

RTGWG
Internet-Draft
Intended status: Standards Track
Expires: 17 October 2025

D. Yuan
ZTE Corporation
Y. Zhang
China Unicom
D. Huang
C. Miao
ZTE Corporation
15 April 2025

Hybrid-Function-Chain (HFC) Framework
draft-yuan-rtgwg-hfc-framework-01

Abstract

With the maturity of cloud native application development, applications can be decomposed into finer grained atomic services. On the other hand, as a distributed computing paradigm, fine grained micro-services could be deployed and implemented in a distributive way among edges to make computing, storage and run-time processing capabilities as close to users as possible to provide satisfied QoE. Under the circumstances analyzed, a Hybrid-Function-Chain (HFC) framework is proposed in this draft, aiming to wisely allocate and schedule resources and services in order to provide consistent end-to-end service provisioning.

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 17 October 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Brief Introduction of HFC	2
2. Requirements Language	4
3. Terminology	5
4. Hybrid-Function-Chain (HFC) Framework	5
4.1. Administration Plane	7
4.2. Control Plane	9
4.3. Forwarding Plane	10
5. SRv6-based HFC Implementation	11
6. Security Considerations	15
7. Acknowledgements	15
8. IANA Considerations	15
9. Normative References	15
Authors' Addresses	15

1. Brief Introduction of HFC

In the context of cloud native, applications are often no longer provided in the form of monolithic services, but are decomposed into a series of cloud native services deployed in distributed clusters, with inter-connections and joint provision to the outer side.

Traffic lanes, for instance, have emerged and been commonly used for environmental isolation and traffic control of the entire request chain of services for grayscale publishing scenarios, would be reckoned as a typical example for Hybrid-Function-Chain (HFC). In fact, the creation of traffic lanes is still executed by various existing network API configurations of the cluster. The service routes are always configured in the cluster and identified endpoints under a service name to implement various scheduling strategies and perform load balancing schemes among multiple optional instances.

Edge computing, as a distributed computing paradigm, the core idea is to make computing, storage and service processing as close to the clients and customers as possible to reduce latency, improve response speed and enhance data security. When applications are further deconstructed into atomic services as analyzed previously, service inter-connections MAY not only exist in adjacent clusters deployed in

a same edge, but also happen with network paths connecting remote edge data centers. Thus, incremental requirements would be raised correspondingly. Relevant use cases and requirements are discussed in [I-D.huang-rtgwg-us-standalone-sid].

Correspondingly, this draft proposes a Hybrid Function Chain (HFC) architecture aimed at providing end-to-end and consistent service provisioning capabilities which includes multiple service endpoints and corresponding connected network paths. Compared to conventional schemes and patterns, HFC is granted with multiple features and connotations.

