service quality to end users, a service request will need to be sent to and served by an edge with sufficient computing resource and a good network path.

4.2. Computing-Aware Intelligent Transportation

For the convenience of transportation, more video capture devices are required to be deployed as urban infrastructure, and the better video quality is also required to facilitate the content analysis. Therefore, the transmission capacity of the network will need to be further increased, and the collected video data need to be further processed, such as for pedestrian face recognition, vehicle moving track recognition, and prediction. This, in turn, also impacts the requirements for the video processing capacity of computing nodes.

In auxiliary driving scenarios, to help overcome the non-line-of-sight problem due to blind spot or obstacles, the edge node can collect comprehensive road and traffic information around the vehicle location and perform data processing, and then vehicles with high security risk can be warned accordingly, improving driving safety in complicated road conditions, like at intersections. This scenario is also called "Electronic Horizon", as explained in [HORITA]. For instance, video image information captured by, e.g., an in-car, camera is transmitted to the nearest edge node for processing. The notion of sending the request to the "nearest" edge node is important for being able to collate the video information of "nearby" cars, using, for instance, relative location information. Furthermore, data privacy may lead to the requirement to process the data as close to the source as possible to limit data spread across too many network components in the network.

Nevertheless, load at specific "closest" nodes may greatly vary, leading to the possibility for the closest edge node becoming overloaded, leading to a higher response time and therefore a delay in responding to the auxiliary driving request with the possibility of traffic delays or even traffic accidents occurring as a result. Hence, in such cases, delay-insensitive services such as in-vehicle entertainment should be dispatched to other light loaded nodes instead of local edge nodes, so that the delay-sensitive service is preferentially processed locally to ensure the service availability and user experience.

In video recognition scenarios, when the number of waiting people and vehicles increases, more computing resources are needed to process the video content. For rush hour traffic congestion and weekend personnel flow from the edge of a city to the city center, efficient network and computing capacity scheduling is also required. Those would cause the overload of the nearest edge sites if there is no extra method used, and some of the service request flow might be steered to others edge site except the nearest one.

4.3. Computing-Aware Digital Twin

A number of industry associations, such as the Industrial Digital Twin Association or the Digital Twin Consortium (https://www.digitaltwinconsortium.org/), have been founded to promote the concept of the Digital Twin (DT) for a number of use case areas, such as smart cities, transportation, industrial control, among others. The core concept of the DT is the "administrative shell" [Industry4.0], which serves as a digital representation of the information and technical functionality pertaining to the "assets" (such as an industrial machinery, a transportation vehicle, an object in a smart city or others) that is intended to be managed, controlled, and actuated.

As an example for industrial control, the programmable logic controller (PLC) may be virtualized and the functionality aggregated across a number of physical assets into a single administrative shell for the purpose of managing those assets. PLCs may be virtualized in order to move the PLC capabilities from the physical assets to the edge cloud. Several PLC instances may exist to enable load balancing and fail-over capabilities, while also enabling physical mobility of the asset and the connection to a suitable "nearby" PLC instance. With this, traffic dynamicity may be similar to that observed in the connected car scenario in the previous subsection. Crucial here is high availability and bounded latency since a failure of the (overall) PLC functionality may lead to a production line stop, while boundary violations of the latency may lead to loosing synchronization with other processes and, ultimately, to production faults, tool failures or similar.

Particular attention in Digital Twin scenarios is given to the problem of data storage. Here, decentralization, not only driven by the scenario (such as outlined in the connected car scenario for cases of localized reasoning over data originating from driving vehicles) but also through proposed platform solutions, such as those in [GAIA-X], plays an important role. With decentralization, endpoint relations between client and (storage) service instances may frequently change as a result.

4.4. Computing-Aware SD-WAN

SD-WAN provides organizations or enterprises with centralized control over multiple sites which are network endpoints including branch offices, headquarters, data centers, clouds, and more. A enterprise may deploy their services and applications in different locations to achieve optimal performance. The traffic sent by a host will take the shortest WAN path to the closest server. However, the closet server may not be the best choice with lowest cost of network and computing resources for the host. If the path computation element can consider the computing dimension information in path computation, the best path with lowest cost can be provided.

The computing related information can be the number of vCPUs of the VM running the application/services, CPU utilization rate, usage of memory, etc.

The SD-WAN can be aware of the computing resource of applications deployed in the multiple sites and can perform the routing policy according to the information is defined as the computing-aware SD-WAN.

Many enterprises are performing the cloud migration to migrate the applications from data centers to the clouds, including public, private, and hybrid clouds. The clouds resources can be from the same provider or multiple cloud providers which have some benefits including disaster recovery, load balancing, avoiding vendor lock-in.

In such cloudification deployments SD-WAN provides enterprises with centralized control over Customer-Premises Equipments(CPEs) in branch offices and the cloudified CPEs(vCPEs) in the clouds. The CPEs connect the clients in branch offices and the application servers in clouds. The same application server in different clouds is called an application instance. Different application instances have different computing resource.

SD-WAN is aware of the computing resource of applications deployed in the clouds by vCPEs, and selects the application instance for the client to visit according to computing power and the network state of WAN.

Figure 4 below illustrates Computing-aware SD-WAN for Enterprise Cloudification.

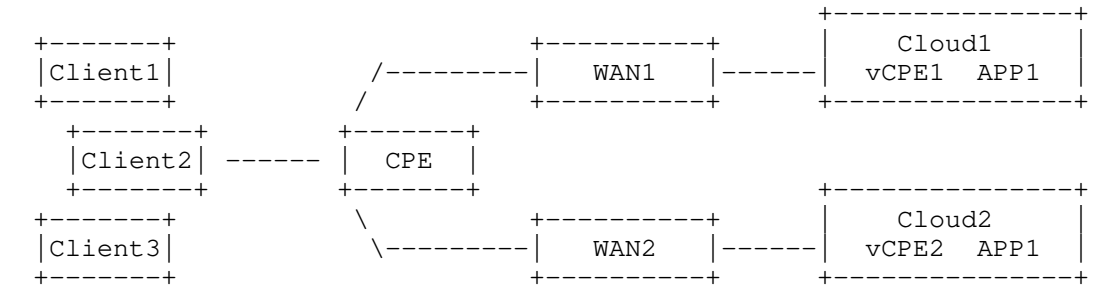


Figure 4: Illustration of Computing-aware SD-WAN for Enterprise Cloudification

The current computing load status of the application APP1 in cloud1 and cloud2 is as follows: each application uses 6 vCPUs. The load of application in cloud1 is 50%. The load of application in cloud2 is 20%. The computing resource of APP1 are collected by vCPE1 and vCPE2 respectively. Client1 and Client2 are visiting APP1 in cloud1. WAN1 and WAN2 have the same network states. Considering lightly loaded application SD-WAN selects APP1 in cloud2 for the client3 in branch office. The traffic of client3 follows the path: Client3 -> CPE -> WAN1 -> Cloud2 vCPE1 -> Cloud2 APP1

4.5. Computing-Aware AI Large Model Inference

AI(Artificial Intelligence) large model refers to models that are characterized by their large size, high complexity, and high computational requirements. AI large models have become increasingly important in various fields, such as natural language processing for text classification, computer vision for image classification and object detection, and speech recognition.

AI large model contains two key phases: training and inference. Training refers to the process of developing an AI model by feeding it with large amounts of data and optimizing it to learn and improve its performance. Training has high demand on computing and memory resource, so that training is usually deployed in large central data centers. On the other hand, inference is the process of using the trained AI model to make predictions or decisions based on new input data. Compared to training, the AI Inference does not consume large amount of computing resources,and it is usually deployed at edge sites and end devices for real time and dynamic service response.

Figure 5 shows the cloud-edge-device co-inference deployment. Single site deployment of the model can not provide enough compute resources and is not sufficient for fulfilling some AI Inference work. The cloud-edge-device co-inference can not only guarantee the compute resources but also achieve low latency as part of AI inference tasks is deployed near clients or even within client devices. When handling AI inference tasks, if traffic load between clients and edge sites is high or edge resource are overloaded, the response of inference may not be accepted by services. And CATS is needed to ensure the QoS of AI inference

There are different types of deployments for cloud-edge-device co-inference. Depending on applications and compute resources. Models can be deployed in edge sites only or in both cloud and edge sites. More specifically, some pruned models can be deployed in end devices for compute offloading and low latency response. In all of the cases above, the problem of steering the traffic to different edge sites fits in the scope of CATS.

The same trained model will be deployed in each edge sites so as to provide undifferentiated inference service. Service selection across different edge sites is for low latency service response just like use cases mentioned in other sections above. Moreover, Specific compute resources like GPUs which are most suitable for AI inference are provided at each edge sites, and relevant metrics should be collected and distributed to the network for better traffic steering decision making. Generalized compute resources like CPUs can also finish AI inference, but they are less efficient and power saving than GPUs.

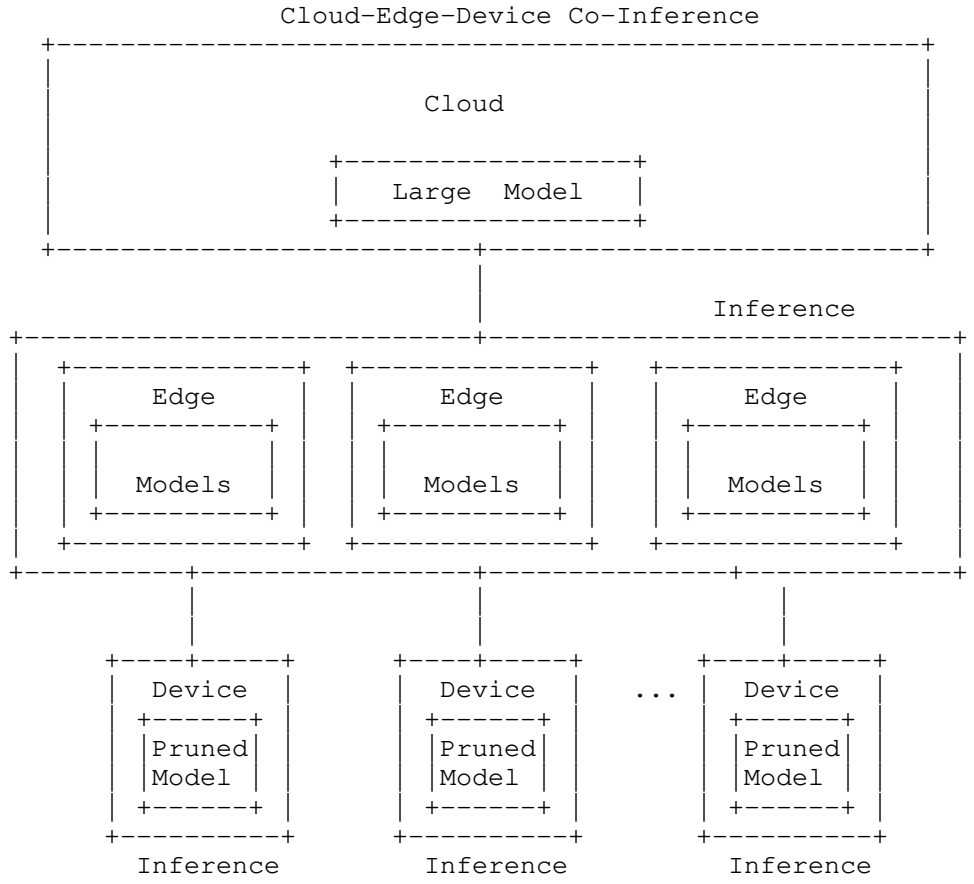


Figure 5: Illustration of Computing-aware AI large model inference

5. Requirements

In the following, we outline the requirements for the CATS system to overcome the observed problems in the realization of the use cases above.

5.1. Support dynamic and effective selection among multiple service instances

The basic requirement of CATS is to support the dynamic access to different service instances residing in multiple computing sites and then being aware of their status, which is also the fundamental model to enable the traffic steering and to further optimize the network and computing services. A unique service identifier is used by all the service instances for a specific service no matter which edge site an instance may attach to. The mapping of this service identifier to a network locator makes sure the data packet CATS potentially reach any of the service instances deployed in various edge sites.

Moreover, according to CATS use cases, some applications require E2E low latency, which warrants a quick mapping of the service identifier to the network locator. This leads to naturally the in-band methods, involving the consideration of using metrics that are oriented towards compute capabilities and resources, and their correlation with services. Therefore, a desirable system

R1: MUST provide a discovery and resolving methodology for the mapping of a service identifier to a specific address.

R2: MUST provide an mapping methods for further quickly selecting the service instance.

R3: SHOULD provide a timeout limitation for selecting the service instance.

R4: MUST provide a method to determine the availability of a service instance.

R5: MUST provide a mechanism for solving the service contention problem when multiple service instances with the same service identifier are all available to provide computing services.

5.2. Support Agreement on Metric Representation

Computing metrics can have many different semantics, particularly for being service-specific. Even the notion of a "computing load" metric could be represented in many different ways. Such representation may entail information on the semantics of the metric or it may be purely one or more semantic-free numerals. Agreement of the chosen representation among all service and network elements participating in the service instance selection decision is important. Therefore, a desirable system

R6: MUST agree on using metrics that are oriented towards compute capabilities and resources and their representation among service elements in the participating edges.

R7: MUST include network metrics.

5.3. Support Moderate Metric Distributing

Network path costs in the current routing system usually do not change very frequently. Network traffic engineering metrics (such as available bandwidth) may change more frequently as traffic demands fluctuate, but distribution of these changes is normally damped so that only significant changes cause routing protocol messages.

However, metrics that are oriented towards compute capabilities and resources in general can be highly dynamic, e.g., changing rapidly with the number of sessions, the CPU/GPU utilization and the memory consumption, etc. It has to be determined at what interval or based on what events such information needs to be distributed. Overly frequent distribution with more accurate synchronization may result in unnecessary overhead in terms of signaling.

Moreover, depending on the service related decision logic, one or more metrics need to be conveyed in a CATS domain. The problem to be addressed here may be the frequency of such conveyance, thanks to the comprehensive load that a signaling process may add to the overall network traffic. While existing routing protocols may serve as a baseline for signaling metrics, other means to convey the metrics can equally be considered and even be realized. Specifically, a desirable system

R8: MUST provide mechanisms for metric collection.

