CATS WG
Internet-Draft
Intended status: Standards Track
Expires: 30 August 2025

CJ. Bernardos
UC3M
A. Mourad
InterDigital

26 February 2025

Computing Aware Traffic Steering using IP address anchoring
draft-bernardos-cats-ip-address-anchoring-03

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 defines new extensions for a terminal connected to a network infrastructure, to request a service with specific connectivity and computing requirements, so traffic is steered to an instance meeting both requirements. Both CATS-aware and -unaware terminals are considered. Exemplary signaling control messages and operation extending the well-known Proxy Mobile IPv6 protocol are also defined.

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 30 August 2025.

Table of Contents

1. Introduction and Problem Statement

1.1. Use case scenario

Let's consider a possible use case scenario, just for the sake of illustrating the scenario. Several nodes (UEs in this example) are acting as sensors in an Integrated Sensing and Communications (ISAC) case. The sensors generate/collect sensing data that needs to be processed timely and appropriately to generate an accurate sensing result. Part of this service is executed in the network infrastructure, posing some requirements on the connectivity (e.g., delay between the terminals and the node where the service is executed on the network infrastructure) and computing resources (e.g., capabilities to render the XR video within a certain latency budget). Within the network domain where the terminals are connected to there are multiple sites capable of hosting the service, each with potentially different connectivity and computing characteristics.

Figure 1 shows an exemplary scenario. Considering the connectivity and computing latencies (just as an example of metrics), the best service site is #n-1 in the example used in the Figure.

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.