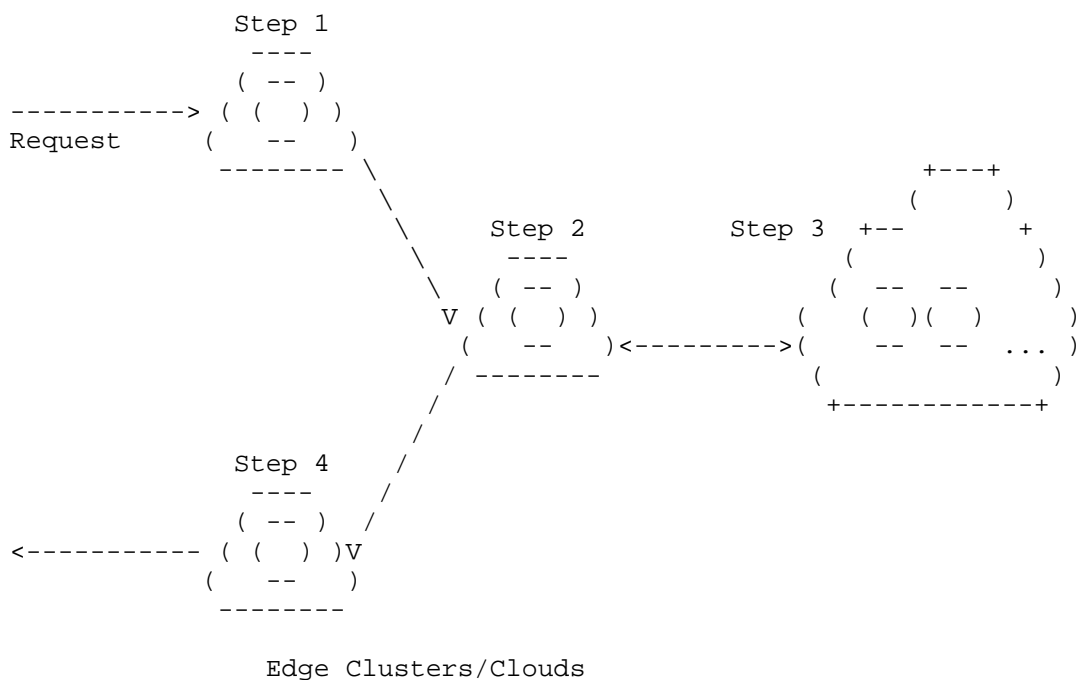


Figure 1: HFC across Multi-edges

- * Hybrid service types and distributed forms: Considering the deployment phase, services and application functions can be deployed in one or multiple clusters in the form of containers, or deployed based on virtual machines. Service instances can be deployed with multiple instances with dynamic implementation and released based on a Serverless framework. Based on the run-time state, microservices and atomic functions form a diverse set of external services, and correspondingly, often raise various requirements for the resources and network capabilities. Compared to conventional Service Function Chain (SFC), the service functions targeted and discussed in HFC contains functions from both underlay network and application (L7) services.
- * Hybrid inter-connections between service instances: The inter-connection, interaction, and collaboration schemes between upstream and downstream functions are always allocated and implemented within various forms. For instance, upstream functions MAY propose unidirectional notifications. Bidirectional requests and responses MAY also be observed between upstream function and single or multiple downstream functions. Multiple inter-connection forms MAY be implemented simultaneously in an overall HFC.
- * Hybrid stacks and techniques: For conventional SFC in which Firewalls and Accelerators MAY be included, service functions are not tended and deployed to terminate data packets. However, for HFC functions, packets and payloads are always terminated at endpoints and payloads would be reorganized and regenerated. The provisioning of HFC process requires collaboration among multiple techniques and extends across TCP/IP stacks.

Based on the concepts of HFC proposed here, this draft further analyzes HFC framework and deconstructs it into several planes with incremental functions and features based on conventional network and service mesh techniques.

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.

3. Terminology

- * HFC: Hybrid Function Chain.
- * SFC: Service Function Chain.
- * QoE: Quality of Experience.
- * SLA: Service Level Agreement.
- * SRv6: Segment Routing over IPv6.
- * OAM: Operation, Administration and Maintenance.
- * APM: Application Performance Management.

4. Hybrid-Function-Chain (HFC) Framework

Hybrid-Function-Chain (HFC) framework generally includes administration plane, control plane and forwarding plane. Based on conventional framework of network and cloud native practices, several incremental functions and features are added and deployed.