Collecting metrics from all of the services instances may incur much overhead for the decision maker, and thus hierarchical metric collection is needed. That is,

R9: SHOULD provide mechanisms to aggregate the metrics.

CATS components do not need to be aware of how metrics are collected behind the aggregator.

R10: MUST provide mechanisms to distribute the metrics.

R11: MUST realize means for rate control for distributing of metrics.

5.4. Support Alternative Definition and Use of Metrics

Considering computing resources assigned to a service instance on a server, which might be related to some critical metrics like the processing delay, is crucial in addition to the network delay in some cases. Therefore, the CATS components might use both the network and computing metrics for service instance selection. For this reason:

R12: a computing semantic model SHOULD be defined for the mapping selection.

We recognize that different network nodes, e.g., routers, switches, etc., may have diversified capabilities even in the same routing domain, let alone in different administrative domains. So, metrics that are oriented towards compute capabilities and resources that have been adopted by some nodes may not be supported by others, either due to technical reasons, administrative reasons, or something else. There exist scenarios in which a node supporting service-specific metrics might prefer some type of metrics to others[TR22.874]. Of course, specific metrics might not be utilized at all in other scenarios. Hence:

R13: In addition to common metrics that are agreed by all CATS components like processing delay, there SHOULD be some other ways for metrics definition, which is used for the selection of specific service instance.

Therefore, a desirable system

R14: MUST set up metric information that can be understood by CATS components.

For metrics that CATS components do not understand or support, CATS components will ignore them.

5.5. Support Instance Affinity

In the CATS system, a service may be provided by one or more service instances that would be deployed at different locations in the network. Each instance provides equivalent service functionality to their respective clients. The decision logic of the instance selection are subject to the normal packet level communication and packets are forwarded based on the operating status of both network and computing resources. This resource status will likely change over time, leading to individual packets potentially being sent to different network locations, possibly segmenting individual service transactions and breaking service-level semantics. Moreover, when a client moves, the access point might change and successively lead to the same result of the change of service instance. If execution changes from one (e.g., virtualized) service instance to another, state/context needs transfer to another. Such required transfer of state/context makes it desirable to have instance affinity as the default, removing the need for explicit context transfer, while also supporting an explicit state/context transfer (e.g., when metrics change significantly). So in those situations:

R15: Instance affinity MUST be maintained when state information is needed.

The nature of this affinity is highly dependent on the nature of the specific service, which could be seen as a 'instance affinity' to represent the relationship. The minimal affinity of a single request represents a stateless service, where each service request may be responded to without any state being held at the service instance for fulfilling the request.

Providing any necessary information/state in-band as part of the service request, e.g., in the form of a multi-form body in an HTTP request or through the URL provided as part of the request, is one way to achieve such stateless nature.

Alternatively, the affinity to a particular service instance may span more than one request, as in the AR/VR use case, where previous client input is needed to render subsequent frames.

However, a client, e.g., a mobile UE, may have many applications running. If all, or majority, of the applications request the CATS-based services, then the runtime states that need to be created and accordingly maintained would require high granularity. In the extreme scenario, this granular requirement could reach the level of per-UE per-APP per-(sub)flow with regard to a service instance. Evidently, these fine-granular runtime states can potentially place a heavy burden on network devices if they have to dynamically create and maintain them. On the other hand, it is not appropriate either to place the state-keeping task on clients themselves.

Besides, there might be the case that UE moves to a new (access) network or the service instance is migrated to another cloud, which cause the unreachable or inconvenient of the original service instance. So the UE and service instance mobility also need to be considered.

Therefore, a desirable system

R16: MUST maintain instance affinity which MAY span one or more service requests, i.e., all the packets from the same application-level flow MUST go to the same service instance unless the original service instance is unreachable

R17: MUST avoid keeping fine runtime-state granularity in network nodes for providing instance affinity.

R18: MUST provide mechanisms to minimize client side states in order to achieve the instance affinity.

R19: SHOULD support the UE and service instance mobility.

5.6. Preserve Communication Confidentiality

Exposing the information of computing resources to the network may lead to the leakage of computing domain and application privacy. In order to prevent it, it need to consider the methods to process the sensitive information related to computing domain. For instance, using general anonymous methods, including hiding the key information representing the identification of devices, or using an index to represent the service level of computing resources, or using customized information exposure strategies according to specific application requirements or network scheduling requirements. At the same time, when anonymity is achieved, it is also necessary to consider whether the computing information exposed in the network can help make full use of traffic steering. Therefore, a CATS system

R20: MUST preserve the confidentiality of the communication relation between user and service provider by minimizing the exposure of user-relevant information according to user needs.

6. Security Considerations

CATS decision making process is deeply related to computing and network status as well as some service information. Some security issues need to be considered when designing CATS system.

Service data sometimes needs to be moved among different edge sites to maintain service consistency and availability. Therefore:

R21: service data MUST be protected from interception.

The act of making compute requests may reveal the nature of user's activities, so that:

R22: the nature of user's activities SHOULD be hidden as much as possible.

The behavior of the network can be adversely affected by modifying or interfering with advertisements of computing resource availability. Such attacks could deprive users' of the services they desires, and might be used to divert traffic to interception points. Therefore,

R23: secure advertisements are REQUIRED to prevent rogue nodes from participating in the network.

7. IANA Considerations

This document makes no requests for IANA action.

8. References

8.1. 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>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <https://www.rfc-editor.org/rfc/rfc7285>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/rfc/rfc7665>.

[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>.

8.2. Informative References

[GAIA-X] Gaia-X, "GAIA-X: A Federated Data Infrastructure for Europe", 2021.

[HORITA] Horita, Y., "Extended electronic horizon for automated driving", Proceedings of 14th International Conference on ITS Telecommunications (ITST), 2015.

[I-D.contreras-alto-service-edge] Contreras, L. M., Randriamasy, S., Ros-Giralt, J., Perez, D. A. L., and C. E. Rothenberg, "Use of ALTO for Determining Service Edge", Work in Progress, Internet-Draft, draft-contreras-alto-service-edge-10, 13 October 2023, <https://datatracker.ietf.org/doc/html/draft-contreras-alto-service-edge-10>.

[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-04, 17 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-04>.

[Industry4.0] Industry4.0, "Details of the Asset Administration Shell, Part 1 & Part 2", 2020.

[TR22.874] 3GPP, "Study on traffic characteristics and performance requirements for AI/ML model transfer in 5GS (Release 18)", 2021.

Appendix A. Appendix A

This section presents several attempts to apply CATS solutions for improving service performance in different use cases. It is a temporary and procedural section which might be deleted or merged in future updates. The major reason is to help facilitate the discussion and definition of CATS metrics and solidify CATS framework and CATS solutions.

A.1. CATS for Content Delivery Network(CDN)

CDN is mature enough to deploy contents like high resolution videos near clients, so as to provide good performance. However, when existing CDN servers can not guarantee the quality of service, for example, there is not enough query per second(QPS) offering capabilities, CATS can help relieve the problem and improve the overall performance through better load balancing. Two deployment methods of CATS are tested in two different provinces of China, i.e. distributed solution and centralized solution. Some preliminary results show that CATS can guarantee the service performance at client side when there are large number of concurrent sessions coming, compared to existing CDN scheduling solutions. Key performance indicators include the end-to-end scheduling delay, average computing capabities after load balancing(measured as queries per second).

Figure 6 below illustrates the deployment of CATS solution in ten cities of Henan, China. Two ingress nodes are deployed in Zhengzhou, which is the provincial capital of Henan. All egress nodes are deployed in the other nine cities, with each egress node settled in only one city respectively. Client terminals are deployed near ingress nodes in Zhengzhou. The provincial networking can test the performance gains of CATS solutions over 100 kilometers.

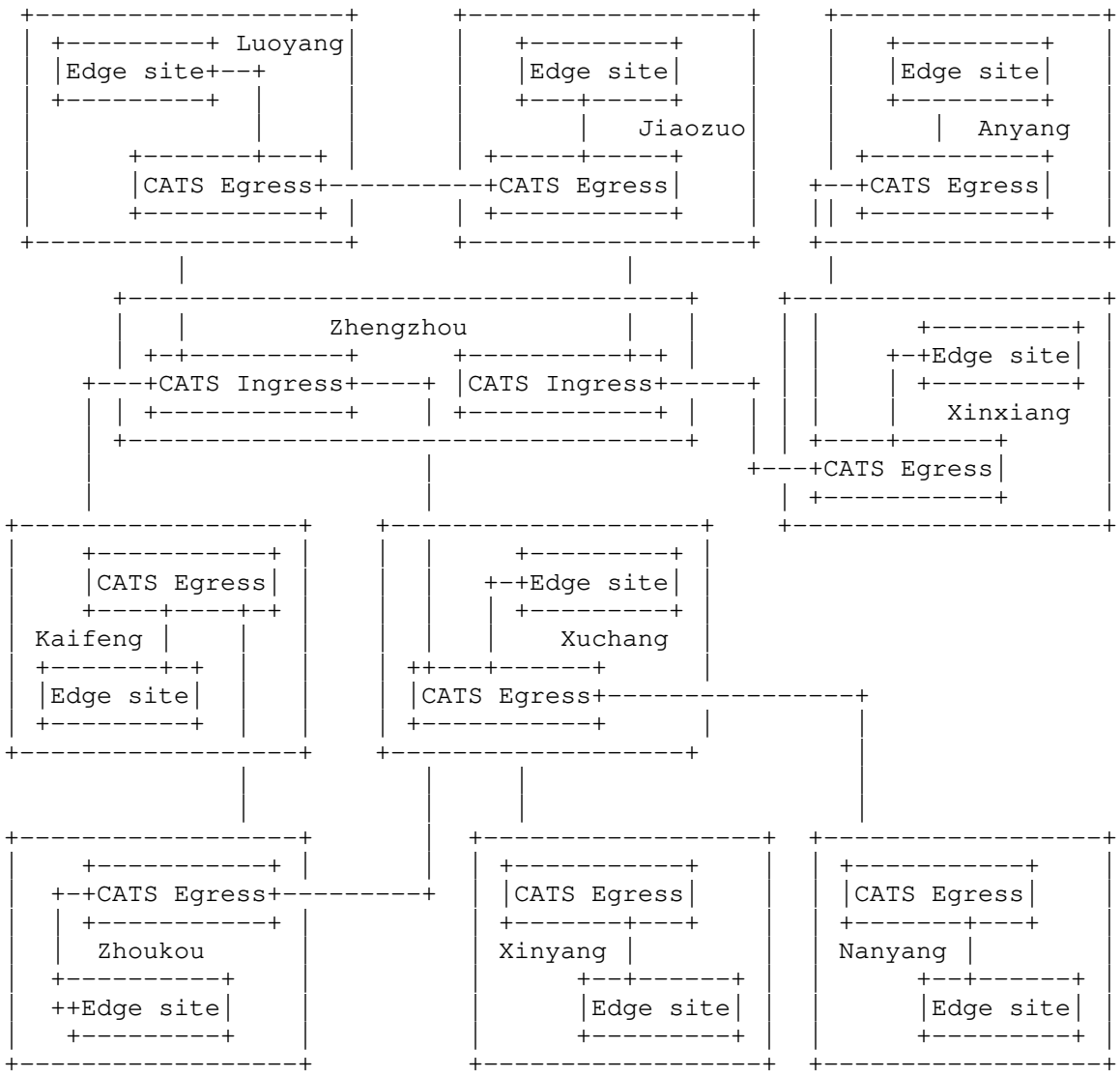


Figure 6: Deployment of CATS in ten cities of Henan, China

Figure 7 below illustrates the deployment of CATS solution in three cities of Jiangsu, China. The ingress node is deployed in Wuxi, while the other two egress nodes are deployed in Xuzhou and Suzhou, respectively. Client devices like laptops and the centralized controller are deployed near the ingress node in Wuxi, since Wuxi owns the largest computing capabilities inside the city and can support over 200,000 connections per second at its peak value. CATS can help alleviate the stress of load balancing at peak hours.

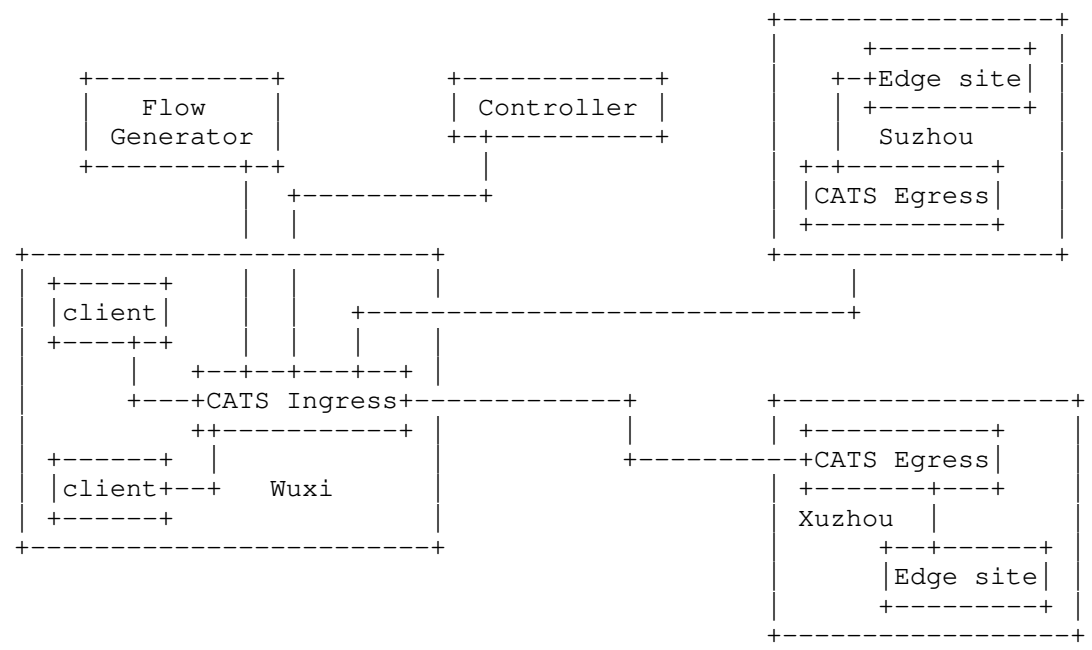


Figure 7: Deployment of CATS in three cities of Jiangsu, China

From the experiments in CDN, benefits of CATS can be summarized as follows. CATS can adapt to network quality degradation more timely than traditional approaches. The frequency of DNS request for available service instances is set to be 600 seconds normally, which is a bit too long when the network quality can not be guaranteed. In a CATS system, the metric update and distribution frequency is set to be 15 seconds in this case, which is the same as the normal refresh frequency of BGP update. Therefore, after the first DNS request for a service instance, CATS will alternatively select other instances no later than 15 seconds if the current service instance do not work well or the quality of path towards this instance drops.