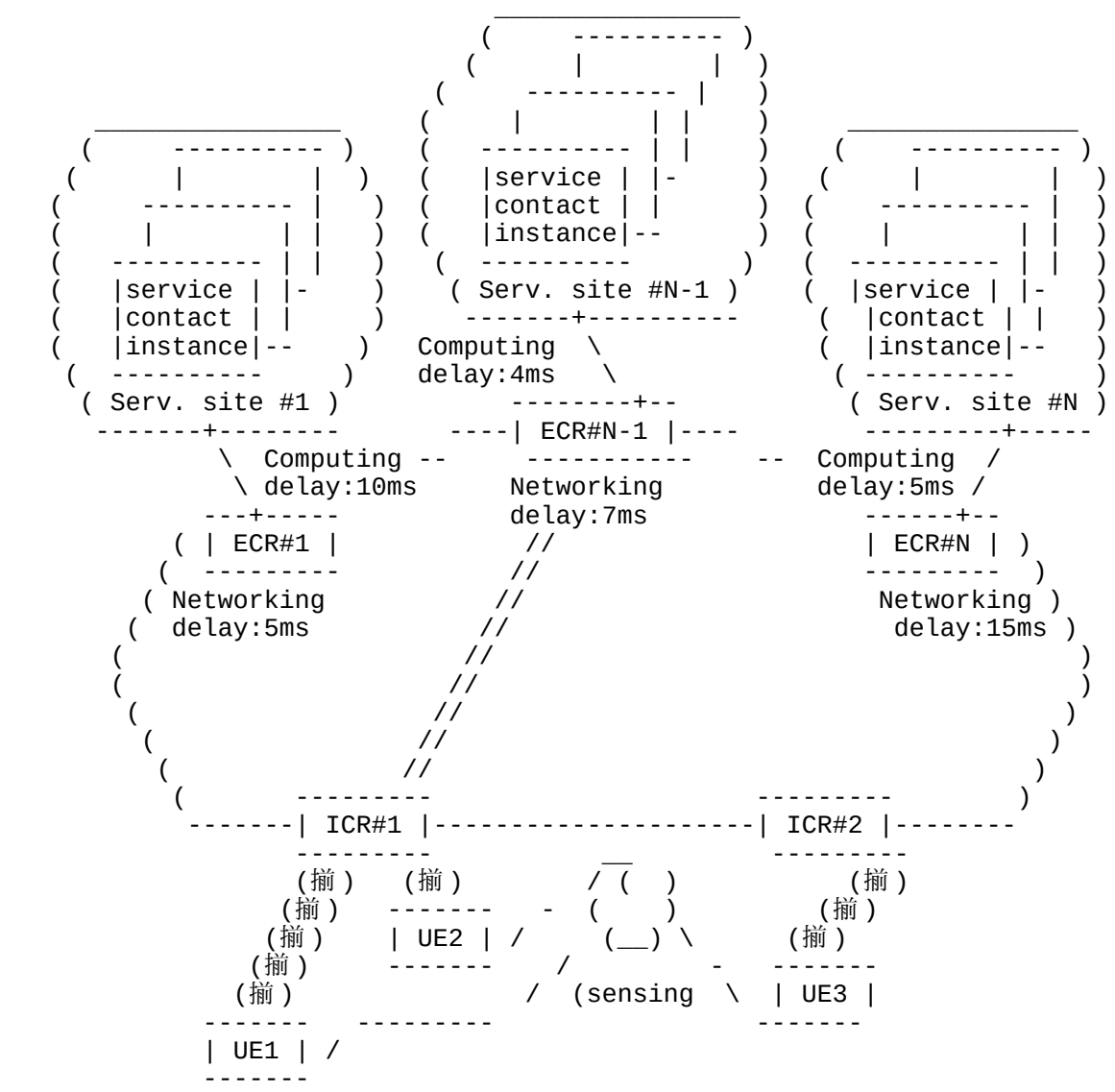


Figure 1: Exemplary scenario

1.2. Problem statement

The main problem that this document tries to address is the following. The network does not have mechanisms yet to enable service-specific connectivity and computing-aware traffic steering, which benefit from optimal service instance location selection and traffic steering.

Based on the former, this document proposes solutions to enable the network to select the best site to instantiate a terminal service, taking into account service-specific requirements at both connectivity and computing levels. In particular, this document addresses the following questions: (i) what information does the network need to be able to select the best location for a service to be instantiated?; and, (ii) how to steer traffic between the terminal and the selected service site, in a way that is transparent to the network forwarding infrastructure, and even to the terminal?

2. 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 [RFC2119].

The following terms are used in this document:

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].

3. Enabling IP address service-specific anchoring for CATS

We describe next an example of operation and signaling for the network to be able to select the best site to instantiate a service to be consumed by a terminal, so traffic can be steered simultaneously meeting connectivity and computing requirements. A CATS agent is defined to run 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 instantiating services. Optionally, the CATS agent functionality can also run on the terminal to aid the network deciding or actively influence its site selection. The CATS agent might have the following functionality:

* Instance selection engine: it deals with the procedures required to perform service and terminal specific instance selection. For example, ICRs, ECRs and sites need this functionality so they can select the location of a given service instance. Optionally, a terminal might also run this engine, to actively participate in the selection process.

* Traffic steering engine: it deals with the ICR and ECR selection and the associated traffic steering between them, in order to meet the connectivity and computing requirements of the service. This functionality might be present at CATS agents running at ICRs and ECRs.

In the following we describe an extended terminal service request procedure enabling the network infrastructure to select a service instance meeting the connectivity and computing requirements of the service, and the setup of the required traffic steering for the service traffic. Extensions and new behavior are highlighted. Note that variations are possible over this exemplary signaling diagram.

Figure 2: Exemplary signaling

A terminal wants to execute a service/app which is requires some functionality to be run on the network infrastructure (e.g., an AR/VR/XR service). This service has specific requirements in terms of both connectivity and computing. We refer to them as CATS requirements.

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

    a. Target bounded latency.

    b. Target minimum bandwidth.

    c. Target computing latency (type of operation, offered load).

    d. Target required computing resources (e.g., hardware specific features).

    e. Affinity constraints (e.g., "not to execute where function Y is already running").

    f. Etc.

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.

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. (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.

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

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

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.

   * 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 [RFC4191] policies for the selected service. Alternatively, other options such as DHCP can be used to provide the prefix.

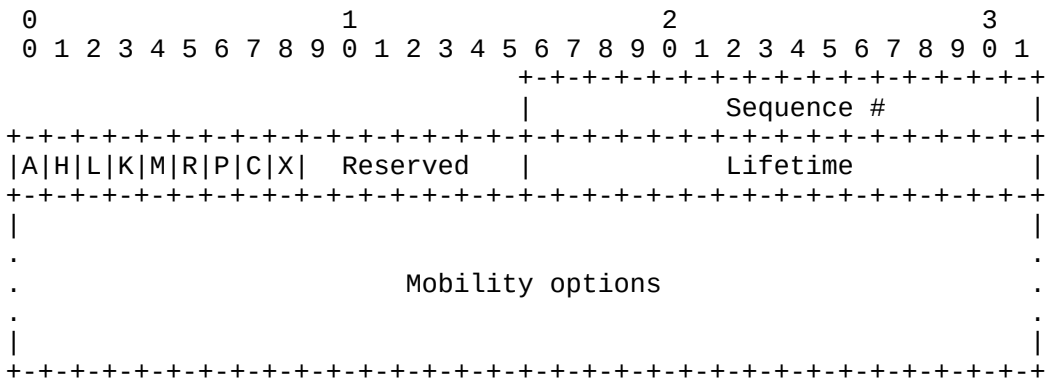
9. Traffic of the service for this terminal is steered using the IP tunnel.

