CATS WG
Internet-Draft
Intended status: Standards Track
Expires: 2 September 2026

CJ. Bernardos
UC3M
A. Mourad
InterDigital
1 March 2026

AI/ML-Enabled Computing Aware Traffic Steering using IP address anchoring
draft-bernardos-cats-anchoring-aiml-selection-00

Abstract

The IETF CATS WG addresses the problem of how the network infrastructure can steer traffic between clients of a service and sites offering the service, considering both network metrics (such as bandwidth and latency), and compute metrics (such as processing, storage capabilities, and capacity).

This document describes solutions to enable the network to select the best site to instantiate a processing service (using distributed sensing as an application example), augmenting CATS enabled solutions that consider both connectivity and computing, to also consider AI/ML and data capabilities and governance policies.

Table of Contents

1. Introduction and Problem Statement

1.1. Use case scenario

There are sensing scenarios and use cases that involve a distributed sensing task, in which one ore multiple sensors participate, and that requires a supporting sensing service (e.g., fusing sensing measurements from different sensors, and/or applying AI/ML techniques to process and obtain an accurate sensing result). This sensing service needs to be executed on some sort of sensing processing/computing function that would typically require AI/ML capabilities to provide accurate results. Being capable of processing the results at a node that has been trained with the proper data and/or has the required computing capabilities for the AI/ML processing is key for the accuracy and timeliness of the sensing results. That adds one more requirement, in addition to the connectivity and computing ones, to be able to properly operate (and deliver timely the sensing results).

A terminal requests a sensing service with certain i) connectivity and computing associated requirements (CATS requirements), ii) AI/ML processing requirements, and potentially iii) sensing data governance requirements about privacy, security and trustworthiness. Multiple sites where the service can be instantiated exist in the domain where the terminal is attached. The network selects the best site to instantiate the service, instantiates it, and provides an IP address

to the terminal. This address is anchored at a router close (or at) the site where the selected service instance runs. Computing, connectivity, AI/ML and data governance related demands are met.

Note that this is just an example, other services would also benefit from compute and connectivity traffic steering. For the sake of having a simpler service, we can also consider an AR/VR/XR service where a terminal connected to the network needs to instantiate a service in the network to aid in the AR/VR/XR service by providing computing capabilities with latency constraints.

Note on terminology. In this document we use the old terminology in which by ICR we mean Ingress CATS-Forwarder [I-D.ietf-cats-framework], and by ECR we mean Egress CATS-Forwarder.

Figure 1 shows an exemplary scenario. There is a distributed sensing task (e.g., requested by an Application Function or Network Function). This involves one sensor function (hosted at terminal #1) and a sensing processing function (which can be potentially hosted at several service sites). In general, the sensor(s) might be of the same or different technology and might be connected to the same RAN or different ones (which might also be of different access technologies). The selection of the composition of the sensing group is out of the scope of this document.

In this particular example, the processing of the sensing data poses some specific requirements, not only in terms of general-purpose computing and connectivity (what we generally refer to as "CATS requirements"), but also in terms of AI/ML processing, data training capabilities (e.g., it is preferred to process the data at a site that has already a model trained fitting this specific scenario) and data policy/governance requirements. In this example, service site #n-1 is the one selected.

