Distributed Mobility Management
Internet-Draft
Intended status: Informational
Expires: 2 September 2026

K. Nguyen-Trung
N. Tran
Y. Kim
Soongsil University

1 March 2026

Service Segmentation Considerations for CATS-MUP
draft-dcn-dmm-cats-mup-service-segmentation-00

Abstract

Service segmentation introduces an emerging deployment paradigm in which a service is composed of multiple distributed subtasks forming a service pipeline. This document discusses architectural considerations for a MUP Sequence Session Transform to support ordered traversal across multiple subtask instances and to maintain service continuity during pipeline updates, particularly when stateful subtasks are involved.

Table of Contents

1. Introduction 2
2. Terminology 3
3. Architecture Overview 3
4. MUP Sequence Session Transform Mechanism 5
5. Service Continuity for Stateful Service Segmentation 7
6. IANA Considerations 9
7. Security Considerations 9
8. References 9
8.1. Normative References 9
8.2. Informative References 9
Authors' Addresses 10

1. Introduction

The document [I-D.draft-dcn-dmm-cats-mup] describes how to integrate Computing-Aware Traffic Steering (CATS) capabilities into the Mobile User Plane (MUP) architecture. In that design, when multiple candidate service instances are deployed at different locations, the MUP Controller (MUP-C), as the core component of the architecture, selects the optimal service instance by considering computing and network information. The MUP-C receives user mobility session information from the control plane entity and converts such session information into IPv6 dataplane routing information. Instead of relying on traditional anchoring mechanisms or intermediate user plane forwarding nodes, the session is directly mapped into SRv6-based routing instructions in the underlay network. The Type 2 Session Transformed Route (T2ST) and Type 1 Session Transformed Route (T1ST) are used to convert session information into SRv6-based routing paths toward the selected service endpoint. As a result, traffic steering is realized at the IP routing level without requiring dedicated anchor or intermediate nodes in the mobile user plane.

However, emerging 6G applications introduce a new deployment paradigm, where a service is decomposed into multiple subtasks that are distributed across different edge locations. These subtasks can be organized as a sequential service pipeline, where traffic must

traverse service instances in a predefined sequence, or as a parallel service pipeline, where traffic is distributed to multiple service instances and later merged. Such service segmentation scenarios are discussed in [I-D.draft-dcn-cats-req-service-segmentation].

The existing T1ST and T2ST mechanisms are defined for mapping a session to a selected service instance, where traffic is steered toward a specific execution endpoint. While this model is suitable when a session is directed to one optimal service instance, service segmentation introduces scenarios in which a sequential service pipeline spans multiple service instances as part of a single service execution. In such cases, the session-to-route transformation must not only select appropriate service instances, but also support ordered traversal across them and preserve overall service continuity, particularly when the pipeline includes stateful subtasks.

The existing CATS-MUP architecture focuses on mapping a session to a selected service instance. This document discusses scenarios where a service execution consists of multiple distributed subtasks forming a pipeline. It discusses extensions to session transformation behavior to enable ordered traversal across multiple subtask instances and to support service continuity during pipeline updates, particularly when stateful subtasks are involved.

2. Terminology

This document uses the terminology defined in [I-D.draft-dcn-dmm-cats-mup] and [I-D.draft-dcn-cats-req-service-segmentation].

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].

3. Architecture Overview

This document does not define mechanisms for constructing or orchestrating service pipelines. Pipeline composition, subtask placement, and lifecycle management are considered out of scope. Furthermore, it does not modify or redefine existing SRv6 behavior definitions and relies on procedures specified in existing SRv6 and SFC specifications.