A.2. CATS for MIGU Cloud Rendering

MIGU is the digital content subsidiary of China Mobile, offering various digital and interactive applications which are enriched by 5G, cloud rendering, and AI. Cloud rendering needs a lot of compute resources to be deployed at edge sites, in order to satisfy the real-time modeling requirements. In cooperation with China Mobile Zhejiang Co. Ltd, MIGU has deployed its cloud rendering applications at various edge sites in Zhejiang, in order to test how CATS solution can improve the overall performance of image rendering. Key performance indicators include the resolution and sharpness of images or videos, and processing delay.

Figure 8 below illustrate the deployment of MIGU cloud rendering applications in Zhejiang. Three edge sites are deployed in Wenzhou, Ningbo, and Jiaxing, respectively. In each site, there are some servers for processing rendering services and some corresponding management nodes. One CATS egress node is deployed outside each site for service instance selection. There is a MIGU cloud platform which collects the compute metrics from three different sites, and then it passes these information to the central controller and the controller will synchronize these information to the ingress node which is deployed in Hanzhou, near the client.

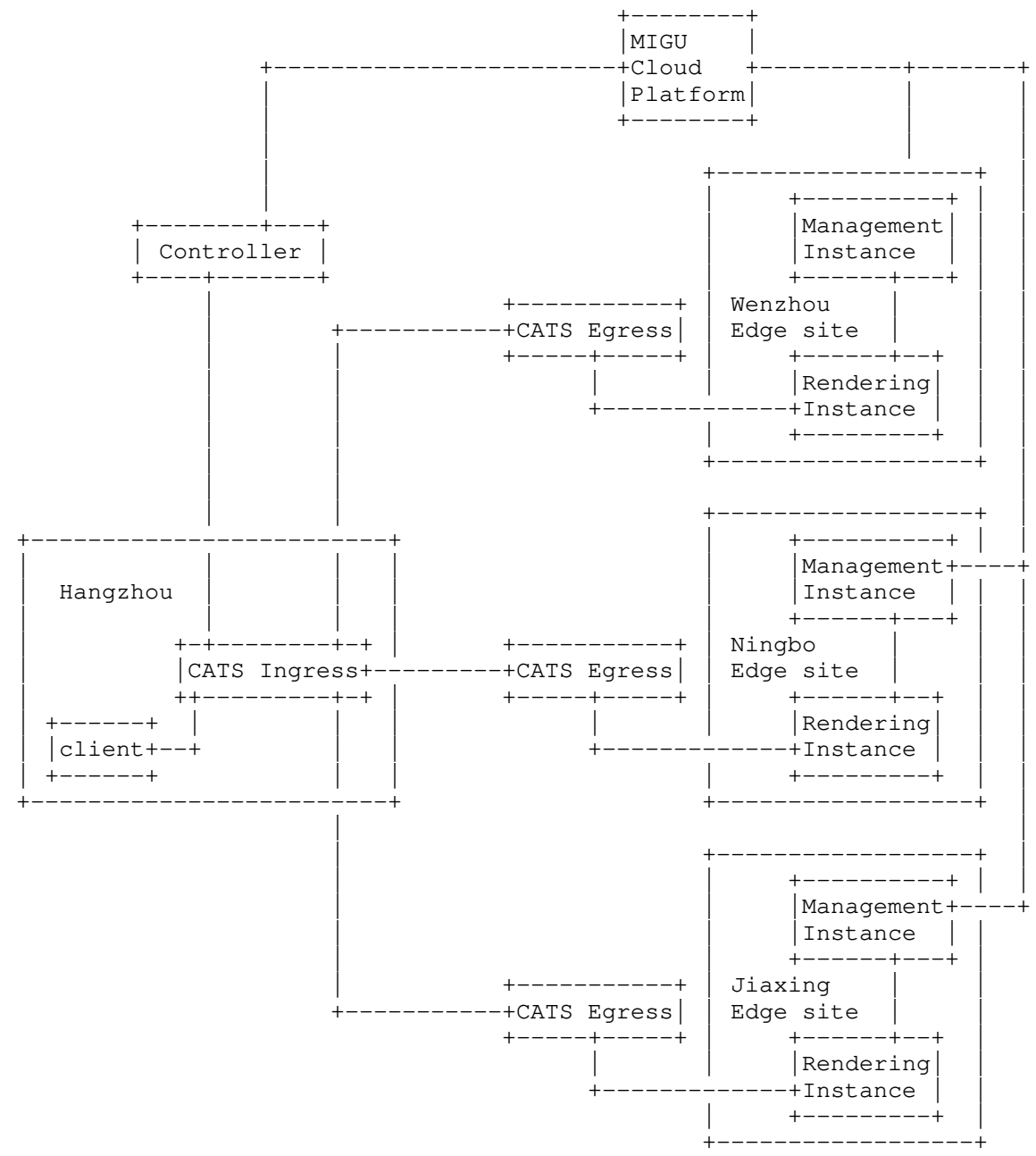


Figure 8: Deployment of CATS for MIGU App Performance Improvement in Zhejiang, China

A.3. CATS for High-speed Internet of Vehicles

In high-speed Internet of vehicles (IoV), high-speed vehicles, such as cars, trains, subways, etc., need to communicate with other vehicles, infrastructure or cloud service through network to run applications carried by vehicles. These applications can be divided into two types. One type of applications will affect the driving, such as autonomous driving, remote control, and intelligent driving services. They have extremely high requirements on network delay, and the console needs to make quick judgments and responses. Otherwise, as the vehicle travels quickly, a brief misoperation may lead to extremely serious traffic accidents. And they have extremely high requirements for service switching efficiency. The coverage of a base station is limited, and the capabilities of different service sites are also different. Vehicle movement will inevitably cause switching of access station and service site. The delay and service changes caused by switching also directly affect the experience and safety of driving. Another type of applications is not related to driving, such as voice communications, streaming media, and other entertainment services. They do not have strict requirements on real-time performance and service switching efficiency, but may requires higher computing capability. Due to the complex requirements of high-speed IoV on computing capability, an efficient way is needed to jointly schedule computing and network resources..

The hybrid CATS scheme combines both the characteristics of centralized and distributed schemes. In hybrid CATS scheme, the awareness and advertisement of computing status are performed by a centralized orchestration system, service selection and path computation are performed by network devices. As shown in Figure 9, the centralized computing status advertisement can reduce the deployment hurdle of CATS.

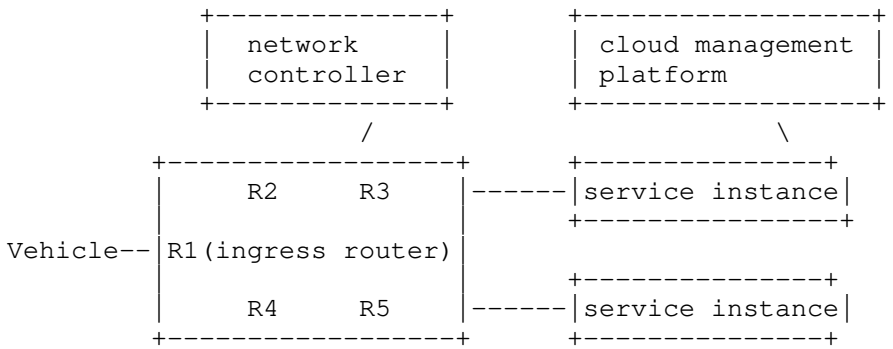


Figure 9: Deployment of CATS for Intelligent Transportation in Hebei, China

The hybrid CATS scheme based high-speed IoV solution has been deployed and validated for the first time in Hebei, China by China Unicom. The computing and network status were comprehensively used for service selection and path computation, which provided high quality computing service with the optimal service site and optimal forwarding path for vehicle terminal applications. Distributed routing mode provides real-time optimization and fast switching capabilities for delay-sensitive applications, such as the autonomous driving. Some non-delay sensitive applications, such as online media and entertainment, used centralized routing decision mode to achieve global resource scheduling.

Acknowledgements

The authors would like to thank Adrian Farrel, Peng Liu, Luigi Iannone, Christian Jacquenet and Yuexia Fu for their valuable suggestions to this document.

The authors would like to thank Yizhou Li for her early IETF work of Compute First Network (CFN) and Dynamic Anycast (Dyncast) which inspired the CATS work.

Contributors

The following people have substantially contributed to this document:

Yizhou Li
Huawei Technologies
Email: liyizhou@huawei.com

Dirk Trossen
Huawei Technologies
Email: dirk.trossen@huawei.com

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

Peter Willis
Email: pjw7904@rjt.edu

Philip Eardley
Email: philip.eardley@googlemail.com

Tianji Jiang
China Mobile
Email: tianjijiang@chinamobile.com

Markus Amend
Deutsche Telekom
Email: Markus.Amend@telekom.de

Guangping Huang
ZTE
Email: huang.guangping@zte.com.cn

Dongyu Yuan
ZTE
Email: yuan.dongyu@zte.com.cn

Xinxin Yi
China Unicom
Email: yixx3@chinaunicom.cn

Authors' Addresses

Kehan Yao
China Mobile
Email: yaokehan@chinamobile.com

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

Hang Shi
Huawei Technologies
Email: shihang9@huawei.com

Shuai Zhang
China Unicom
Email: zhangs366@chinaunicom.cn

Qing An
Alibaba Group
Email: anqing.aq@alibaba-inc.com

Computing-Aware Traffic Steering Working Group
Internet-Draft
Intended status: Informational
Expires: 8 May 2025

J. Jeong, Ed.
B. Mugabarigira
Sungkyunkwan University

4 November 2024

Applicability of Computing-Aware Traffic Steering to Intelligent Transportation Systems
draft-jeong-cats-its-use-cases-04

Abstract

This document describes the applicability of Computing-Aware Traffic Steering (CATS) to Intelligent Transportation Systems (ITS). CATS provides the steering of packets of a traffic flow for a specific service request toward the corresponding service instance at an edge computing server at a service site. CATS are applicable for Computing-Aware ITS including (i) Context-Aware Navigation Protocol (CNP) for Terrestrial Vehicles and Unmanned Aerial Vehicles (UAV), (ii) Edge-Assisted Cluster-Based MAC Protocol (ECMAC) for Software-Defined Vehicles, and (iii) Self-Adaptive Interactive Navigation Tool (SAINT) for Cloud-Based Navigation Services, and (iv) Cloud-Based Drone Navigation (CBDN) for Efficient Drone Battery Charging.

Table of Contents

1. Introduction

Nowadays, various networked services are provided by leveraging edge computing infrastructure. Either a closest or a lightest edge computing server (simply called an edge server) can be selected to serve a request service. In this trend, Computing-Aware Traffic Steering (CATS) is standardized to provide the steering of packets of a traffic flow for a specific service request toward the corresponding service instance at an edge server at a service site [I-D.ietf-cats-usecases-requirements][I-D.ietf-cats-framework].

This document proposes four use cases about the applicability of CATS for Computing-Aware Intelligent Transportation Systems (ITS). They are (i) Context-Aware Navigation Protocol for Terrestrial Vehicles and Unmanned Aerial Vehicles (UAV) [CNP-Vehicle] [CNP-UAV], (ii) Edge-Assisted Cluster-Based MAC Protocol for Software-Defined

Vehicles (SDV) [ECMAC], (iii) Self-Adaptive Interactive Navigation Tool (SAINT) for Cloud-Based Navigation Services [SAINT], and (iv) Cloud-Based Drone Navigation (CBDN) for Efficient Drone Battery Charging [CBDN].

2. Terminology

This document uses the terminology described in [I-D.ietf-cats-usecases-requirements] and [I-D.ietf-cats-framework]. In addition, the following terms are defined below:

* Context-Aware Navigation Protocol (CNP): It is an application protocol that enables either terrestrial vehicles (i.e., ground vehicles) or Unmanned Aerial Vehicles (UAV) to move in road networks or fly in the sky to maneuver safely without collisions, respectively [CNP-Vehicle][CNP-UAV].

* Edge-Assisted Cluster-Based MAC Protocol (ECMAC): It is a Media Access Control (MAC) protocol that enables Software-Defined Vehicles (SDV) to communicate with each other using Software-Defined Vehicular Networks with edge computing servers [ECMAC].

* Self-Adaptive Interactive Navigation Tool (SAINT): It is an application protocol that guides terrestrial vehicles to navigate efficiently towards their destination through the interaction between the vehicles and the vehicular cloud for navigation services [SAINT].

* Cloud-Based Drone Navigation (CBDN): It is an application protocol for efficient drone battery charging in drone networks by finding globally coordinated drone routes that minimize the total traffic delay in a drone network while reducing the overall Quick Battery-Charging Machine (QCM) congestion level [CBDN].

3. Vehicular Network Architecture

This section explains a vehicular network architecture for vehicles in Computing-Aware ITS.

Software-Defined Vehicles (SDV) include terrestrial vehicles and Unmanned Aerial Vehicles (UAV). The standardization and implementation of SDVs are performed by AUTOSAR [AUTOSAR], Eclipes SDV [Eclipse-SDV], and COVESA [COVESA]. These SDVs need to communicate with each other to avoid collisions or accidents.

Figure 1 shows a Vehicular Network Architecture for Software-Defined Vehicles (SDV) such as terrestrial vehicles and Unmanned Aerial Vehicles (UAV). This vehicular network architecture is based on the vehicular network architecture for IPv6 Wireless Access in Vehicular Environments (IPWAVE) in [RFC9365].