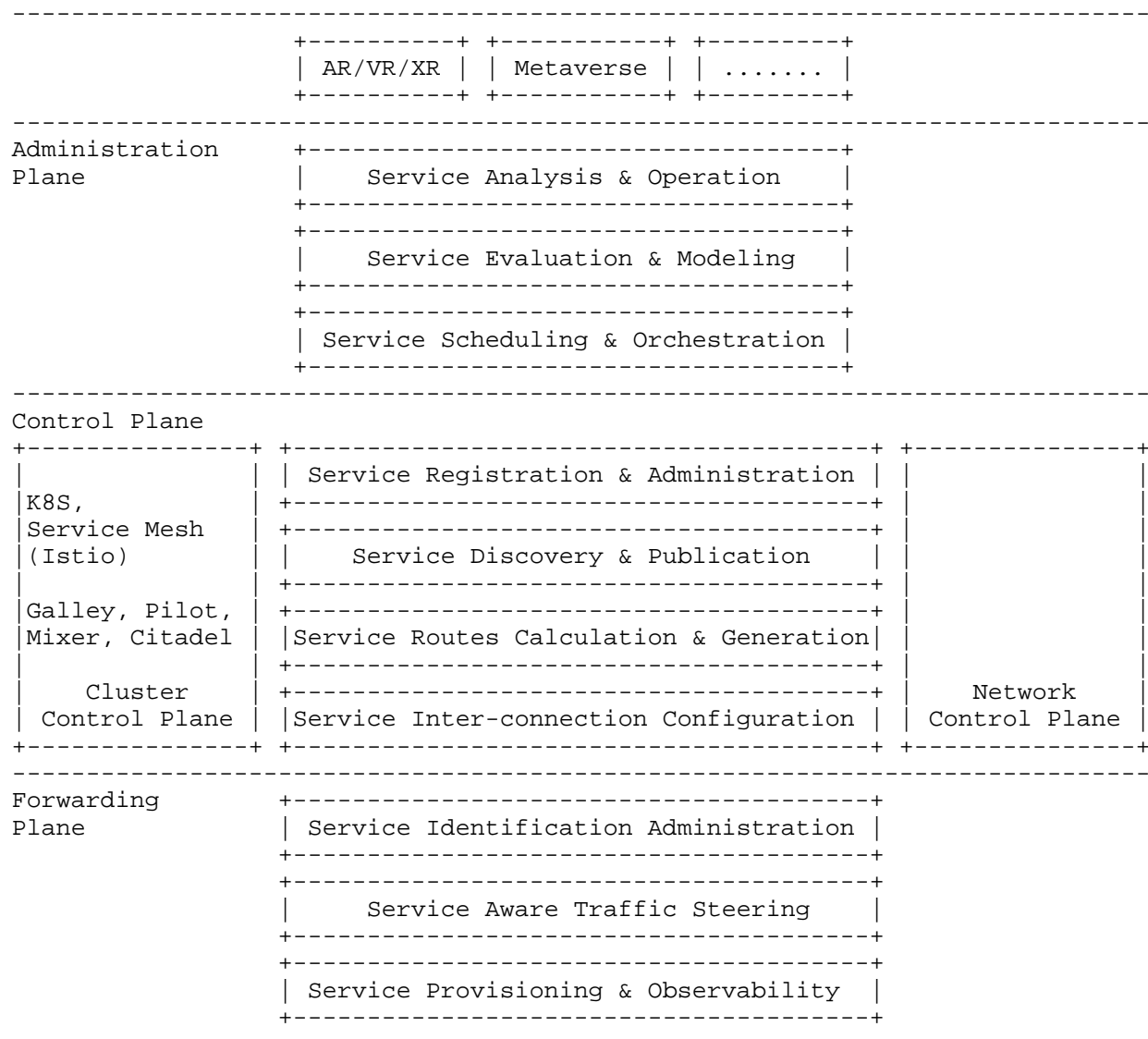


Figure 2: HFC Framework

4.1. Administration Plane

Service Analysis and Operation: The increasingly complex and diverse applications and services display different characteristics and features for outer users. In terms of orchestration for distributed micro-services, Service Analysis and Operation interprets the external and internal forms of overall applications and services as corresponding deconstruction patterns.

- * Deep learning: The overall deep learning process would be decomposed into several successive or related phases and steps, data pre-processing, model training, prediction and estimation, model evaluation based on rewards functions, data storage and API interactions for instance.
- * Live Broadcast: Relevant micro-services MAY include user authentication, live stream administration, live recording, online payment and data migration.

The above deconstructed microservices have serial, parallel, unidirectional, and bidirectional relationships, and their interconnection and collaboration are comprehensively presented as user and outer oriented applications.