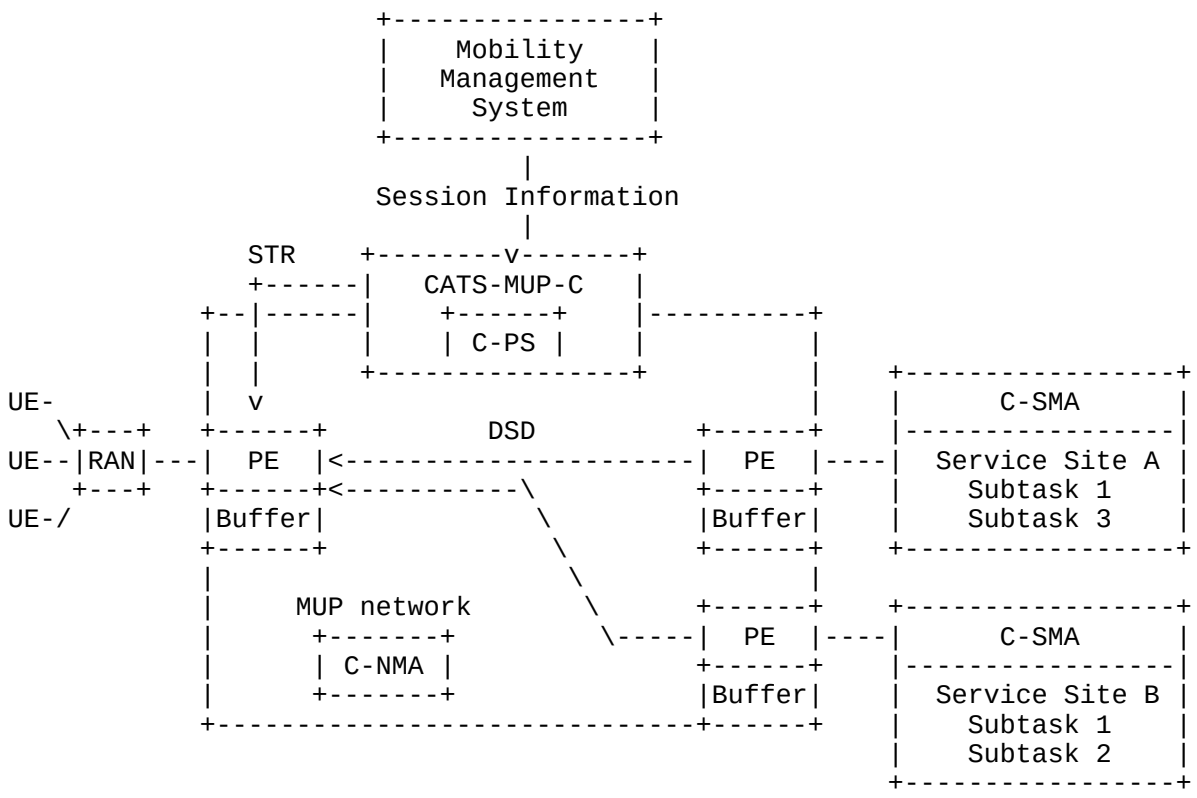

Figure 1: Service Segmentation Extensions for CATS-MUP Architecture

Figure 1 illustrates the high-level architecture for supporting service segmentation in the CATS-MUP environment, where an application is decomposed into multiple subtasks deployed across different edge sites. Depending on deployment and execution logic, these subtasks may form sequential chains or parallel branches, requiring traffic steering to evolve from single-instance selection toward pipeline-aware steering across distributed subtasks.

For sequential pipelines, existing Session Transform Routes, including T1ST and T2ST, require the C-PS to configure traffic steering independently for each MUP-PE along the execution chain. As a result, the completion time of the CATS-MUP-C operation increases proportionally with the number of subtasks composing the pipeline. The situation becomes more complex when the active pipeline must be updated or replaced due to mobility events or changing resource conditions, since multiple steering configurations must be recomputed and consistently applied across all participating nodes. To address

this limitation, this document proposes a new Session Transform Route referred to as the _MUP Sequence Session Transform_, enabling pipeline-level traffic steering for sequential service execution. The detailed design is presented in Section 4.

For parallel pipelines, existing T1ST and T2ST mechanisms remain applicable because traffic distribution and result aggregation can be handled by split and merge functions. User requests may be distributed across parallel subtasks, and intermediate results are combined before delivering the final outcome to the UE. However, each parallel branch may itself consist of a sequential chain of subtasks, in which case the same scalability and steering reconfiguration issues as observed in sequential pipelines arise. Therefore, the proposed _MUP Sequence Session Transform_ can also be applied within such branches to enable pipeline-level traffic steering.

Service segmentation may involve stateful subtasks, making subtask migration necessary when an existing pipeline no longer satisfies QoS requirements. Maintaining service continuity during pipeline transitions requires preventing packet loss while new subtask instances are activated and traffic steering is updated. One approach to enabling such transitions is the use of SRv6-based buffering, which temporarily stores packets during path updates and releases them once the new execution pipeline becomes operational.

Furthermore, a single subtask type may have multiple instances deployed across different MEC sites. Selecting an optimal pipeline based solely on per-instance evaluation may introduce significant decision overhead as deployment scale increases. To address this challenge, Service Pipeline Metrics are introduced to evaluate pipelines as unified entities rather than independent instances, enabling efficient pipeline selection as described in [I-D.draft-dcn-cats-req-service-segmentation].

4. MUP Sequence Session Transform Mechanism

The MUP Sequence Session Transform defines how the Mobile User Plane steers packets across an ordered set of service stages using a single session transformation operation. Under this mechanism, the execution sequence of a service pipeline is encoded directly into packet forwarding behavior at MUP-PE nodes.