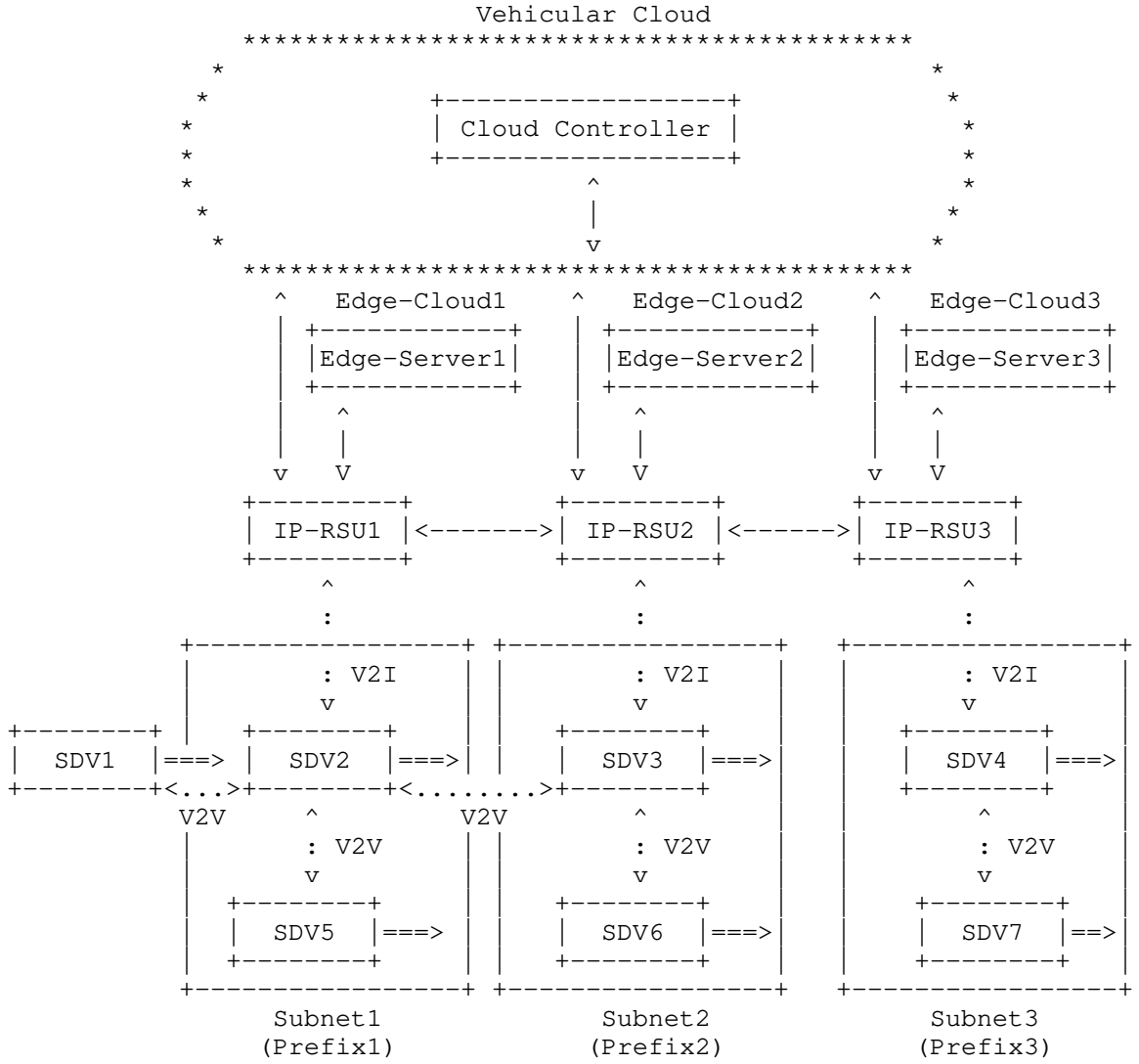


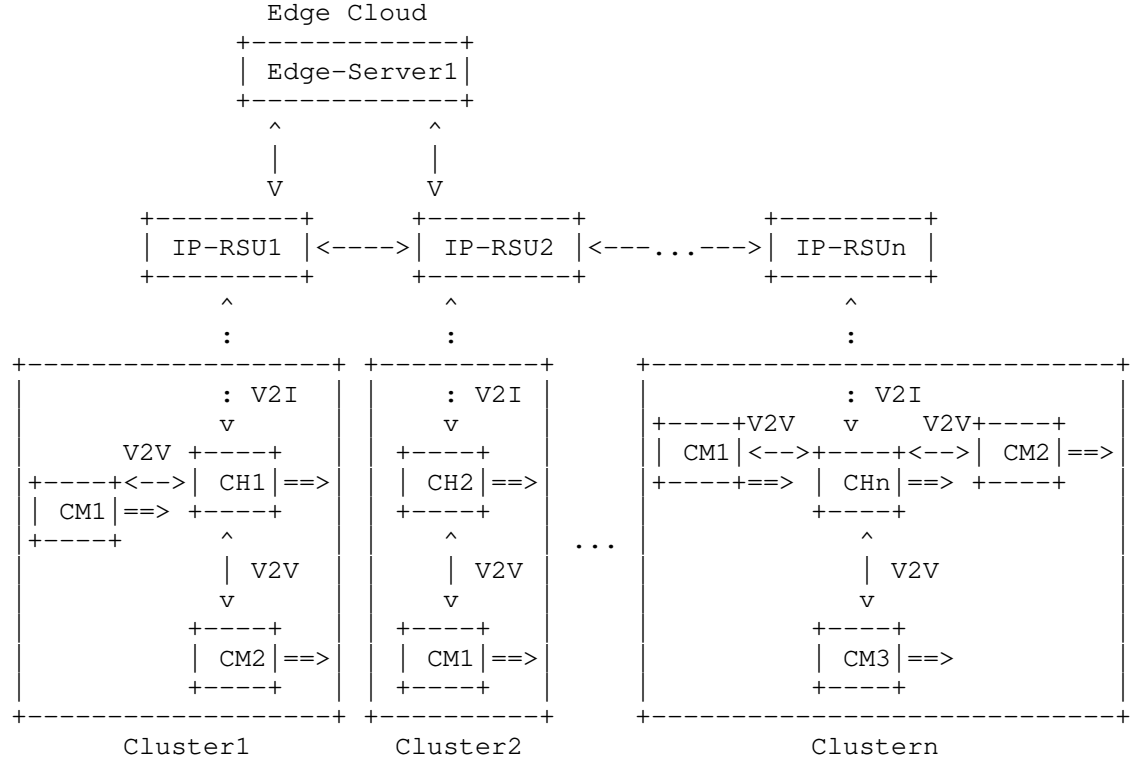
Figure 1: Vehicular Network Architecture for Software-Defined Vehicles

4. Use Cases

This section explains four use cases about the applicability of CATS to Computing-Aware ITS.

4.1. Context-Aware Navigation Protocol

A connected network of automated vehicles on roads can increase the driving safety of driverless vehicles (i.e., autonomous vehicles). The critical level of dangerous situations on the road while driving can be increased by the speed, orientation, and traffic density of the vehicles involved. Therefore, there is a need for a maneuvering mechanism that handles both the current driving vehicle and the oncoming vehicles headed toward an emergency zone (e.g., road hazard and road accident spot).



<----> Wired Link <....> Wireless Link ===> Moving Direction

Figure 2: The Illustration of Context-Aware Navigator Protocol

Context-Aware Navigation Protocol (CNP) enhances the safety of vehicles driving in urban roads [CNP-Vehicle][CNP-UAV]. Firstly, CNP includes a collision avoidance module that builds on both vehicular networks and on-board sensors to track vehicles' behaviors, and this module analyzes the driving risks to determine the necessary maneuvers in dangerous situations. Secondly, CNP establishes a collision mitigation strategy that limits the severity of collision damages in hazardous road during non-maneuverable scenarios. Through a theoretical analysis as well as extensive simulations, the effectiveness of CNP is shown in terms of the reduction of both communication overhead and the risk of road collisions.

To use CNP, vehicles need to report their mobility information (e.g., vehicle identifier, destination, current position, direction, and speed) to a central cloud or an edge cloud for a CNP-based vehicle collision avoidance service as shown in Figure 2. Service instances at either the edge cloud or the central cloud need to work for the vehicles. The packets with the mobility information per vehicle need to steered to an appropriate service instance for CNP. The service instance needs to provide a appropriate maneuver direction to each vehicle moving on the roadway.

Since the vehicle is moving along the roadway, to serve the vehicle for collision avoidance, a new service instance needs to be selected for it, considering the network delay between the vehicle and service instance and also computing resources for the service instance. For the service instances to continue to compute the maneuvers smoothly, they need to exchange the mobility information as context while the vehicles are moving and change their service instance over time. That is, the context migration should be supported in the CATS infrastructure having the central clouds and the edge clouds to foster service instances.

4.2. Edge-Assisted Cluster-Based MAC Protocol

Vehicular networks have emerged as a promising means to mitigate safety hazards in modern transportation systems. On highways, emergency situations associated with vehicles necessitate a reliable Media Access Control (MAC) protocol that can provide timely warnings of possible vehicle collisions.

An Edge-Assisted Cluster-Based MAC Protocol (ECMAC) is a vehicular MAC protocol for reliable and fast packet dissemination in software-defined vehicular networks [ECMAC]. To reduce the control messaging overhead for clustering, ECMAC separates the cluster control plane (i.e., managing cluster formation) from the data plane (i.e., actual data transmission and forwarding) by using a software-defined network controller in a cellular network edge server as illustrated in Figure 3.

For transmitting packets effectively and efficiently, ECMAC tries to channel interference minimization among adjacent clusters by using a joint optimization of channel assignment and a time slot scheduling. The joint optimization consists of two phases such as the channel assignment phase and the time slot allocation phase. In the first phase for the channel assignment, ECMAC allocates different wireless channels to the adjacent channels by minimizing the total inter-cluster interference by reusing the available channels. In the second phase for the time slot allocation, ECMAC uses a time-division multiple access (TDMA) schedule algorithm to guarantee a high reliability and a low latency. The TDMA schedule in ECMAC is determined by a joint optimization process in the cellular edge, which is formulated as a binary integer linear programming problem and solved by a heuristic approach based on the divide-and-conquer paradigm. This joint optimization process minimizes the signal interference by jointly considering channel assignment and time slot allocation, thereby ensuring reliable communication. Through extensive simulations, the effectiveness of ECMAC is demonstrated a higher delivery ratio of emergency packets than the legacy data delivery approaches.

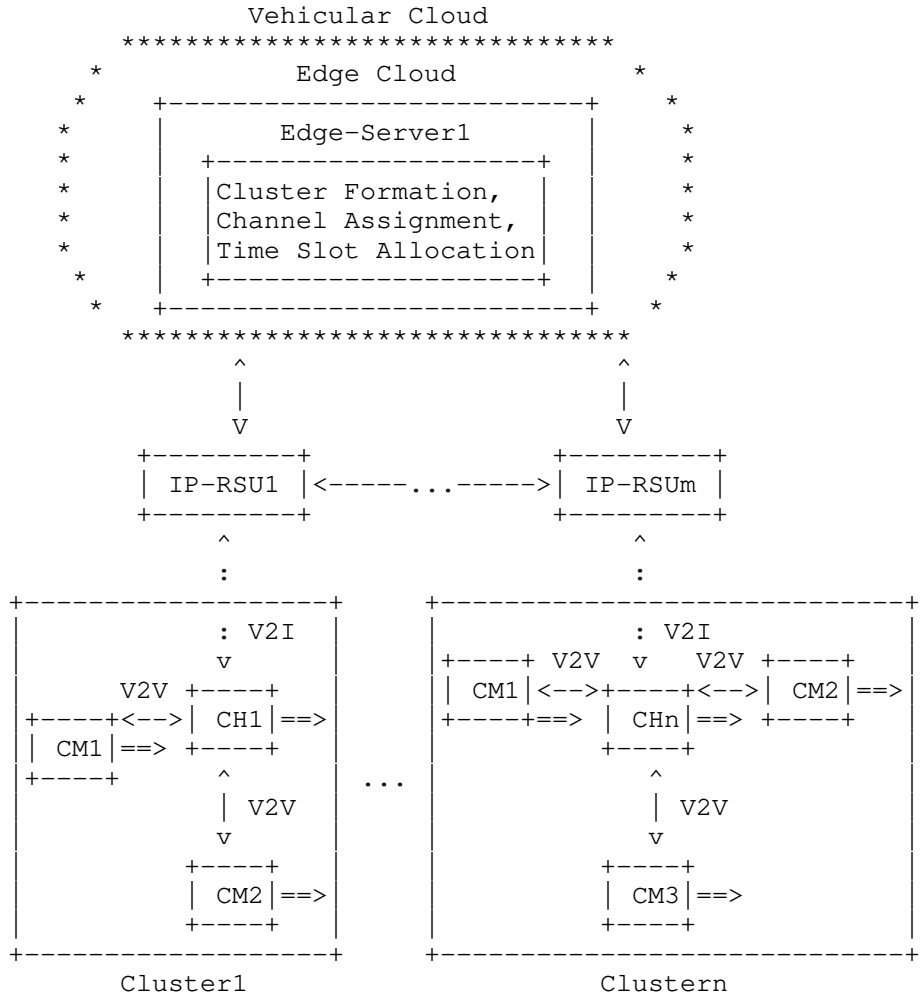


Figure 3: The Illustration of Edge-Assisted Clusterer-Based MAC Protocol

In ECMAC, the cellular network edge server can be implemented as a service instance in the CATS infrastructure. In the same way with CNP, service instances need to efficiently perform the context migration (e.g., mobility information and cluster membership) of vehicles so that they can continue to form clusters of vehicles, allocate wireless channels to the vehicles, and assign time slots to the vehicles over time.

4.3. Self-Adaptive Interactive Navigation Tool for Cloud-Based Navigation

Efficient navigation services are important in Intelligent Transportation Systems because they allow vehicles to move towards destinations quickly. For this efficient navigation, vehicles need to interact with a central cloud or an edge cloud in real time.

Self-Adaptive Interactive Navigation Tool (SAINT) is a cloud-based navigation guidance system for vehicular traffic optimization in road networks [SAINT]. The legacy navigation systems guide vehicles to take their navigation paths with real-time traffic statistics in road maps without considering the navigation paths of other vehicles. This uncoordinated navigation planning may incur traffic congestion in certain areas in the road networks.