4. Proxy Mobile IPv6 signaling extensions to enable IP address service-specific anchoring for CATS

The control plane extensions introduced in the previous section can be implemented over different protocols. This section specifies extensions to Proxy Mobile IPv6.

4.1. CATS query/respond/request/ACK: CATS PBU/PBA

The CATS query message and request can be implemented as an extended Proxy Binding Update (PBU) message (defined in RFC 5213 [RFC5213]):



A CATS query can be sent by an ICR to an ECR, and also by an ICR to a CATS controller. A CATS request can be sent by an ICR to an ECR.

Message fields:

* Sequence #: Same as defined in RFC 6275 [RFC6275].

* Flags: as defined in RFC 5213, 6275 and IANA registries for the mobility flags. A new flag 'C' is defined to identify a CATS query. A new flag 'X' is defined to identify a CATS request. Note that the location of the 'C' and 'X' flags might be different from the ones shown in the figure above.

* Lifetime: Same as defined in 6275. Basically, it indicates the number of time units remaining before the association between the ICR and the ECR (including the associated IP prefix) MUST be considered expired.

* Mobility options: This field contains one or more mobility options, whose encoding and formats are defined in RFC 6275. In order to uniquely identify the target terminal, the terminal identifier MUST be contained in the Mobile Node Identifier option. This option is used to carry the terminal ID parameter described in this document.

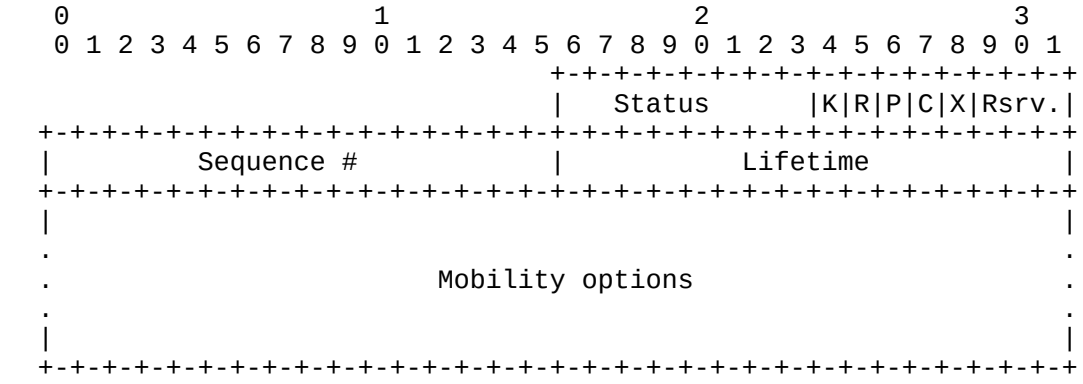
The following new options can be used in this message:

* CR_ID.

* Service_ID.

* CATS requirements.

A CATS response / CATS ACK can be implemented as an extended Proxy Binding Acknowledgement (PBA) message (defined in RFC 5213):



A CATS response can be sent by an ECR to an ICR, and also by a CATS controller to an ICR. A CATS ACK can be sent by an ECR to an ICR, and also by a CATS controller to an ICR.

Message fields:

* Status: same as defined in RFC 6275, with new status codes defined to report: "Success, CATS sites available" and "Error, no CATS sites available".

* Flags: as defined in RFC 5213, 6275 and IANA registries for the mobility flags. A new flag ’C’ is defined to identify a CATS response. A new flag ’X’ is defined to identify a CATS ACK. Note that the location of the ’C’ and ’X’ flags might be different from the ones shown in the figure above.

* Sequence #: Same as defined in RFC 6275.

* Lifetime: Same as defined in 6275. Basically, it indicates the number of time units remaining before the association between the ICR and the ECR (including the associated IP prefix) MUST be considered expired.

* Mobility options: This field contains one or more mobility options, whose encoding and formats are defined in RFC 6275.

The following new options can be used in this message:

* CR_ID.