Service Evaluation and Modeling: There are different and various resource requirements raised by multiple microservices, including but not limited to:

- * Computing resources and network capabilities: Computing-related services MAY be sensitive to computing resources, related indicators include CPU cores, available memories, floating number calculation. On the other hand, large amount of data transmission MAY be implemented between upstream and downstream services. Thus, the network connecting them would have to reserve sufficient bandwidth and provide abilities of low packet loss rate.
- * Constraints for inter-connection patterns: the inter-connection patterns between upstream and downstream services and functions MAY be classified as unidirectional, bidirectional, one-to-one and one-to-many. Furthermore, due to security concerns for instance, relevant services MAY be deployed at adjacent endpoints or a same edge center geographically.

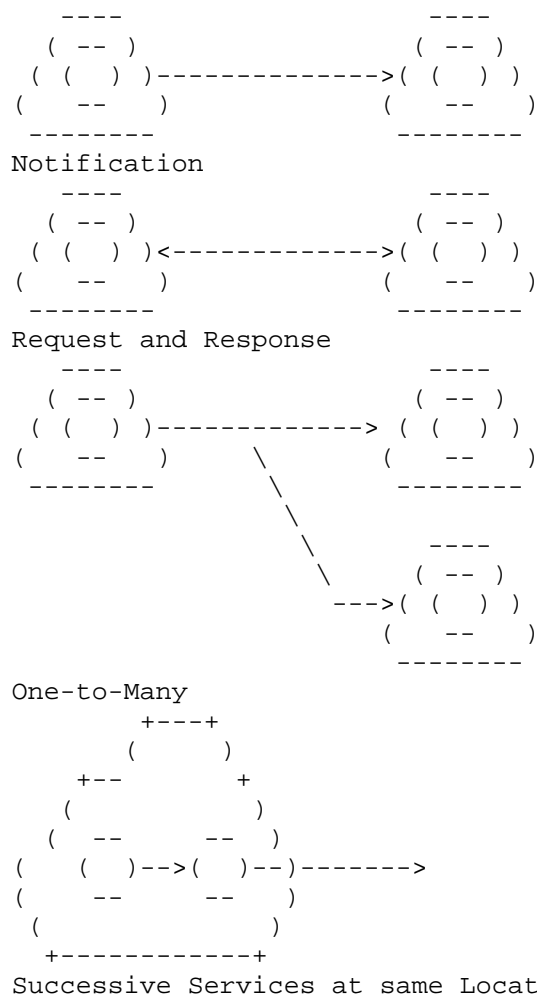


Figure 3: Inter-Connection between Services and Functions

Service Orchestration and Scheduling: Service administration would customize strategies or specific algorithms depending on circumstances of infrastructure and required proposals. Providing low-latency experiences or achieving load balance among available instances and resources SHOULD be selected as specific inclinations for further scheduling.

4.2. Control Plane

Service Registration and Administration: Based on the results and conclusions analyzed by Service Analysis and Operation, the overall service and included micro-services MAY be represented and administered by corresponding identifications. In this draft, Service IDs for micro-services and HFC IDs for HFC processes and services are generally defined. Therefore, HFCs and corresponding micro-services would be displayed and labeled in the control plane. Appropriate identifications would facilitate indicating the service traffic of the workflow.