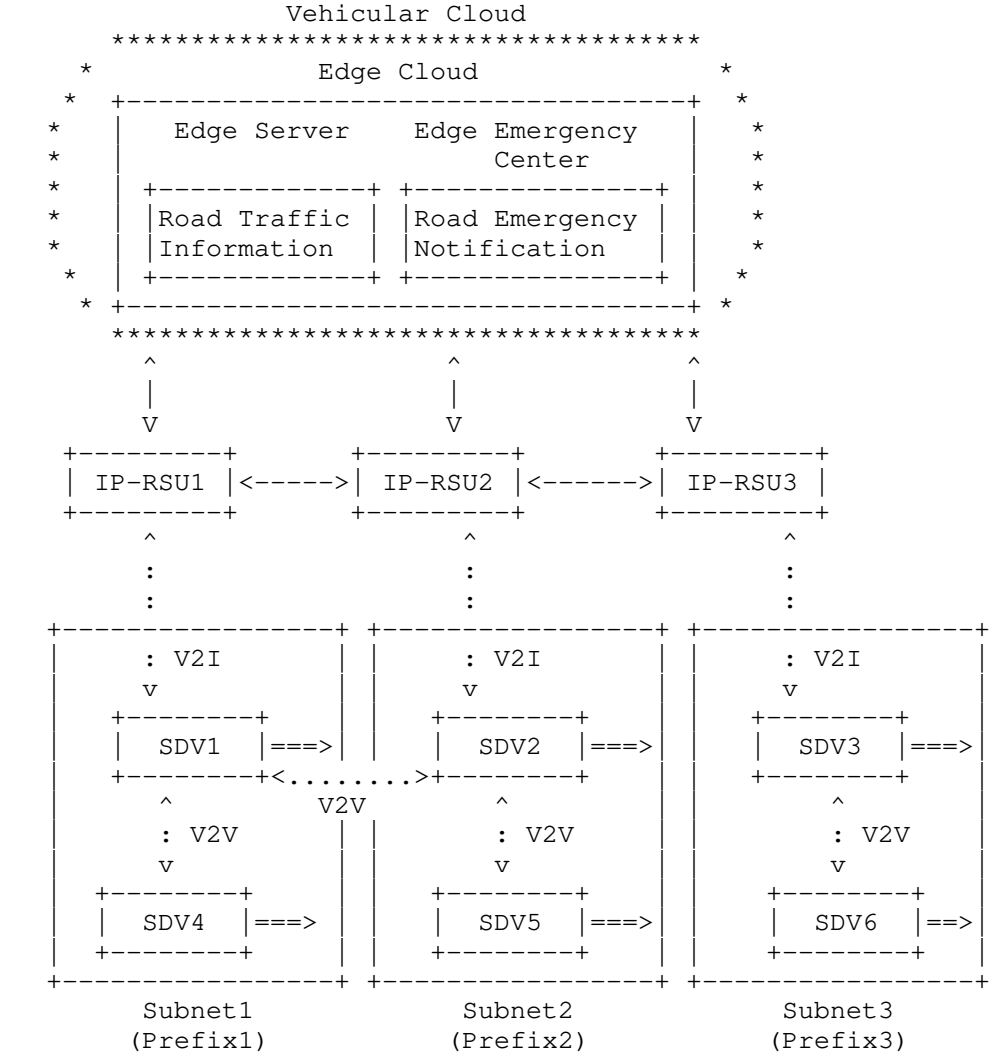


Figure 4: The Illustration of Self-Adaptive Interactive Navigation

On the other hand, SAINT uses a virtual metric called congestion contribution that estimates traffic congestion in each road segment in the current time and near-future time by considering the planned navigation paths of the vehices in the target road network. SAINT guides each vehicle to have a certain-level detour in order to make the whole road network have spread vehicular traffic and lessen possible traffic congestion in certain road segments or intersections.

For this cooperative navigation in SAINT, while vehicles are moving along the roadways, they need to send their periodic navigation queries and their mobility information to appropriate service instances in a central cloud or an edge cloud in the CATS infrastructure. The service instances need to process their navigation queries and reply to them with good navigation paths, considersing the road-wide traffic optimization as depicted in Figure 4. Due to the movement of the vehicles, the switching from a service instance to another service instance should be performed efficiently, considering the network delay between the service instance and each vehicle and the computing resources of the service instance.

SAINT can support the efficient delivery of emergency vehicles such as ambulance and fire engine to a road accident spot by the management of a congestion contribution matrix in a target road network [SAINTplus]. It can not only guide vehicles within the accident spot, but also can detour vehicles approaching the accident spot. This version of SAINT is called SAINT+.

4.4. Cloud-Based Drone Navigation (CBDN) for Efficient Battery Charging in Drone Networks

The growing popularity of Unmanned Aerial Vehicles (UAV) comes with a need to charge their battery at Quick Battery-Charging Machines (QCMs) due to their limited battery capacity. Without drone coordination, a drone's choice for its QCM may lead to congestion resulting from multiple drones selecting the same QCM, thus increasing the drones' battery-charging delay due to the queueing day at the QCM. This battery-charging delay leads to a long travel delay for each drone at the QCM. A Cloud-Based Drone Navigation (CBDN) efficiently determines drone routes to minimize the overall QCM congestion level for all QCMs in a target drone network [CBDN]. It finds globally coordinated drone routes that minimize the total travel delay in a drone network by reducing the overall QCM congestion level.

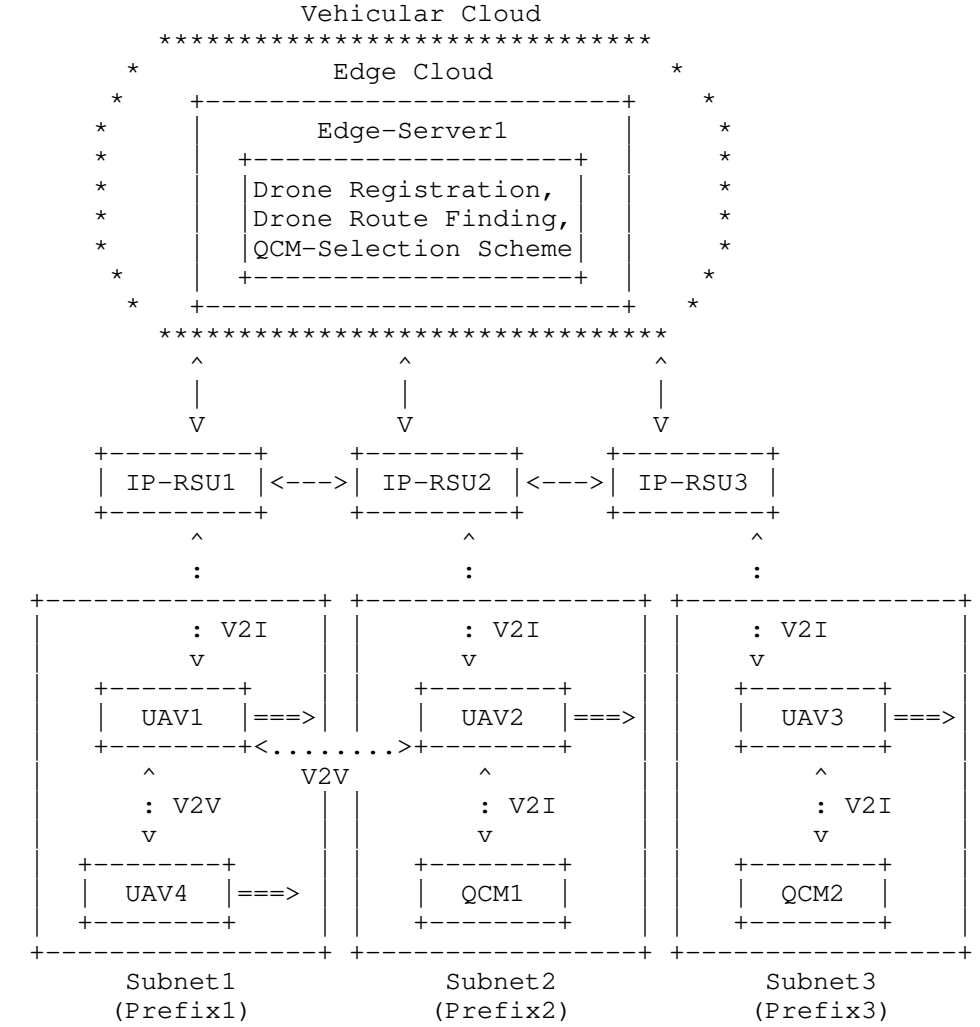


Figure 5: The Illustration of Cloud-Based Drone Navigation

An edge cloud in the CATS infrastructure with computing and storage resources need to compute the trajectories of the drones (i.e., drone routes), along with their average speeds, source positions, and destination positions, as well as the battery charing loads at the QCMs. The wireless communications between drones and infrastrucure nodes (e.g., edge server) can be either 5G and beyond 5G or wireless LAN, as illustated in Figure 5. Drones interact with the edge server to compute navigation paths regarding the drone network-wide traffic

optimization of all drones in the drone network. To decrease battery consumption, the drones only once report their mobility information (i.e., current position, destination, direction, and speed) to the edge computing decice to acquire their navigation paths.

Upon the commencement of the drone service, each drone reports its mobility information to the edge server. A drone's QCM reservation for battery charging acquires the most efficient shortest path regarding the drone-network-wide traffic optimization of all the drones in the drone network. For this drone-network-wide traffic optimization, a drone sends its mobility information to the edge server before its departure, and the edge server computes an optimal navigation path to the drone and notifies the drone of the path in run time.

5. Requirements

This section specifies the requirements for the applicability of CATS to ITS use cases in Section 4.

* R1: Dynamic mapping between a required service and a service instance. Both network delay and computing delay are considered over time.

* R2: Run-time context migration of vehicles between edge servers (i.e., service instances). Each vehicle's context (e.g., mobility information, communications parameters (e.g., channel, time slot)) is transferred to an appropriate service instance along with its movement over time.

* R3: Proactive load balancing among service instances considering the required Quality of Service (QoS) and Quality of Experience (QoE) for vehicles. The trajectories of vehicles are considered for such load balancing.

* R4: Dynamic clustering of geographically adjacent vehicles. Clusters of vehicles are dynamically reconstructed over time.

* R5: Dynamic network configuration for vehicles and network forwarding entities (e.g., base stations and switches/routers). In wireless networks, network resources (e.g., channel and time slot) per vehicle are dynamically configured by base stations. In wired networks, a network slice from a base station to a service instance are dynamically adjusted for each vehicle.

* R6: Differentiated packet scheduling for service types. Packets of real-time services (e.g., autonomous driving) and packets of non-real-time services (e.g., infotainment) are handled differently.

6. IANA Considerations

This document does not require any IANA actions.

7. Security Considerations

The same security considerations for Computing-Aware Traffic Steering (CATS) are applicable to the use cases for the Computing-Aware ITS [I-D.ietf-cats-usecases-requirements] [I-D.ietf-cats-framework].

8. References

8.1. Normative References

[RFC9365] Jeong, J., Ed., "IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases", RFC 9365, DOI 10.17487/RFC9365, March 2023, <https://www.rfc-editor.org/info/rfc9365>.

8.2. Informative References

[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-04, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-usecases-requirements-04>.

[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-04, 17 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-04>.

[AUTOSAR] "AUTOSAR Adaptive Platform", Available: https://www.autosar.org/standards/adaptive-platform, March 2024.

[Eclipse-SDV]
"Eclipse Software Defined Vehicle Working Group Charter", Available: https://www.eclipse.org/org/workinggroups/sdv-charter.php, March 2024.

[COVESA] "Connected Vehicle Systems Alliance", Available: https://covesa.global/, March 2024.

[CNP-Vehicle]
Mugabarigira, B., Shen, Y., Jeong, J., Oh, T., and H. Jeong, "Context-Aware Navigation Protocol for Safe Driving in Vehicular Cyber-Physical Systems", IEEE Transactions on Intelligent Transportation Systems, Volume 24, Issue 1, Available: https://ieeexplore.ieee.org/document/9921182, January 2023.

[CNP-UAV] Mugabarigira, B. and J. Jeong, "Context-Aware Navigation Protocol for Safe Flying of Unmanned Aerial Vehicles", KICS Winter Conference, Available: http://iotlab.skku.edu/publications/international-journal/CNP-TITS-2023.pdf, January 2024.

[ECMAC] Shen, Y., Jeong, J., Jun, J., Oh, T., and Y. Baek, "ECMAC: Edge-Assisted Cluster-Based MAC Protocol in Software-Defined Vehicular Networks", IEEE Transactions on Vehicular Technology, Volume 73, Issue 9, Available: https://ieeexplore.ieee.org/document/10505005, September 2024.

[SAINT] Jeong, J., Jeong, H., Lee, E., Oh, T., and D. Du, "SAINT: Self-Adaptive Interactive Navigation Tool for Cloud-Based Vehicular Traffic Optimization", IEEE Transactions on Vehicular Technology, Volume 65, Issue 6, Available: https://ieeexplore.ieee.org/document/7243355, June 2016.

[SAINTplus]
Shen, Y., Lee, J., Jeong, H., Jeong, J., Lee, E., and D. Du, "SAINT+: Self-Adaptive Interactive Navigation Tool+ for Emergency Service Delivery Optimization", IEEE Transactions on Intelligent Transportation Systems, Volume 19, Issue 4, Available: https://ieeexplore.ieee.org/document/7953571, April 2018.

[CBDN] Kim, J., Kim, S., Jeong, J., Kim, H., Park, J., and T. Kim, "CBDN: Cloud-Based Drone Navigation for Efficient Battery Charging in Drone Networks", IEEE Transactions on

Intelligent Transportation Systems, Volume 20, Issue 11, Available: https://ieeexplore.ieee.org/document/8574043, November 2019.

Appendix A. Changes from draft-jeong-cats-its-use-cases-03

The following changes are made from draft-jeong-cats-its-use-cases-03:

* This version specifies the requirements for the applicability of CATS to ITS use cases in Section 5.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Ministry of Science and ICT (MSIT) (No. RS-2024-00398199 and RS-2022-II221015).