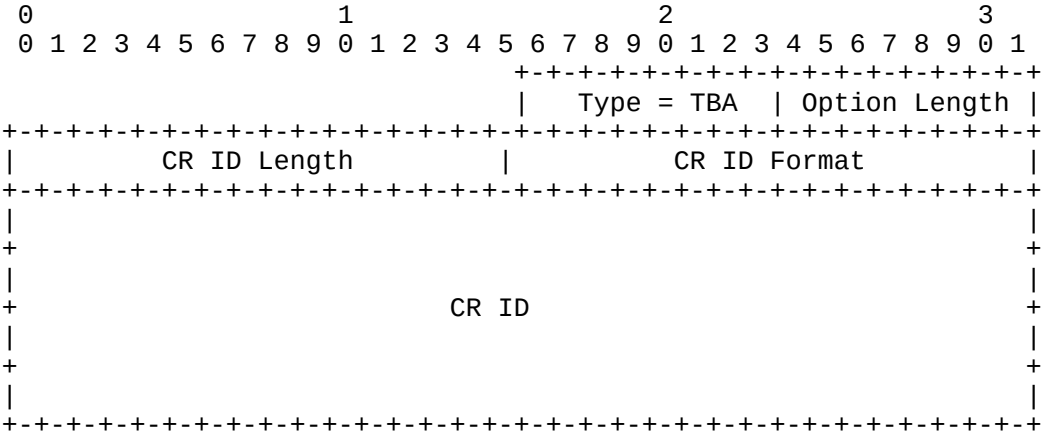
* Service_ID.

* CATS conditions.

* Home Network Prefix option (as defined in RFC 5213).

4.2. CR_ID mobility option

The CR_ID option has the following format:



* Option Type: TBA by IANA.

* Option Length: 8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields.

* CR ID Length: 8-bit unsigned integer. Length of the CR ID field, in octets.

* CR ID Format: 8-bit unsigned integer. Identifies the format of the CR ID. Possibles values:

  - 0: Reserved.

  - 1: IP address (v4 or v6, determined by CR ID Length).

  - 2: L2 address (48 or 64 bit, determined by CR ID Length).

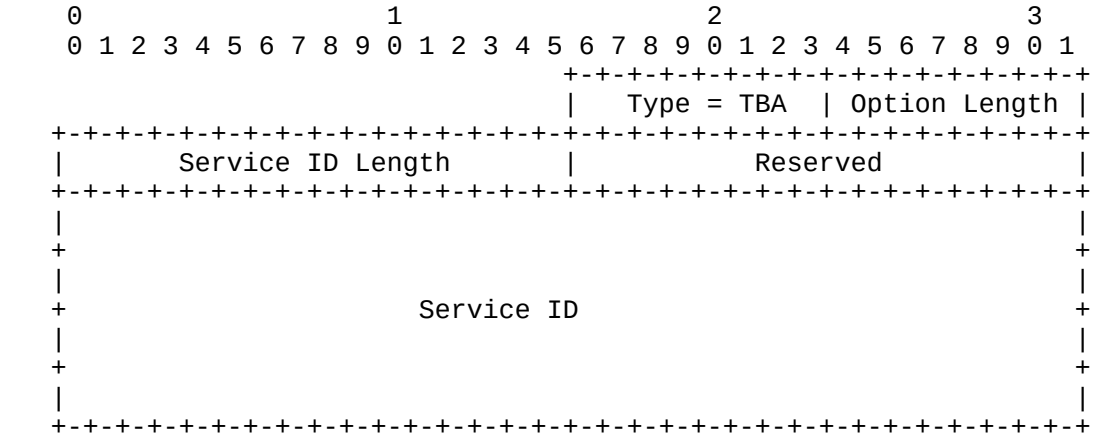
  - 3: URI.

  - 4-255: reserved for future use.

* CR ID: variable length field that identifies the ECR/ICR/selected ECR.

4.3. Service_ID mobility option

The Service_ID option has the following format:



* Option Type: TBA by IANA.

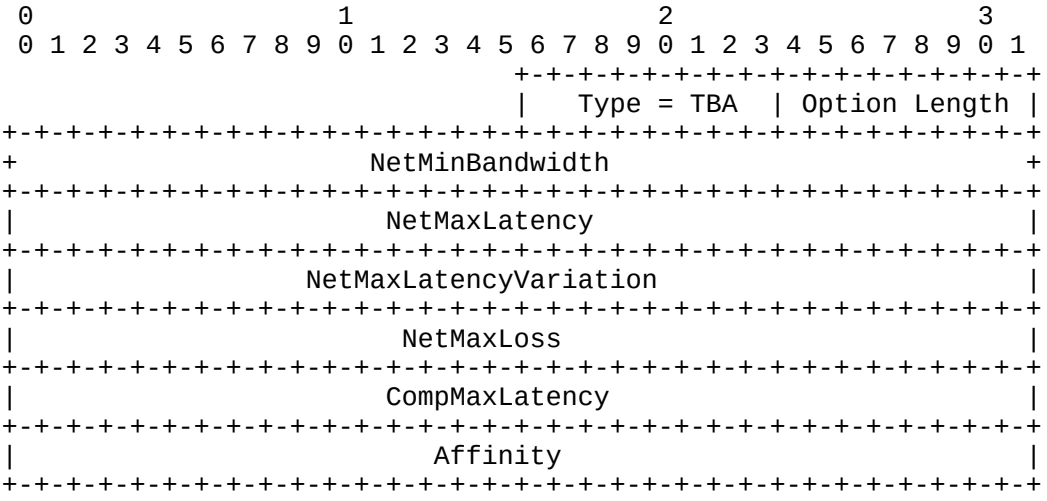
* Option Length: 8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields.