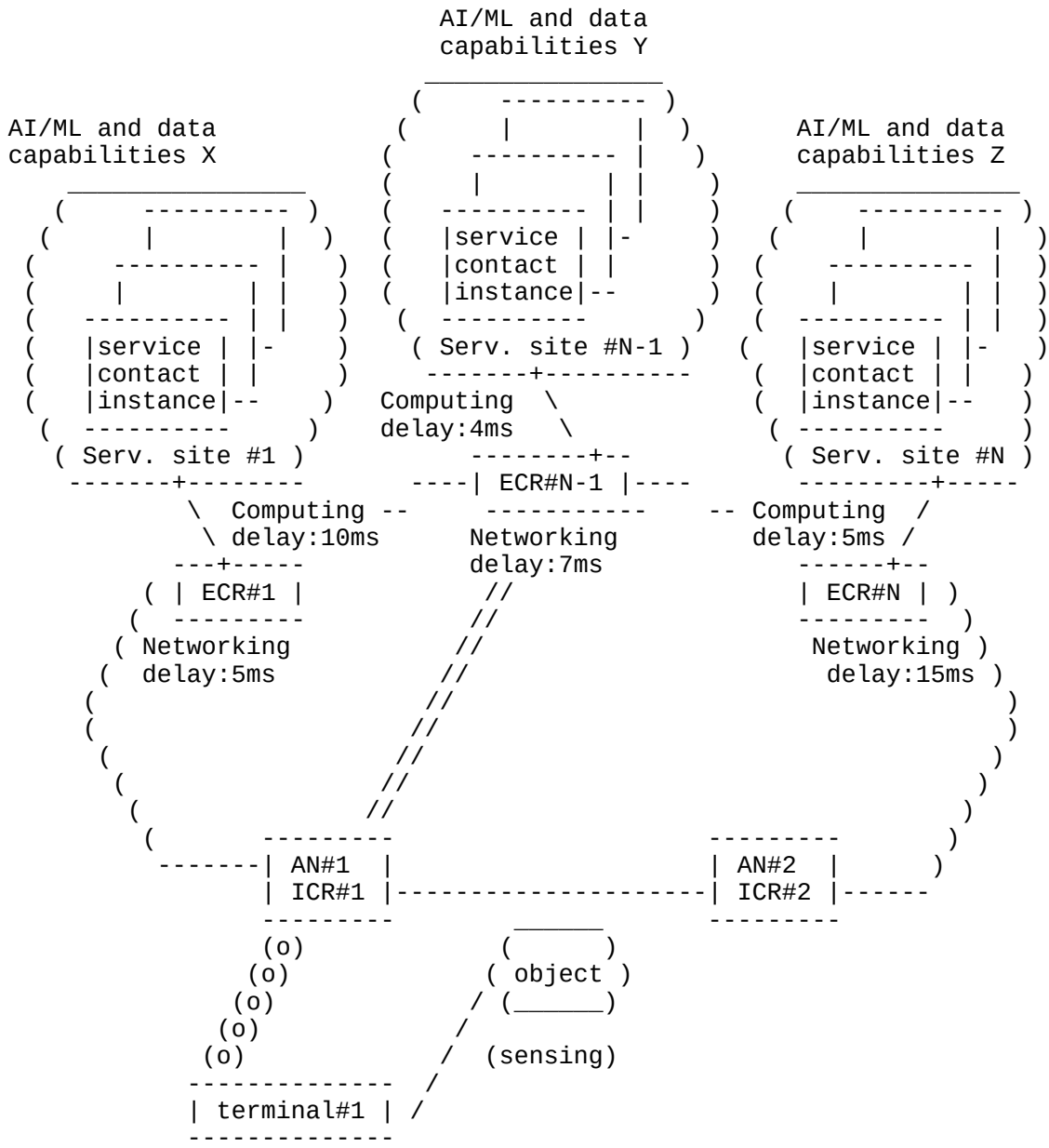

Figure 1: Exemplary scenario

1.2. Problem statement

The main problem that this document tries to address is the following: current networking systems mainly take into consideration connectivity characteristics when deciding how to route traffic. While some recent mechanisms start to consider jointly compute and networking, there are no solutions that account at the same time for AI/ML and trained models' availability and data governance policies.

Based on the former, this document proposes solutions to enable the network to select the best site to instantiate a sensing processing service, augmenting CATS enabled solutions that consider both connectivity and computing, to also consider AI/ML and data capabilities and governance policies. In particular, this document addresses the following question: what information does the network need to select most suitable AI/ML-enabled sensing service to be instantiated?, leveraging the architecture defined in [I-D.bernardos-cats-ip-address-anchoring]?

2. Terminology

The following terms used in this document are defined by the IETF:

ECR: Egress CATS router. This refers to the Egress CATS-Forwarder as defined in [I-D.ietf-cats-framework].

ICR: Ingress CATS router. This refers to the Ingress CATS-Forwarder as defined in [I-D.ietf-cats-framework].

The following terms ara used in this document:

(Distributed) Sensing Group: a group of devices participating on a sensing task.

Sensing Traffic: traffic used (after some processing) to generate a sensing result.

Sensing Processing Function: a function processing sensing traffic (potentially from different sources) to generate a sensing result (or something that can be further processed to generate a sensing result).

Sensing Signal: radio signal used in the processing.

Sensor Function: function running on a device participating on a sensing task that generates and/or processes a sensing signal.

ISF: integrated sensing function. This is a logical in charge of controlling the distributed sensing task.

CATS Agent: logical entity performing a function related to computing aware traffic steering.

Note that we use UE or terminal to refer to a mobile host.