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (No. 2023R1A2C2002990).

Contributors

This document is made by the group effort of CATS WG, greatly benefiting from inputs and texts by Peng Liu (China Mobile), Yong-Geun Hong (Daejeon University), and Joosang Youn (Dong-Eui University). The authors sincerely appreciate their contributions.

The following are coauthors of this document:

Juwon Hong
Department of Computer Science & Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4106
Email: hongju2024@skku.edu
URI: http://iotlab.skku.edu/people-Joo-Won-Hong.php

Yiwen Shen
Department of Computer Science & Engineering
Sungkyunkwan University

2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4106
Email: chrisshen@skku.edu
URI: https://chrisshen.github.io/

Authors' Addresses

Jaehoon Paul Jeong (editor)
Department of Computer Science & Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4957
Email: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

Bien Aime Mugabarigira
Department of Electrical & Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 10 5964 8794
Email: bienaime@skku.edu
URI: http://iotlab.skku.edu/people-Bien-Aime.php

cats
Internet-Draft
Intended status: Informational
Expires: 24 April 2025

L. Contreras
Telefonica
M. Watts
Verizon
T. Jiang
China Mobile

21 October 2024

Compute-Aware Traffic Steering for Midhaul Networks
draft-lcmw-cats-midhaul-02

Abstract

Computing-Aware Traffic Steering (CATS) takes into account both computing and networking resource metrics for selecting the appropriate service instance to forwarding the service traffic.

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 24 April 2025.

Table of Contents

1. Introduction

The radio functional split architecture proposed by O-RAN [ORAN-Arch] functionally separates the processing of the mobile radio signal originally performed in a single radio base station by placing functionality in three entities, namely the Radio Unit (RU), the Distributed Unit (DU) and the Centralized Unit (CU). Both DU and CU are typically deployed as service functions on virtualized compute nodes in the network.

The network segment between RU and DU is known as Fronthaul (FH), while the network segment between DU and CU is known as Midhaul (MH), or F1 interface according to 3GPP terminology. Both FH and MH have specific needs and characteristics in terms of latency and bandwidth, constrained by the nature of the data payload and the protocols intrinsic for the support of the radio functional split. More details can be found in [ORAN-Req]. The requirements on the FH are much stringent that the ones in MH.

In the current O-RAN framework, a DU selects a CU and then creates the association DU <â\200\223> CU-UP (CU User Plane) in a dynamic way. Such association is established before a UE (or end device) comes to register with the mobile operator via the DU, and then the associated CU. From an architectural point of view, it is possible to consider scenarios where traffic flows from the DUs can be delivered to different CUs depending on the compute and network metrics observed during runtime. It is in these situations where CATS proposition can play a distinctive role at the time of ensuring proper delivery of the midhaul traffic and its processing. Thus, the DU <â\200\223> CU-CP can be dynamic in a way that a DU might optimally select a CU based on compute and network metrics.

1.1. Terminology

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 RFC 2119 [RFC2119].

In addition, this document uses the terms defined in [I-D.ietf-cats-framework].

2. Midhaul Scenario

The connection of RU, DU and CU can be performed by means of an IP-based aggregation network. In O-RAN terminology [ORAN-Transport], the aggregation routers acting as PE-nodes are called Transport Network Elements (TNEs). The control and management of the TNEs is performed by a Transport Network Manager (TNM) [ORAN-TransportManagement]. Figure 1 illustrates a packet-switched based aggregation network in O-RAN.

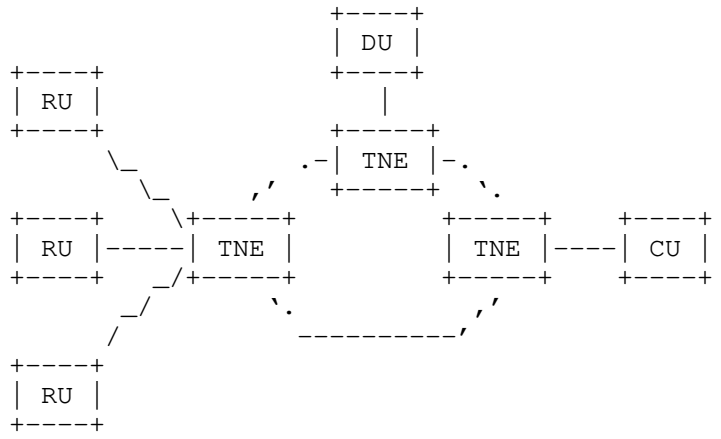


Figure 1: Midhaul Scenario

The FH segment connecting RUs and DUs is typically static in the sense that RUs are anchored with the same DU along the time. However, in the case of MH, the association between DUs and CUs could be more dynamic, subject to runtime situations such as DU and CU load, protection, workload migration (in the case of virtualized CU), energy efficiency, etc.

It is in these situations where the steering of the flows between DU and CU can take into consideration both service (including compute) and network metrics, as proposed by CATS. The focus in this document refers to the user plane of DU and CU connection (i.e., CU-UP).

The CUs can be deployed in different regions of the network, representing different service instances deployed in distinct service sites. For the illustration of the scenario, Figure 2 considers a number of CU instances in different Data Centers (DCs) and a DU running on a server, all of them interconnected by an aggregation network. Note that the DU could also run as an instance in a DC or even be a physical appliance.

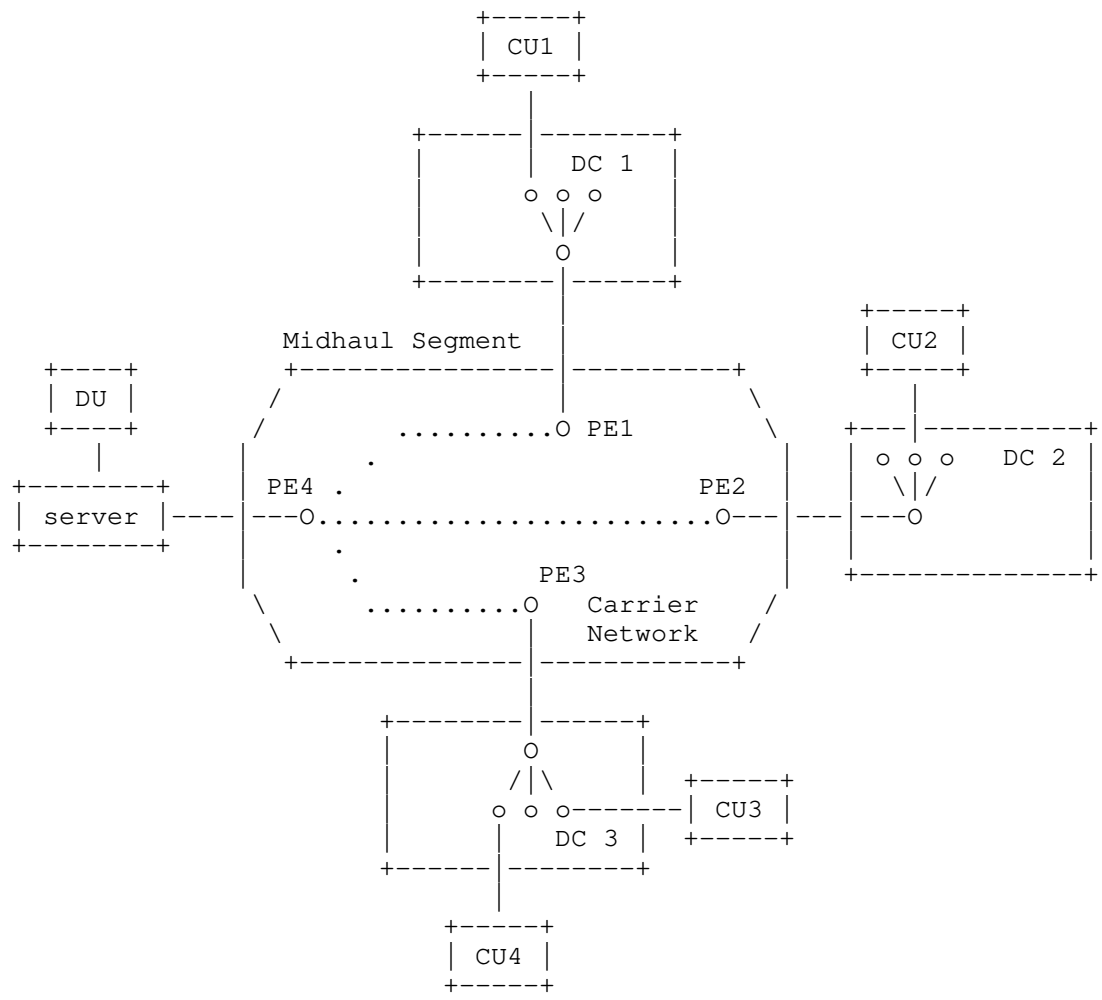


Figure 2: Midhaul Scenario

The aggregation network is IP-based, so the MH is realized by means of packet-switching technologies. This is consistent with the assumption in CATS that the underlay technology is IP/MPLS network. Figure 3 (according to the specification of the F1 / midhaul interface in [TS38.470] by 3GPP) illustrates the concern of CATS in the connection between DU and CU User Plane (CU-UP), considering as example an MPLS-based VPN connectivity.

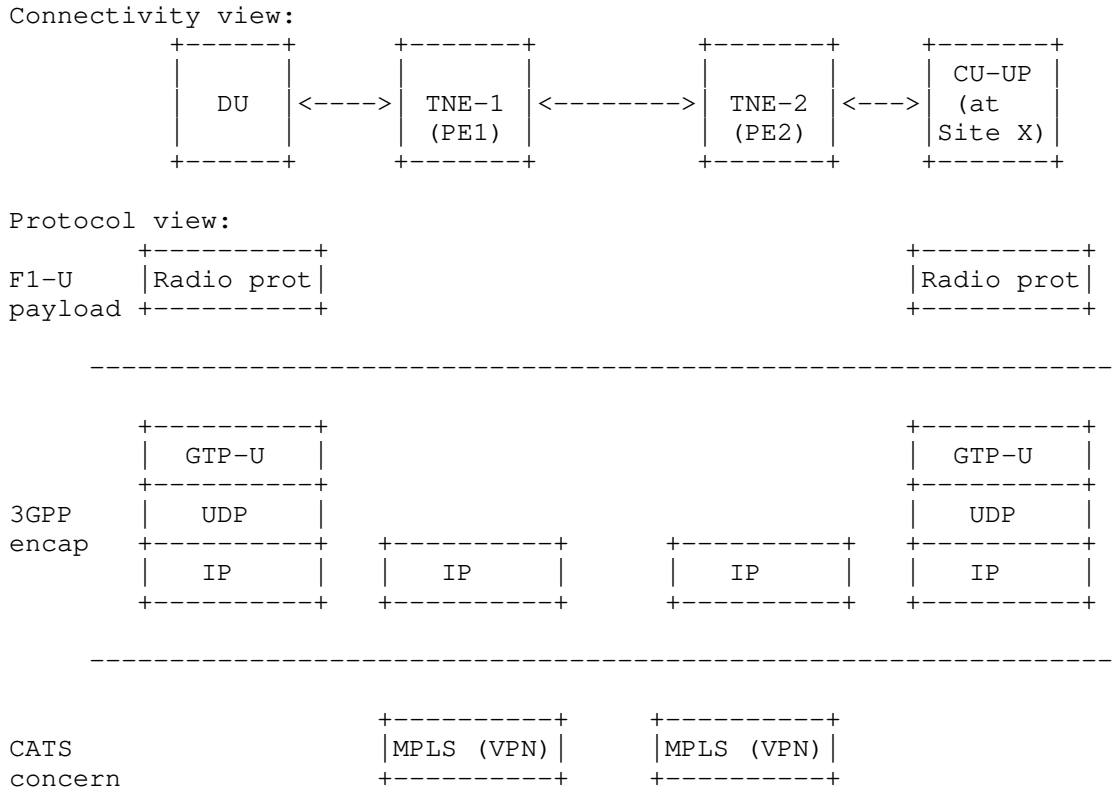


Figure 3: CATS concern

3. CATS framework applicability for Midhaul

The DU traffic cannot be separated in different flows. That is, the payload between DU and CU cannot be discriminated in individual flows since the payload represents a pre-processed analog radio signal, which will be entirely processed by the CU for obtaining the particular end-user flows. In this situation, the steering decision for the selection of a particular CU instance applies to the entire DU traffic. This simplifies the traffic classification since all the traffic from a DU is forwarded to the CU until any change is needed.

Note: Since all the midhaul traffic has the same service instance as destination (until any change applies), it is not necessary in this particular case the usage of CS-ID for accessing the service. The traffic classification is simple because all packets belong to the same service request.

The PE nodes (being TNEs in O-RAN terminology) in Figure 2 play the role of CATS-Forwarders. Each DC is expected to count with a CATS Service Metric Agent (C-SMA), while the network part is expected to count with a CATS Network Metric Agent (C-NMA). These agents will report different metrics and data to the CATS Path Selector (C-PS), which in this case can be assumed to be part of the TNM (i.e., considering that a centralized deployment model is followed, with the TNM playing the role of centralized control and management element).

Example of metrics related to compute could be the CPU average utilization or the memory usage of every CU-UP instance [ORAN-OCloud].

Figure 4 maps the CATS framework to the midhaul case.