* Service ID Length: 8-bit unsigned integer. Length of the Service ID field, in octets.

* Service ID: variable length field that identifies Service.

4.4. CATS requirements/conditions mobility option

The CATS requirements/conditions option has the following format:



* Option Type: TBA by IANA. A different value is used for the CATS requirements and for the CATS conditions. In the subfields below, the difference between the requirements and the conditions is that for the CATS conditions messages, the values included are what the associated ECR/site can provide, in reference to the target values included in the CATS requirements option.

* Option Length: 8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields.

* NetMinBandwidth: 32-bit unsigned integer. NetMinBandwidth is the minimum network bandwidth that has to be guaranteed for the flow. NetMinBandwidth is specified in octets per second.

* NetMaxLatency: 32-bit unsigned integer. NetMaxLatency is the maximum latency between ICR and service instance for a single packet of the flow. NetMaxLatency is specified as an integer number of nanoseconds.

* NetMaxLatencyVariation: 32-bit unsigned integer. NetMaxLatencyVariation is the difference between the minimum and the maximum end-to-end, one-way latency. NetMaxLatencyVariation is specified as an integer number of nanoseconds.

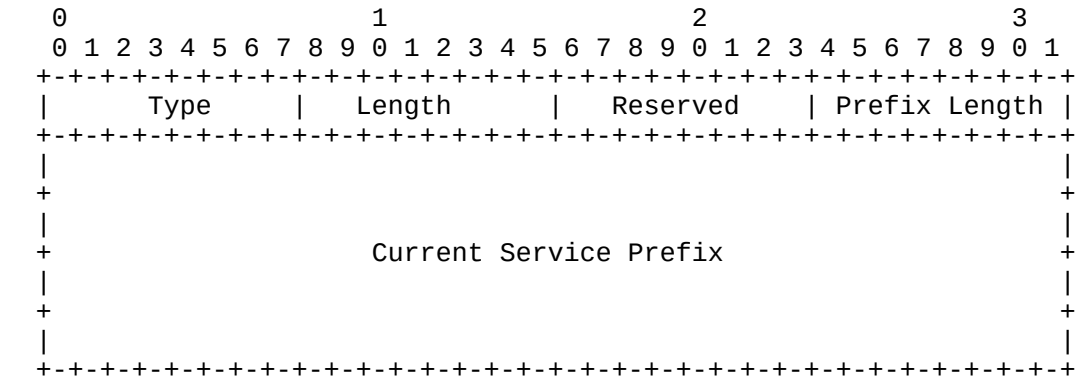
* NetMaxLoss: 32-bit unsigned integer. NetMaxLoss defines the maximum Packet Loss Rate (PLR) requirement for the flow between the ICR and the service instance and the loss measurement interval.

* CompMaxLatency: 32-bit unsigned integer. CompMaxLatency is the maximum latency incurred by the service instance for a single packet of the flow. CompMaxLatency is specified as an integer number of nanoseconds.

* Affinity: Variable length field used to indicate affinity requirements. Different formats/types of affinity may be used.

4.5. Service prefix mobility option

The Service prefix option has the following format:



* Option Type: TBA by IANA.

* Length: 8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields. This field MUST be set to 18.

* Reserved: This 8-bit field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

* Prefix Length: 8-bit unsigned integer indicating the prefix length of the IPv6 prefix contained in the option.

* Service Prefix: A sixteen-byte field containing the IPv6 prefix used by service for the specific terminal.

5. IANA Considerations

TBD.

6. Security Considerations

TBD.

7. Acknowledgments

The work of Carlos J. Bernardos in this document has been partially supported by the Horizon Europe PREDICT-6G (Grant 101095890), DESIRE6G (Grant 101096466) and UNICO I+D 6G-DATADRIVEN project.

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/info/rfc2119>.

8.2. 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-05, 10 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cats-framework-05>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <https://www.rfc-editor.org/info/rfc5213>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <https://www.rfc-editor.org/info/rfc6275>.

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/