3. Enabling AI/ML-aware CATS with IP address anchoring

We describe next an example of operation and signaling for the network to be able to select the best site to instantiate a sensing service consumed by a terminal, so traffic can be steered, not only simultaneously meeting connectivity and computing requirements, but also considering AI/ML capabilities and availability of suited data/ trained models and data governance policies. A CATS agent runs on both the ingress (the router to which the terminal is attached to) and egress (a router close to or at the site where the service instance is running) routers, and also at the sites capable of instating services. A CATS agent functionality can also run on the terminal to aid the network deciding or actively influence its site selection. A CATS agent might also run on logical controller entity which might be hosted at the network infrastructure. In addition to the functionality defined in [I-D.bernardos-cats-ip-address-anchoring], [I-D.bernardos-cats-anchoring-service-mobility] and [I-D.bernardos-cats-anchoring-site-mobility], this documents defines a new functionality:

* AI/ML and data/models: it deals with the procedures required to (i) monitor and track capabilities relevant for AI/ML processing, including hardware enablers, trained models, data available for training and/or participating in a distributed learning, data governance policies, etc; (ii) perform the required signaling to facilitate distributed learning mechanisms.

Figure 2 shows the message sequence chart of the AI/ML-aware IP address service-specific anchoring for CATS which is explained next:

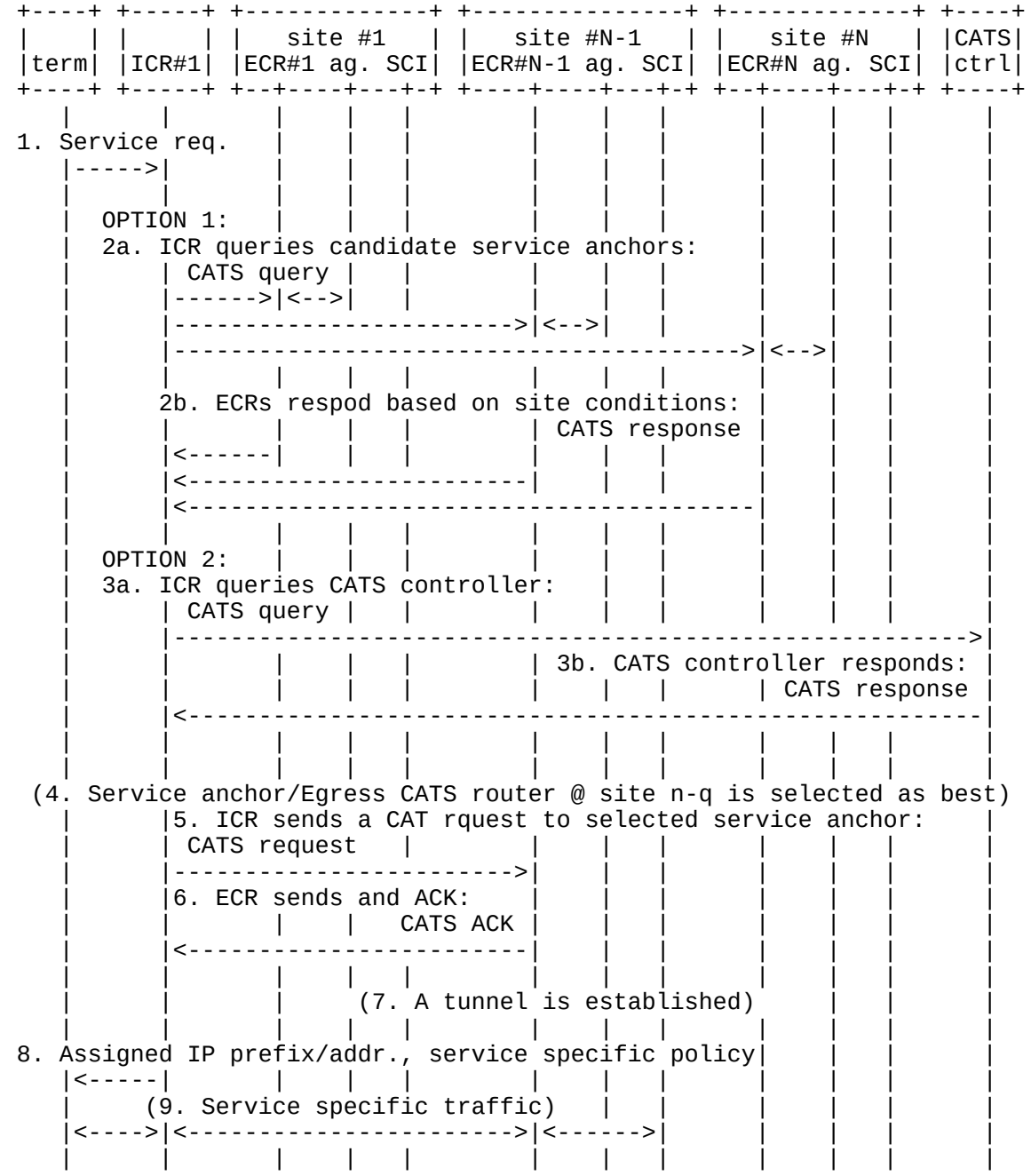

Figure 2: Exemplary signaling

0. A terminal wants to execute a sensing service on a capable node in the network infrastructure. This service has not only connectivity and computing requirements, but also requires some AI/ML processing to provide accurate and timely results. We refer to the connectivity and computing requirements as CATS requirements, and to the AI/ML processing and trained models/ data availability and associated data governance policies as AI/ ML requirements.

1. The terminal sends a Sensing service request to the ICR, including a service ID and, optionally, if the terminal is CATS aware, a list of CATS requirements and AI/ML requirements. Note that this request might be addressed to an ICR or just intercepted by an ICR. If present, the list of AI/ML requirements might include information such as (not limited to any particular combination of parameters):

   * Processing related requirements, such as, but not limited to:

     i. Target hardware required resources (e.g., GPUs).

     ii. Target AI/ML processing with provided data.

     iii. Size of the data to be processed.

     iv. Does the sensing processing function need to be trusted or not by the network?

     v. Requirements on available specific trained models to be used.

     vi. Target on AI explainability for the used models.

     vii. Requirements for available data for potential distributed training on demand.

   * Data governance related requirements, such as, but not limited to:

     i. Security of the data (how and where the data can be stored and isolated, etc.).

     ii. Privacy of the data (which entities may gain access to the data and in what form e.g. partial, full, anonymized).

iii. Trustworthiness of the data (how to make use of the data and which entities may be using it and for what purpose).

iv. Quality of the data (e.g., the quality of the data used to train the model used for the processing).

Additionally, and optionally, the terminal might also include some sensing context information that can help decide what AI/ML processing to use, or be used, as a parameter to the model, such as:

* Specification and/or characteristics of the radio hardware used for sensing (e.g., chipset, antennae, configuration, etc.). This might be used to select the most appropriate model (for example if there is one trained already for this hardware configuration).

* Monitored characteristics of the wireless media used for sensing (such as received power, interference, etc.). This might be used to select the most appropriate model (for example if there is one trained already for these specific conditions).

* Information about physical environment, such as presence of reference objects, that can be used as an additional parameter of the sensing processing function to provide more accurate results. Another example would be positioning coordinates.

* Metadata characterizing desirable data for potential distributed training of a model to be used for the processing of the sensing data.

There are two main options considered:

2. OPTION 1:

a. The ICR sends a query to all ECRs of the domain, or a subset selected based on the location of the ICR. This query may include the following parameters:

i. Service ID: an identifier of the service requested by the terminal. This allows to check if the service can be instantiated, or it is already instantiated.

ii. Terminal ID: an identifier of the terminal requesting the service. This is useful for example for affinity purposes. It might not include information that can be used to identify the user.

iii. ICR ID: identifier of the requesting ICR.

iv. CATS requirements: list of requirements, e.g., connectivity and computing requirements.

v. AI/ML requirements: list of requirements to be useful to filter and rank potential instances.

vi. Sensing context: additional data that is useful in the instance selection, the AI/ML model and configuration as additional parameters for the AI/ML training and/or inference.

b. Each ECR, possibly after checking with the CATS agent of the site(s) it provides connectivity, responds, including the following information:

i. Service ID.

ii. Terminal ID.

iii. ECR ID: identifier of the ECR sending the response.

iv. CATS conditions: how the site meets each of the requirements included in the request.

v. AI/ML capabilities: describing how the site meets each of the requirements included in the request, and information about available data and trained models available at the site.

vi. (Optional): URI to get to the service instance. A CATS agent at a site might be collocated with the ECR. Examples of a CATS agent at a site are network controllers or orchestrators at the site. Note that the way a CATS agent at an ECR may interact with the CATS agent of the site is out of the scope of this document. Examples include using monitoring and telemetry interfaces with an orchestrator managing the site.