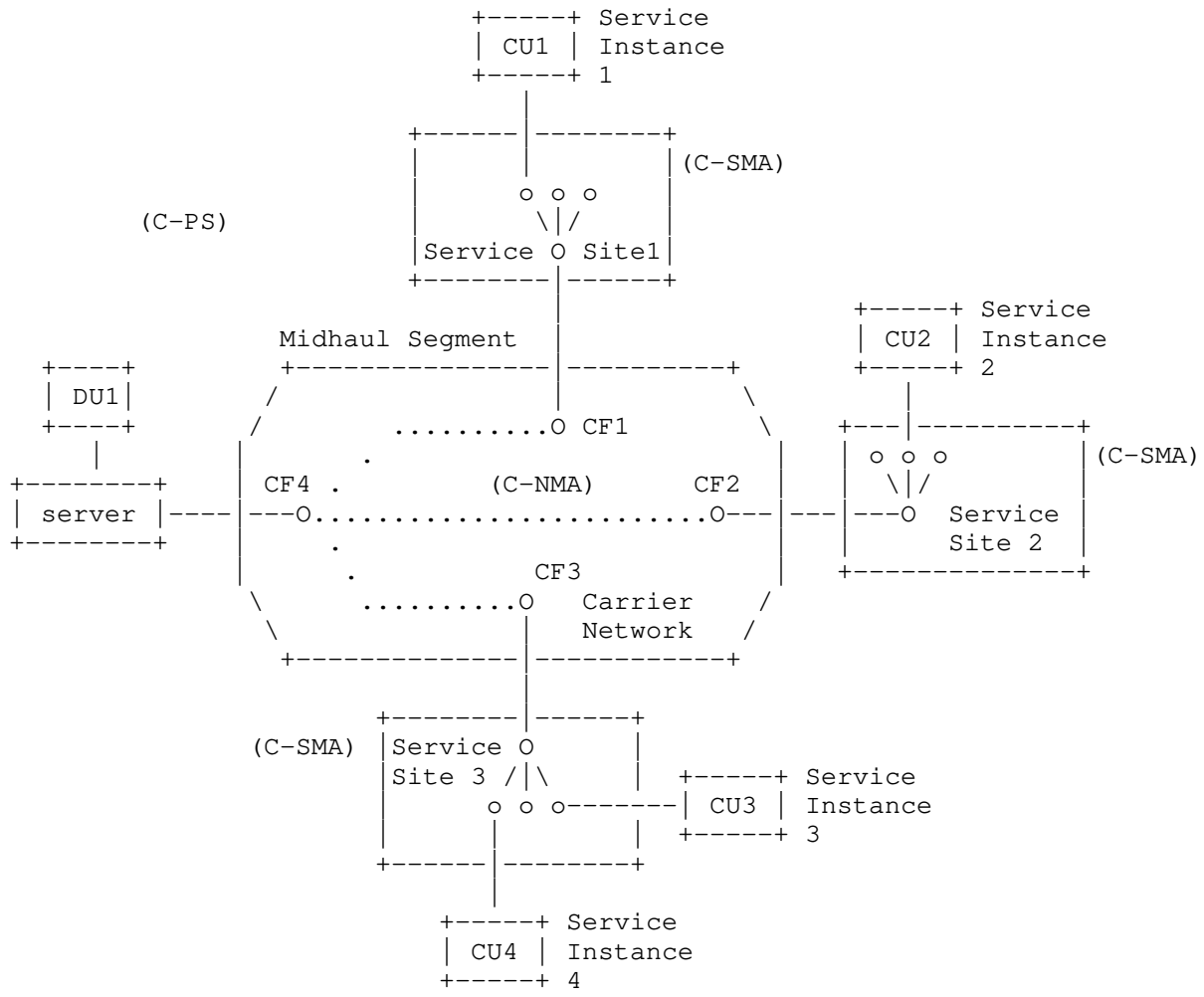


Figure 4: CATS applicability to Midhaul Scenario

3.1. Control plane interactions between O-RAN and IETF management entities

The connectivity between O-RAN radio functional entities is assumed to be managed by a Transport Network Manager (TNM) in charge of the control and management of the network. The interplay between the O-RAN Service and Management Orchestrator (SMO) and the TNM is currently under definition. The TNM function is assumed to be performed following IETF specifications. That role could be played, for instance, by the Network Slice Controller as defined in [RFC9543] for the provision of network slice services. Figure 5 represents the

relationship between O-RAN SMO and the TNM.

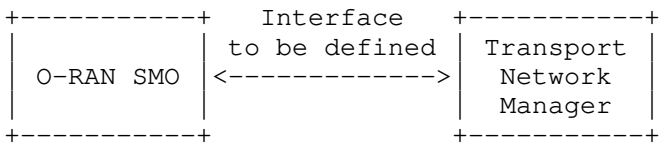


Figure 5: Interworking between SMO and TNM

For the specific case of CU-UP instance selection, every instance can be associated to various kinds of runtime information, i.e., including both (i) network metrics like bandwidth, delay, path-loss, reliability, etc., and (ii) compute or service metrics (of the CU-UP) like CPU load, memory, storage, service-load, etc. That information can assist on the selection of the most convenient CU-UP instance for a given DU. After the selection of a given instance, the O-RAN SMO will proceed to the necessary configurations on the O-RAN functional entities and instruct the TNM for performing the traffic steering of the service flows. The TNM, in consequence, represents the CATS entities necessary for the selection of the steering path and the forwarding of the traffic on it.

3.2. Example of connectivity based on IETF Network Slice Service

The connectivity in the MH segment could be realized for instance by means of IETF Network Slice Services [RFC9543], as described in [I-D.ietf-teas-5g-network-slice-application] and [I-D.ietf-teas-5g-ns-ip-mpls], according to the Service Level Objectives of the Midhaul traffic. With that connectivity in place for each of the possible CUs as Service Instances, the C-PS could decided which slice to use for delivering the traffic to a specific CU. Note that the realization of the IETF Network Slice Service could be performed either by means of a common slice for connecting the DU with all the CUs, or a slice per DU to CU connection. Once the C-PS takes deicsion on which CU (or Service Instance) deliver all the DU traffic, a policy could be applied (e.g., usage of the IP address of the CU-UP instance as match criteria in [I-D.ietf-teas-ietf-network-slice-nbi-yang]) for mapping the DU traffic to the proper connectivty construct of the IETF Network Slice Service.

Thus, the IETF Network Slice Service could be defined as hub-and-spoke from each DU to any of the CU-UP instances, and realized, e.g., by means of a VPN. A potential definition of the slice service using [I-D.ietf-teas-ietf-network-slice-nbi-yang]) could be as follows in Figure 6:

```
            "connection-groups": {
              "connection-group": [
                {
                  "id": "matrix1",
                  "connectivity-type": "ietf-vpn-common:hub-spoke",
                  "connectivity-construct": [
                    {
                      "id": "1",
                      "p2mp-sender-sdp": "du1",
                      "p2mp-receiver-sdp": [
                        "cu-up1",
                        "cu-up2",
                        "cu-up3",
                        "cu-up4"
                      ],
                      },

                      "status": {}
                    }
                  ]
                }
              }
```

Figure 6: Definition of the CATS steering paths as IETF NEtwork Slice Service

Moreover, based on the metrics collected for both network and compute, the C-PS could take the decision of steering the traffic of the DU towards a particular CU-UP instance, properly configuring the math criteria, setting it to the destination IP address of the CU-UP instance of interest, as exemplified in Figure 7.

```
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-nss:destination-ip-prefix",
                      "value": ["2001:db8::1/64"],
                      "target-connection-group-id": "matrix1"
                    }
                  ]
                }
```

Figure 7: Enforcement of the path steering leveraging on match-criteria

4. Open points for discussion

This version is an initial attempt of applicability of CATS for Midhaul scenarios as defined in O-RAN. The followoing are identified open points for further discussion, which will be elaborated in next versions of the document.

* Actions / situations changing the service affinity in the case of midhaul (and potential interaction with O-RAN specific service orchestration capabilities).

* Provide insights of control plane interactions (O-RAN SMO with TNM as IETF NSC [RFC9543])

5. Security Considerations

Same security considerations as in [I-D.ietf-cats-framework] apply also here.

6. Acknowledgements

TBC

7. References

7.1. Normative References

[RFC9543] Farrel, A., Ed., Drake, J., Ed., Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "A Framework for Network Slices in Networks Built from IETF Technologies", RFC 9543, DOI 10.17487/RFC9543, March 2024, <https://www.rfc-editor.org/info/rfc9543>.

7.2. Informative 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-04, 17 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-04>.

[I-D.ietf-teas-5g-network-slice-application]
Geng, X., Contreras, L. M., Rokui, R., Dong, J., and I. Bykov, "IETF Network Slice Application in 3GPP 5G End-to-End Network Slice", Work in Progress, Internet-Draft, draft-ietf-teas-5g-network-slice-application-03, 10 June 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-5g-network-slice-application-03>.

[I-D.ietf-teas-5g-ns-ip-mpls]
Szarkowicz, K. G., Roberts, R., Lucek, J., Boucadair, M., and L. M. Contreras, "A Realization of Network Slices for 5G Networks Using Current IP/MPLS Technologies", Work in Progress, Internet-Draft, draft-ietf-teas-5g-ns-ip-mpls-13, 11 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-5g-ns-ip-mpls-13>.

[I-D.ietf-teas-ietf-network-slice-nbi-yang]
Wu, B., Dhody, D., Rokui, R., Saad, T., and J. Mullooly, "A YANG Data Model for the RFC 9543 Network Slice Service", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slice-nbi-yang-16, 28 August 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-ietf-network-slice-nbi-yang-16>.

[ORAN-Arch]
"O-RAN Architecture Description, V11.00", February 2024.

[ORAN-OCloud]
"O-Cloud Information Model, V01.00", June 2024.

[ORAN-Req] "O-RAN Xhaul Transport Requirements, V01.00", February 2021.

[ORAN-Transport]
"O-RAN Xhaul Packet Switched Architectures and Solutions, V07.00", February 2024.

[ORAN-TransportManagement]
"O-RAN Management interfaces for Transport Network Elements, V07.00", October 2023.

[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/info/rfc2119>.

[TS38.470] "F1 general aspects and principles, V16.2.0", 2020.

Authors' Addresses

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

Mark Watts
Verizon
Email: mark.t.watts@verizon.com

Tianji Jiang
China Mobile
Email: tianjijiang@chinamobile.com

cats
Internet-Draft
Intended status: Informational
Expires: 9 January 2025

J. Wang
Y. Fu
China Mobile
C. Li
Huawei Technologies

8 July 2024

Green Challenges in Computing-Aware Traffic Steering (CATS)
draft-wang-cats-green-challenges-04

Abstract

As mobile edge computing networks sink computing tasks from cloud data centers to the edge of the network, tasks need to be processed by computing resources close to the user side. Therefore, CATS was raised. Reducing carbon footprint is a major challenge of our time. Networks are the main enablers of carbon reductions. The introduction of computing dimension in CATS makes it insufficient to consider the energy saving of network dimension in the past, so the green for CATS based on network and computing combination is worth exploring. This document outlines a series of challenges and associated research to explore ways to reduce carbon footprint and reduce network energy based on CATS.

Table of Contents

1. Introduction

With the continuous development and progress of the Internet, a large amount of computing resources is required to complete data processing. In order to disperse the pressure of cloud data centers, computing power gradually moves from the center to the edge, forming scattered computing resources in mobile networks. In order to make full use of scattered computing resources and provide better services, Computing-Aware Traffic Steering (CATS) is proposed to support steering the traffic among different edge sites according to both the real-time network and computing resource status as mentioned in [I-D.ietf-cats-usecases-requirements]. It requires the network to be aware of computing resource information and select a service instance based on the joint metric of computing and networking.

Green has become a global topic. The United Nations and the vast majority of governments agree that climate change and the need to curb greenhouse gas emissions are the major challenges of our time. Therefore, improving energy efficiency and reducing electricity

consumption are becoming increasingly important for society and many industries. The networking industry is no exception. The IETF conducted a study on the energy costs of the IETF meeting three times a year. The results showed that it was found that 99% of energy consumption came from air travel.

In addition, there are several papers that discuss green networks, and some work [I-D.cx-green-ps] summarizes the energy-saving possibilities that exist in the network. However, there is no discussion of joint optimization of green and energy savings with computing and networking. Therefore, this document outlines a series of challenges and related research to explore ways to reduce carbon emissions and reduce network energy based on CATS.

2. Definition of Terms

Computing-Aware Traffic Steering (CATS): Aiming at computing and network resource optimization by steering traffic to appropriate computing resources considering not only routing metric but also computing resource metric.

Service: A monolithic functionality that is provided by an endpoint according to the specification for said service. A composite service can be built by orchestrating monolithic services.

Service instance: Running environment (e.g., a node) that makes the functionality of a service available. One service can have several instances running at different network locations.

3. Challenges

Considering energy savings in CATS creates challenges in the following aspects

3.1. Computing Resource Energy Consumption Modeling

Computing resource status is considered in Cats, so it is necessary to research the modeling of computing resource energy consumption in order to save energy. The energy consumption of the equipment is different when the load is different. For example, the energy efficiency of equipment is different when it is not loaded or at full load. Therefore, it is also a challenge to consider which factors to consider when modeling the energy consumption of computing resources.

3.2. Joint Optimization of Computing and Network

The magnitude of computing energy consumption may differ from the magnitude of network energy consumption. Therefore, when computing and network are jointly optimized, how to weigh the two in joint optimization becomes a challenge. When the computing energy consumption is large enough, the impact of network energy consumption on the joint optimization results is negligible.

3.3. Service Experience Guarantee

The service experience that takes energy saving factors into account in CATS is distinct from the service experience that does not consider energy saving factors. The implementation of energy conservation may come with sacrifices in user service experience. Users have limitations on factors such as latency when making requests. Therefore, when conducting joint optimization, how to guarantee user service experience while conserving energy is also a challenge.

3.4. Energy Consumption of Other Equipment

The computing resources may be in the data center, edge computing nodes or others. In order to ensure the normal operation of network and computing equipment, the source of energy consumption is not only the equipment itself, but also some other equipment, such as :

Cool equipment : computing resources will emit heat into the air during operation. When the temperature is too high, the operation of the equipment will be affected. So refrigeration is required to reduce the temperature of the equipment to ensure that the equipment operates at a higher performance.

The normal running of computing resources are inseparable from the support of refrigeration equipment and other equipment. Therefore, when performing joint optimization of network and computing, the energy consumption generated by equipment other than network equipment and computing equipment should also be considered.

3.5. Evaluation of Computing Equipment Energy Consumption Performance

The energy efficiency level requirements of computing equipments can also be considered when performing traffic steering. Since there is no standardized definition of computing energy efficiency for different computing equipments. Therefore, it is difficult to consider the computing energy efficiency level of computing equipments when traffic steering.

3.6. Using of Green Energy

Green energy, also known as clean energy, refers to energy that does not emit pollutants and can be directly used for production and daily life, including nuclear energy and renewable energy. Renewable energy refers to energy sources that can be regenerated from raw materials, such as hydroelectric power, wind power, solar energy, bioenergy (biogas), geothermal energy (including geothermal and water sources), and tidal energy. Renewable energy does not have the possibility of energy depletion, therefore, the development and utilization of renewable energy are increasingly valued by many countries, especially those with energy shortages.