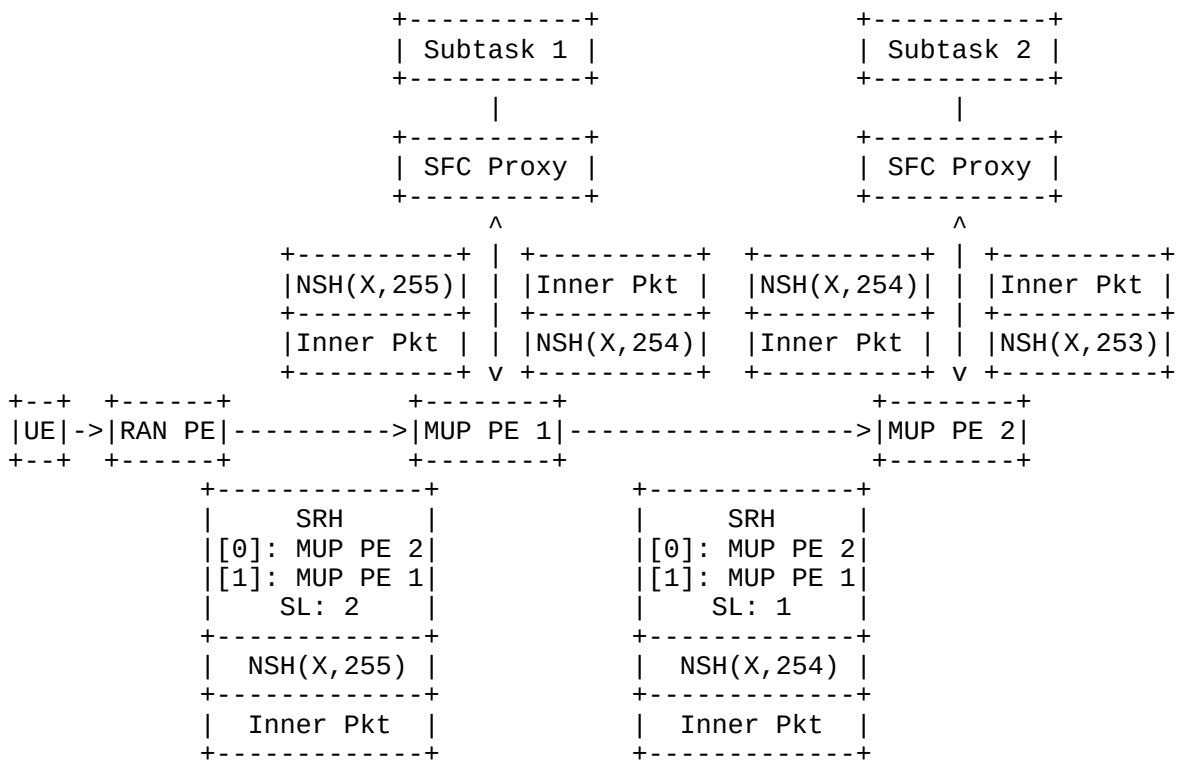

Figure 2: MUP Sequence Session Transform Example

The execution order of a service pipeline is encoded directly into packet forwarding state. Specifically, the SRv6 segment list carried in the SRH defines the traversal sequence across MUP-PE nodes, while service processing context is maintained within the packet using SRv6 Service Function Chaining (SFC) mechanisms, as specified in [RFC9491]. As illustrated in Figure 2, packets advance along the pipeline as each MUP-PE processes its local segment and updates the associated service context.

* The MUP Sequence Session Transform is installed by the CATS-MUP-C at the ingress MUP-PE located at the RAN side. For UE traffic matching the corresponding session policy, the ingress MUP-PE decapsulates the GTP header and re-encapsulates the packet using SRv6 Service Function Chaining (SFC).

* When traffic is steered according to the SRv6 segment list, each MUP-PE performs the following processing steps:

- The MUP-PE processes its local SRv6 SID and removes the outer SRv6 encapsulation, leaving the NSH and inner packet intact.

- The packet is delivered toward the corresponding subtask instance. Depending on deployment, the MUP-PE MAY apply a site-specific encapsulation (such as VXLAN-gre, GRE [ieee-sfc-over-srv6]) before forwarding the packet to the service instance.

- If the subtask instance does not support NSH processing, an SFC proxy MAY be used to terminate or translate the NSH on behalf of the service function.

- After processing, the packet is returned to the MUP-PE together with updated service context.

- The MUP-PE re-applies the SRv6 encapsulation, updates SRH processing state, and forwards the packet toward the next segment indicated in the remaining SRv6 segment list without additional control-plane interaction.

* The NSH context header may include service pipeline identifiers, UE-related attributes, or intermediate processing state, enabling distributed service functions to coordinate execution and maintain continuity across the pipeline.