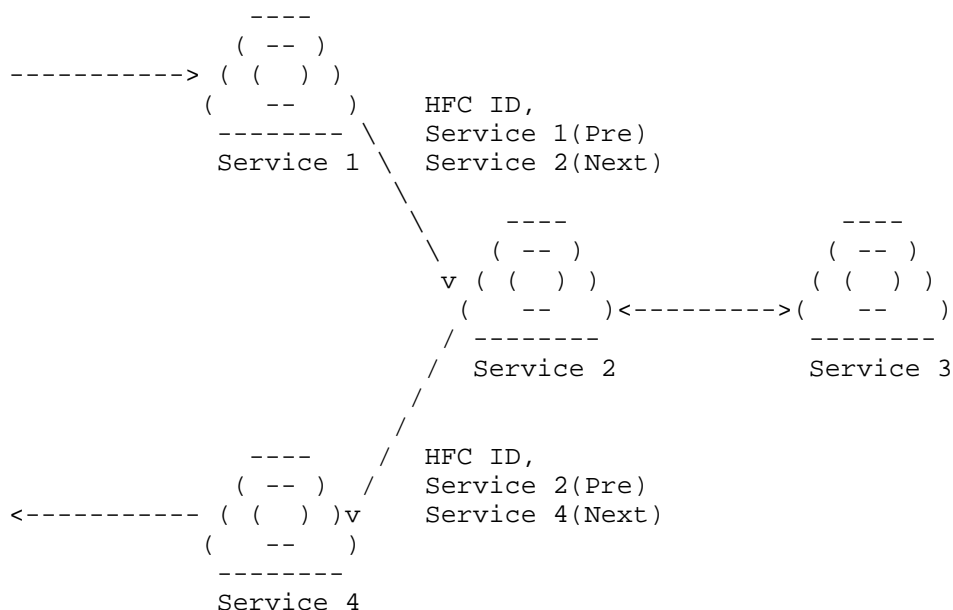


Figure 4: Service Administration by Identification

Service Discovery and Publication: Depending existing and mature control plane protocols and interfaces, distributed services and capabilities of infrastructures SHOULD be able to be collected. Relevant schemes include extended IGP, BGP, BGP-LS, RESTful, Telemetry. The information learned in the control plane MAY include:

- * Computing resources related to services of specific instances.

- * Deployment of service instances or possible and scheduled resources utilization.
- * Network topology and corresponding TE capabilities.

Service Routes Calculation and Generation: Based on the information collected in Service Discovery and Publication, service routes would be calculated to determine appropriate instances and forwarding paths. Service Routes Calculation and Generation SHOULD follow the intentions identified in Service Orchestration and Scheduling. According to Service Registration and Administration, service routes could be distributed and indexed by HFC and service identifications.

Service Inter-connection Configuration: Within conventional schemes for services inter-connection, configurations would be disposed for endpoints distributed in multiple clusters. Istio, for instance, replies APIs including ServiceEntry, VirtualService and DestinationRules to describe inter-connections and relevant principles. Considering the framework of HFC proposed in this draft, service routes would be translated into corresponding configurations issued to clusters for revising API files.

4.3. Forwarding Plane

Service Identification Administration: Traffic would be able to be steered according to identifications distributed from the control plane. Also, the service identifications would inherit and re-generate from the previous ones in the workflow. Proxies, sidecars or gateways SHOULD be able to administer the inheritance and renewal relationship. Suppose an HFC application includes Service ID 1, ID 2 and ID 3, an identifier of {HFC, Service ID 2} implies that the successive function is expected to be Service ID 3. Correspondingly, the identifier would be modified as {HFC, Service ID 3}.

Service Aware Forwarding: Service routes entries would be distributed from the control plane and involved entities and devices would perform traffic forwarding accordingly. Relevant entries include:

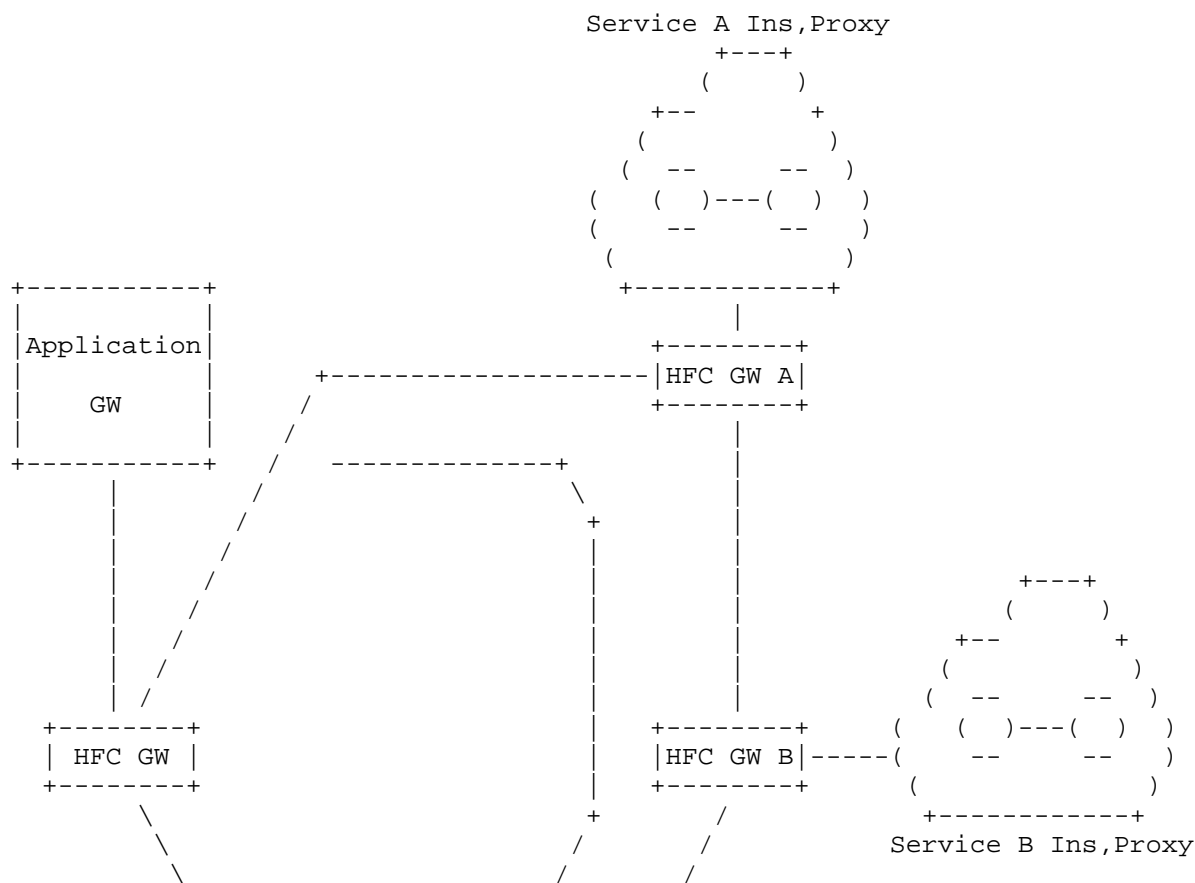
- * Service aware forwarding entries for edges routers in which forwarding paths are indexed by HFC IDs and Service IDs.
- * Service identification administration entries for sidecars, proxies and gateways in which inheritance and correlations would be specified.

Service provisioning and observability: By implementing and performing OAM or APM schemes, forwarding plane would monitor the circumstances and performance of traffic flows. With detections of failures and possible degradations, forwarding plane would be able to support recovering, enhancing and provisioning for traffic flows.

5. SRv6-based HFC Implementation

Based on SRv6, forwarding paths orchestrated for an end-to-end HFC service including specific implementations would be able to be encoded in an SRH, in order to achieve consistent service provisioning across multiple endpoints deployed in distributed clusters and even edge clouds.

The overall paths would be explicitly indicated in the Segment List or be generally displayed. Correspondingly, a strict mode and a loose mode are proposed in this draft.