Based on the received responses, the ICR selects an ECR. (step 4).

3. OPTION 2:

a. The ICR sends a query to a CATS controller in the domain, including the following parameters:

i. Service ID: an identifier of the service requested by the terminal. This allows to check if the service can be instantiated, or it is already instantiated.

ii. Terminal ID: an identifier of the terminal requesting the service. This is useful for example for affinity purposes. It might not include information that can be used to identify the user.

iii. ICR ID: identifier of the requesting ICR.

iv. CATS requirements: list of requirements, e.g., connectivity and computing requirements.

v. AI/ML requirements: list of requirements to be useful to filter and rank potential instances.

vi. Sensing context: additional data that is useful in the instance selection, the AI/ML model and configuration as additional parameters for the AI/ML training and/or inference.

b. The CATS controller, which has the overall view of all the sites and ECRs of the domain, responds back including the following information:

i. Service ID.

ii. Terminal ID.

iii. ECR ID: identifier of the ECR sending the response.

iv. CATS conditions: how the site meets each of the requirements included in the request.

v. Selected ECR: IP address of the selected ECR.

vi. AI/ML capabilities: describing how the site meets each of the requirements included in the request, and information about available data and trained models available at the site.

4. At this point, there is an ECR (and site) selected for use for the specific service requested by the terminal.

5. The ICR requests the proposed/selected ECR to establish a traffic steering session with it, sending a CATS request. This request includes the same information that was included in the CATS query (to facilitate stateless operation of the ECRs while being queried).

6. The selected ECR, if it accepts the request, responds back with an acknowledgement, including the following information:

   * Service ID.

   * Terminal ID.

   * ECR ID: identifier of the ECR sending the response.

   * CATS conditions: how the site meets each of the requirements included in the request.

   * AI/ML capabilities: describing how the site meets each of the requirements included in the request, and information about available data and trained models available at the site.

   * IP prefix assigned for the terminal to use to reach the service instance.

   * (Optional): URI to get to the service instance.

7. An IP tunnel is established between the ICR and the selected ECR. Forwarding is also setup so traffic going from/to the allocated IP prefix is sent through the tunnel at the ICR/ECR.

8. The ICR conveys the allocated IP prefix to the terminal. This can be done using Router Advertisements, optionally enhanced with RFC 4191 policies for the selected service. Alternatively, other options such as DHCP can be used to provide the prefix.

9. Traffic of the service for this terminal is steered using the IP tunnel.

4. IANA Considerations

TBD.

5. Security Considerations

TBD.

6. Acknowledgments

The work of Carlos J. Bernardos in this document has been partially supported by the Horizon Europe MultiX (Grant 101192521), DISCO6G-CM (TEC-2024/COM-360) and UNICO I+D 6G-DATADRIVEN projects.

7. Informative References

[I-D.bernardos-cats-anchoring-service-mobility]
Bernardos, C. J. and A. Mourad, "Service Mobility-Enabled Computing Aware Traffic Steering using IP address anchoring", Work in Progress, Internet-Draft, draft-bernardos-cats-anchoring-service-mobility-04, 24 September 2025, <https://datatracker.ietf.org/doc/html/draft-bernardos-cats-anchoring-service-mobility-04>.

[I-D.bernardos-cats-anchoring-site-mobility]
Bernardos, C. J. and A. Mourad, "Site Mobility-Enabled Computing Aware Traffic Steering using IP address anchoring", Work in Progress, Internet-Draft, draft-bernardos-cats-anchoring-site-mobility-00, 19 October 2025, <https://datatracker.ietf.org/doc/html/draft-bernardos-cats-anchoring-site-mobility-00>.

[I-D.bernardos-cats-ip-address-anchoring]
Bernardos, C. J. and A. Mourad, "Computing Aware Traffic Steering using IP address anchoring", Work in Progress, Internet-Draft, draft-bernardos-cats-ip-address-anchoring-04, 24 September 2025, <https://datatracker.ietf.org/doc/html/draft-bernardos-cats-ip-address-anchoring-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-20, 26 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-20>.

Authors' Addresses

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
28911 Leganes, Madrid
Spain
Phone: +34 91624 6236
Email: cjbc@it.uc3m.es
URI: http://www.it.uc3m.es/cjbc/

Alain Mourad
InterDigital Europe
Email: Alain.Mourad@InterDigital.com
URI: http://www.InterDigital.com/