By embedding pipeline progression into dataplane processing, the MUP Sequence Session Transform enables ordered service execution across multiple edge locations.

5. Service Continuity for Stateful Service Segmentation

In practical deployments, service segmentation often involves stateful service instances, where execution state must be preserved across service instance relocation or pipeline reconfiguration. In such scenarios, immediate redirection of traffic to a newly selected service pipeline may result in service disruption if the target instances are not yet ready to process incoming requests or if state migration is still in progress.

For example, consider a sequential service pipeline composed of five subtasks executed in order. The first four subtasks are stateless, while the fifth subtask is a stateful application. When the stateful subtask is migrated due to mobility events or computing resource degradation, packets processed by the upstream stateless subtasks (i.e., Subtasks 1 through 4) may reach at the stateful subtask before migration is completed. As a result, these packets may be dropped or fail to be processed correctly, leading to wasted computing resources and unnecessary consumption of network bandwidth.

To preserve service continuity during pipeline transition, this document considers new use behaviors [future-SRv6-service-continuity], such as _End.M.GTP6.D.Buffer_, an extension of the End.M.GTP6.D function, integrated into all MUP Session Transform Routes. These behaviors enable traffic to be temporarily held while stateful service instances are being migrated.

* The MUP-C installs MUP Session Transform Routes with buffer [ieee-access-service-continuity-stateful-service-segmentation] at the relevant MUP-PEs, inserting a buffer SID as the first segment in the SRv6 segment list. As a result, packets matching the session policy are first steered toward the buffer. The buffer MAY be realized as a dedicated virtual network function (VNF).

* Furthermore, the MUP-C updates the MUP-PE of all subtasks (current pipeline) so that packets returning from upstream subtask are re-encapsulated using _SR Policy headend behaviors_ with a new SRH. The new segment list places the buffer SID as the first segment, followed by the MUP-PE of the next subtask in the newly selected service pipeline.

* Once a new optimal service pipeline is determined, incoming traffic is redirected toward buffer, allowing migration of multiple stateful subtasks to be performed *in parallel* while maintaining service continuity. Parallel migration reduces service disruption time and increases efficiency.

* After migration is completed, the MUP-C triggers buffer release procedures. Buffered packets are flushed toward the new pipeline, and the buffer removes its SID from the SRH.

* Following buffer release, MUP-PE are updated so that subsequent packets bypass the buffer and are steered directly along the new service pipeline.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

TBD

8. References

8.1. Normative References

[I-D.draft-dcn-dmm-cats-mup]
Tran, N., "Computing Aware Traffic Steering Consideration for Mobile User Plane Architecture", 2026, <https://datatracker.ietf.org/doc/draft-dcn-dmm-cats-mup/>.

[I-D.draft-dcn-cats-req-service-segmentation]
Tran, N., "Additional CATS Requirements Consideration for Service Segmentation-related Use Cases", 2026, <https://datatracker.ietf.org/doc/draft-dcn-cats-req-service-segmentation/>.

[RFC9491] Guichard, J. and J. Tantsura, "Integration of the Network Service Header (NSH) and Segment Routing for Service Function Chaining (SFC)", November 2023, <https://datatracker.ietf.org/doc/rfc9491/>.

8.2. Informative References

[ieee-access-service-segmentation]
Tran, M-N., "Design of 5G Architecture Enhancements for Supporting Edge Split Computing Service Pipeline", 2025, <https://doi.org/10.1109/ACCESS.2025.3630182>.

[ieee-access-service-continuity-stateful-service-segmentation]
Nguyen Trung, K., "Enabling Service Continuity for Stateful Service Segmentation in Mobile Edge Computing Toward 6G", 2026, <https://doi.org/10.1109/ACCESS.2026.3661972>.

[ieee-sfc-over-srv6]
Nguyen Trung, K., "A Design and Implementation of Service Function Chaining Over Segment Routing IPv6 Network", 2024, <https://doi.org/10.1109/ICTC62082.2024.10827193>.

[future-SRv6-service-continuity]
Lemmi, J., "SRv6-Based Edge Service Continuity in 5G Mobile Networks", 2023, <https://doi.org/10.3390/fi16040138>.

Authors' Addresses

Kiem Nguyen Trung
Soongsil University
369, Sangdo-ro, Dongjak-gu
Seoul
06978
Republic of Korea
Email: kiemnt@dcn.ssu.ac.kr

Minh-Ngoc Tran
Soongsil University
369, Sangdo-ro, Dongjak-gu
Seoul
06978
Republic of Korea
Email: mipearlska1307@dcn.ssu.ac.kr

Younghan Kim
Soongsil University
369, Sangdo-ro, Dongjak-gu
Seoul
06978
Republic of Korea
Phone: +82 10 2691 0904
Email: younghak@ssu.ac.kr