The development of green CATS cannot be separated from the use of green energy. Although the current use of green energy in network devices has a certain scale, the industry's consideration of the consumption of green energy for equipment in traffic steering is incomplete and further research is needed.

4. Observation

Recently, the document [I-D.cx-opsawg-green-metrics] gives some green networking metrics for network instrumentation to optimize the energy efficiency of the network. It divides the green metrics into four categories according to the subject of the metrics, as follows:

At the device/equipment level: The author considers three factors. The first are energy consumption metrics. Some of these metrics could be provided by the data sheet that comes with the device or could be measured simply in a lab, such as power consumption when idle, power consumption when fully loaded, power consumption at various loads and so on. The others are not fixed and need to be accounted according to the actual operation of the network equipment, such as current power consumption/kB (or gB), current power consumption/packet, power drawn since system started for the past minute and so on. The second is green metrics beyond energy consumption, Wich is related to the power source of the device and the environment in which the device is located. The third is related to network instrumentation virtualization. Nowadays, network instrumentation could be virtualized and hosted (for example) in data centers.

At the flow level: These metrics are related to flows, such as amortized energy consumed over the duration of the flow and Incremental energy consumed over the duration of the flow.

At the path level: These metrics can evaluate the energy consumption of paths and optimize these paths so that the overall footprint is minimized. The author gives some candidate metrics, such as energy rating of a path, current power consumption across a path and incremental power for a packet over a path.

At the network level: These metrics can reflect the energy usage of the entire network.

5. Conclusion

This document highlights the green challenges in Cats and summarizes the latest IETF work which is associated with green networking. As is well known, Cats not only considers network resource status, but also computing resource status. Therefore, energy consumption research of Cats can also consider both network and computing energy consumption from the device/equipment, path and network level.

6. Security Considerations

TBD.

7. IANA Considerations

TBD.

8. Acknowledgements

The authors would like to thank thank Adrian Farrel and Peng Liu for their valuable suggestions to this document. Additionally, the authors would also like to thank Alexander Clemm and Lijun Dong for their related work.

9. Informative References

[I-D.ietf-cats-usecases-requirements]
Yao, K., Trossen, D., Contreras, L. M., Shi, H., Li, Y., 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-03, 3 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-usecases-requirements-03>.

[I-D.cx-green-ps]
Clemm, A., Westphal, C., Tantsura, J., Ciavaglia, L., and M. Odini, "Challenges and Opportunities in Management for Green Networking", Work in Progress, Internet-Draft,

draft-cx-green-ps-02, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-cx-green-ps-02>.

[I-D.cx-opsawg-green-metrics]
Clemm, A., Dong, L., Mirsky, G., Ciavaglia, L., Tantsura, J., Odini, M., Schooler, E., Rezaki, A., and C. Pignataro, "Green Networking Metrics", Work in Progress, Internet-Draft, draft-cx-opsawg-green-metrics-02, 4 March 2024, <https://datatracker.ietf.org/doc/html/draft-cx-opsawg-green-metrics-02>.

Authors' Addresses

Jing Wang
China Mobile
No.32 XuanWuMen West Street
Beijing
100053
China
Email: wangjingjc@chinamobile.com

Yuexia Fu
China Mobile
No.32 XuanWuMen West Street
Beijing
100053
China
Email: fuyuexia@chinamobile.com

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing
100095
China
Email: c.l@huawei.com

cats
Internet-Draft
Intended status: Informational
Expires: 24 April 2025

J. Wang
China Mobile

21 October 2024

A Use case for Green Computing-Aware Traffic Steering
draft-wang-cats-usecase-green-00

Abstract

This draft describes a compute-aware use case for services with green energy requirements. This use case considers both network, computation and energy metrics when selecting a service instance.

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 24 April 2025.

Table of Contents

1. Introduction

As mobile edge computing networks sink computational tasks from cloud data centers to the edge of the network, tasks need to be processed by computational resources close to the user's end as mentioned in [I-D.ietf-cats-usecases-requirements]. Therefore, CATS is proposed. Reducing carbon emissions is a major challenge that needs to be faced in our time. The network is the main enabler to achieve the reduction of carbon emission.The introduction of computational dimension in CATS makes the previous energy saving by considering only the network dimension to be insufficient and hence green for CATS based on the association of network and computation is worth to be explored.

Recently, the GREEN WG was formed. It is chartered to explore use cases, derive requirements, and provide solutions for identifying and characterizing energy efficiency metrics, methods related to energy consumption of network devices, and optimizing energy efficiency across the network. network. There are also a number of contributions that explore green networks, and the document [I-D.wang-cats-green-challenges] summarizes a number of challenges faced by cats considering green.

This document provides a green cats use case.

2. Definition of Terms

Computing-Aware Traffic Steering (CATS): Aiming at computing and network resource optimization by steering traffic to appropriate computing resources considering not only routing metric but also computing resource metric.

Service: A monolithic functionality that is provided by an endpoint according to the specification for said service. A composite service can be built by orchestrating monolithic services.

Service instance: Running environment (e.g., a node) that makes the functionality of a service available. One service can have several instances running at different network locations.

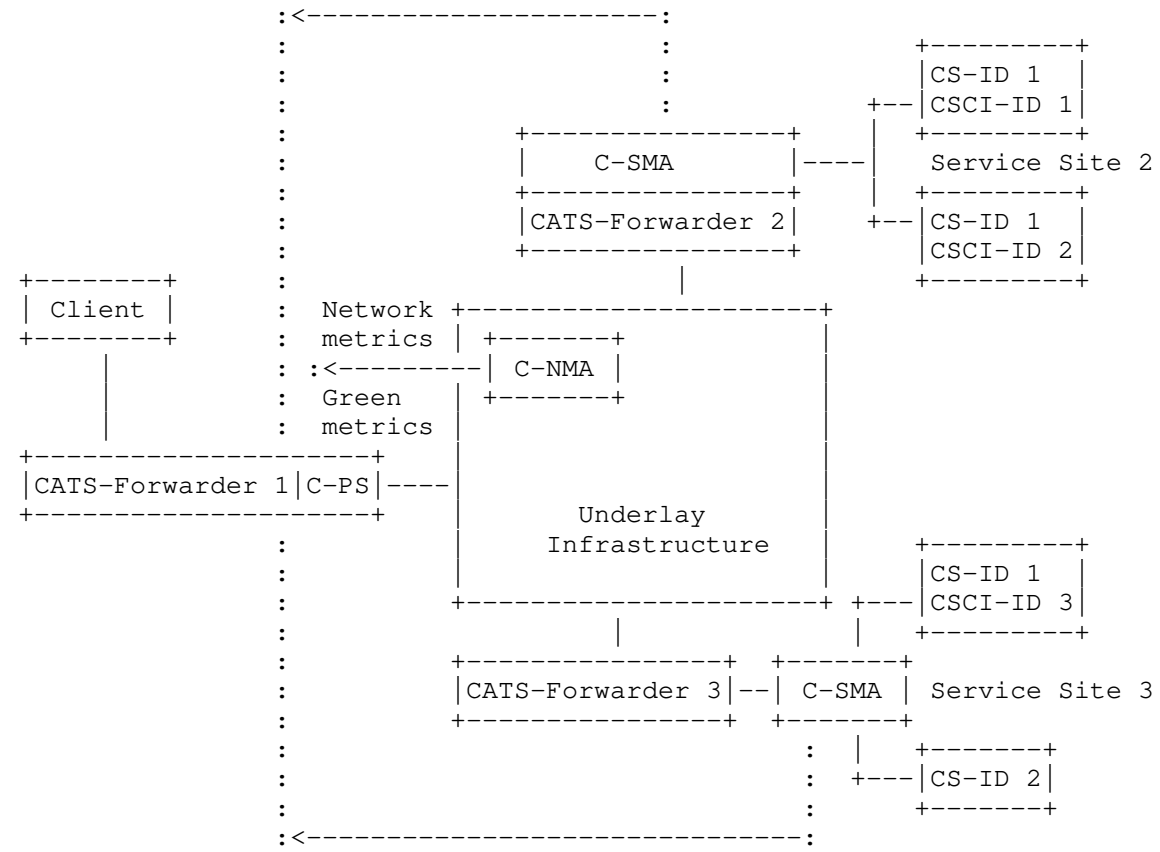
3. Use Case

Geared towards green computing-aware traffics Steering, the green metrics include the energy consumption of network devices as well as the energy consumption of computing resources. The following describes how green metrics are distributed under both distributed and centralized models.

3.1. Distributed Model

Figure 1 shows an example of how Green CATS metrics can be disseminated in the distributed model. In this way, green metrics are distributed among network devices directly using distributed protocols without interactions with a centralized control plane.

Service CS-ID 1, contact instance CSCI-ID 1 <computing metrics, green metrics>
Service CS-ID 1, contact instance CSCI-ID 2 <computing metrics, green metrics>



Service CS-ID 1, contact instance CSCI-ID 3 <computing metrics, green metrics>
Service CS-ID 2, <computing metrics, green metrics>

Figure 1: An Example of Green CATS Metric Dessimination in a Distributed Model

3.2. Centralized Model

In Figure 2, network metrics, computing metrics, and green metrics can be distributed in a centralized way. Green metrics are collected by the centralized control plane, and then the centralized control plane calculates the forwarding path corresponding to the energy efficiency demand request and synchronizes with the Ingress CATS-Forwarder.

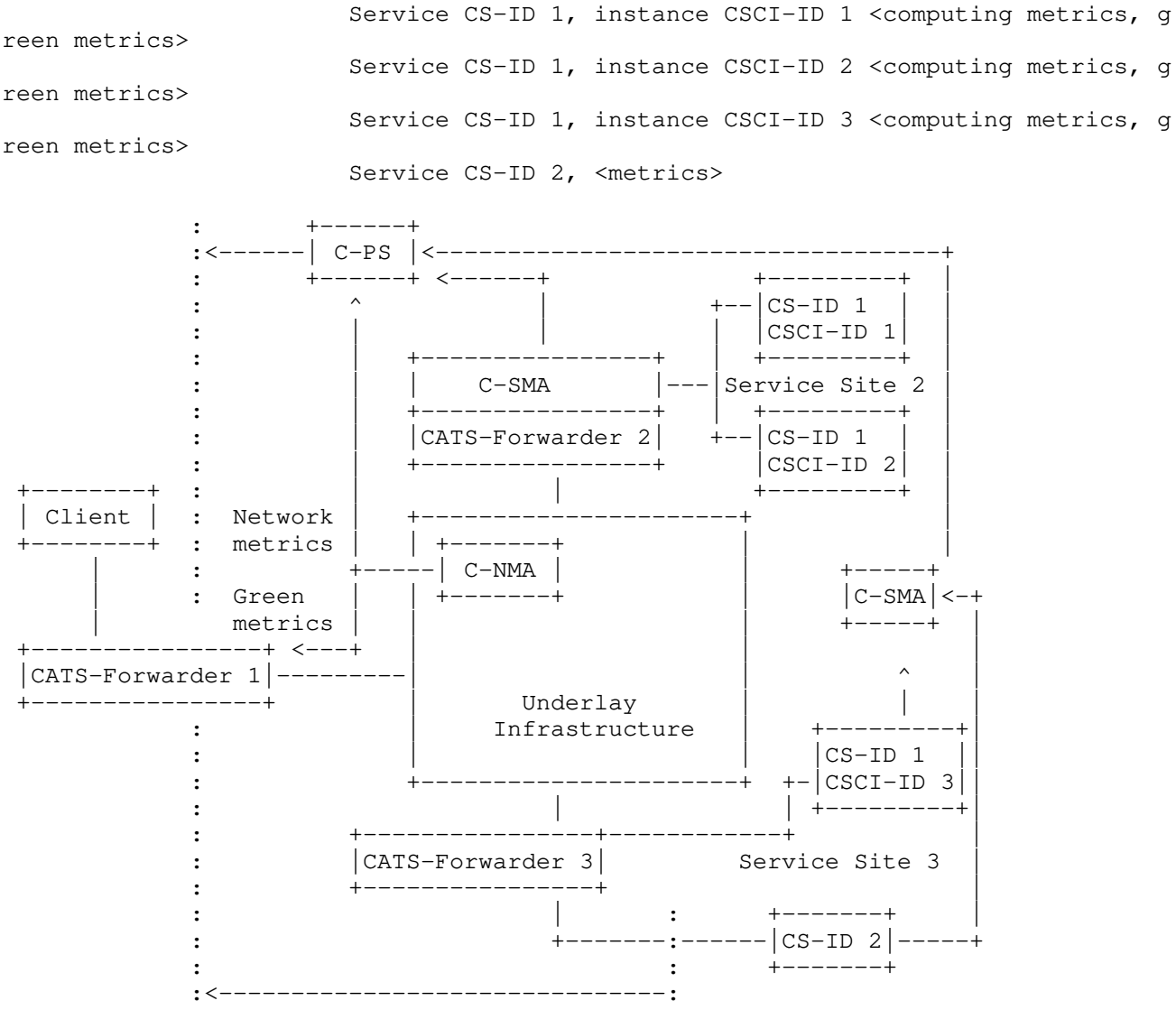


Figure 2: An Example of Green CATS Metric Distribution in a Centralized Model

4. Conclusion

This document gives a CATS use case related to green and further describes how green metrics can be distributed under both distributed and centralized models.

5. Security Considerations

TBD.

6. IANA Considerations

TBD.

7. Informative References

[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-04, 3 July 2024, <https://datatracker.ietf.org/api/v1/doc/document/draft-ietf-cats-usecases-requirements/>.

[I-D.wang-cats-green-challenges]
Wang, J., Fu, Y., and C. Li, "Green Challenges in Computing-Aware Traffic Steering (CATS)", Work in Progress, Internet-Draft, draft-wang-cats-green-challenges-04, 7 July 2024, <https://datatracker.ietf.org/doc/html/draft-wang-cats-green-challenges-04>.

Author's Address

Jing Wang
China Mobile
No.32 XuanWuMen West Street
Beijing
100053
China
Email: wangjingjc@chinamobile.com