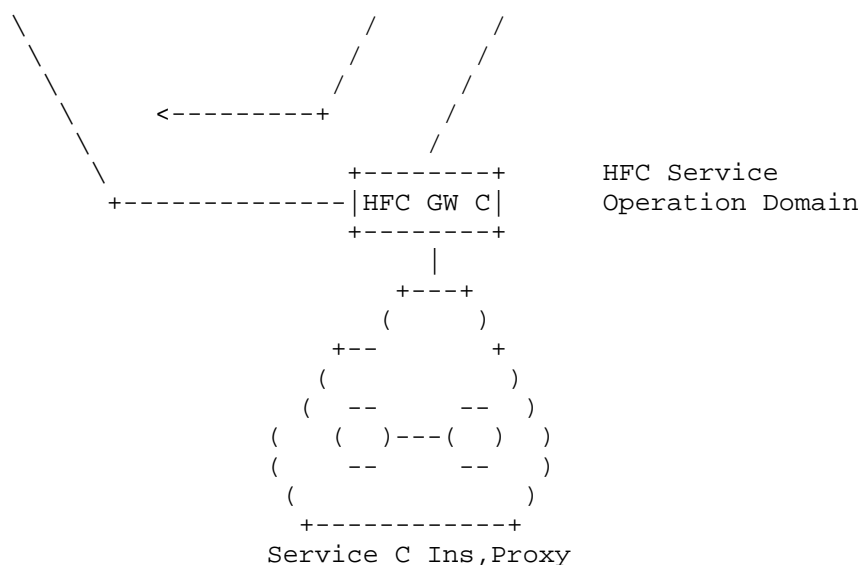


Figure 5: HFC Domain and end-to-end Delivery

- * When a service request accesses a corresponding application gateway, the service is identified as an HFC service agreed and dealt in a previous contract. Suppose the HFC service is decomposed into micro-services A, B and C. The HFC service is named with an identification, HFC ID. To process and fulfill the HFC service, the traffic would be steered to an access HFC gateway with packets carrying HFC ID and Service ID list. The endpoint for this HFC service would be configured as an SRv6 SID at the access gateway.
- * The access gateway identifies the HFC ID and Service ID list carried in the packet according to the SID filled in the destination field. Indexed by HFC ID and next Service ID (Initial Service ID), an HFC policy would be determined. Segment List of the HFC policy would be encoded to the packet with an Insert or Encapsulation pattern. Under a strict HFC mode for instance, Service SIDs for each HFC gateway and binding SIDs for interconnected forwarding paths would be included in the Segment List. Afterwards, the traffic would be steered to next HFC gateway.
- * With above displayed, HFC GW A is the first passing stand. HFC GW A would identify the next Service ID in the packet and implement a local lookup according to a Service SID. Therefore, the traffic

would be steered into a local or adjacent edge cluster. Furthermore, HFC GW A MAY remove the SRHs and record the segment list and segment left in a local cache.

- * When a flow aiming to a target service instance, a service pod for instance, it would be intercepted by a proxy or sidecar. The proxy or sidecar records the HFC ID and next Service ID in the packets and identifies returning traffic accordingly. Based on the HFC service lists, next service would be determined and its Service ID would overwrite the original one.
- * Since the traffic flow returns to HFC GW A, an original segment list and relevant information would be re-attached according to the records in the local cache. Afterwards, the traffic would be steered to next HFC GW similarly.

In the above SRv6 based HFC implementation, several SRv6 SIDs MAY be generally defined in this draft:

- * HFC Service SID: correlates with an HFC service forwarding table indexed by HFC ID and next Service ID, aiming to determine a forwarding path indicated by segment lists.
- * HFC Cache SID: correlates with an HFC local forwarding table and local cache table, aiming to record the forwarding information in the cache and forward the traffic to a local endpoint.
- * HFC Inherit SID: correlated with an HFC local cache table, aiming to determine and re-attach a matched original forwarding path.

```

+-----+           X           +-----+
|HFC GW A|-----|HFC GW B1|
+-----+ ----- +-----+

```

backup path

Service A Ins,Proxy

```

      +----+
      (    )
    +--      +
    (        )
    (  --      --  )
  (  (  )---(  )  )
  (    --      --  )
  (                )
+-----+
      |
+-----+
X+-----+
      +-----+
      (    )
    +--      +
    (        )
    (  --      --  )
  (  (  )---(  )  )

```

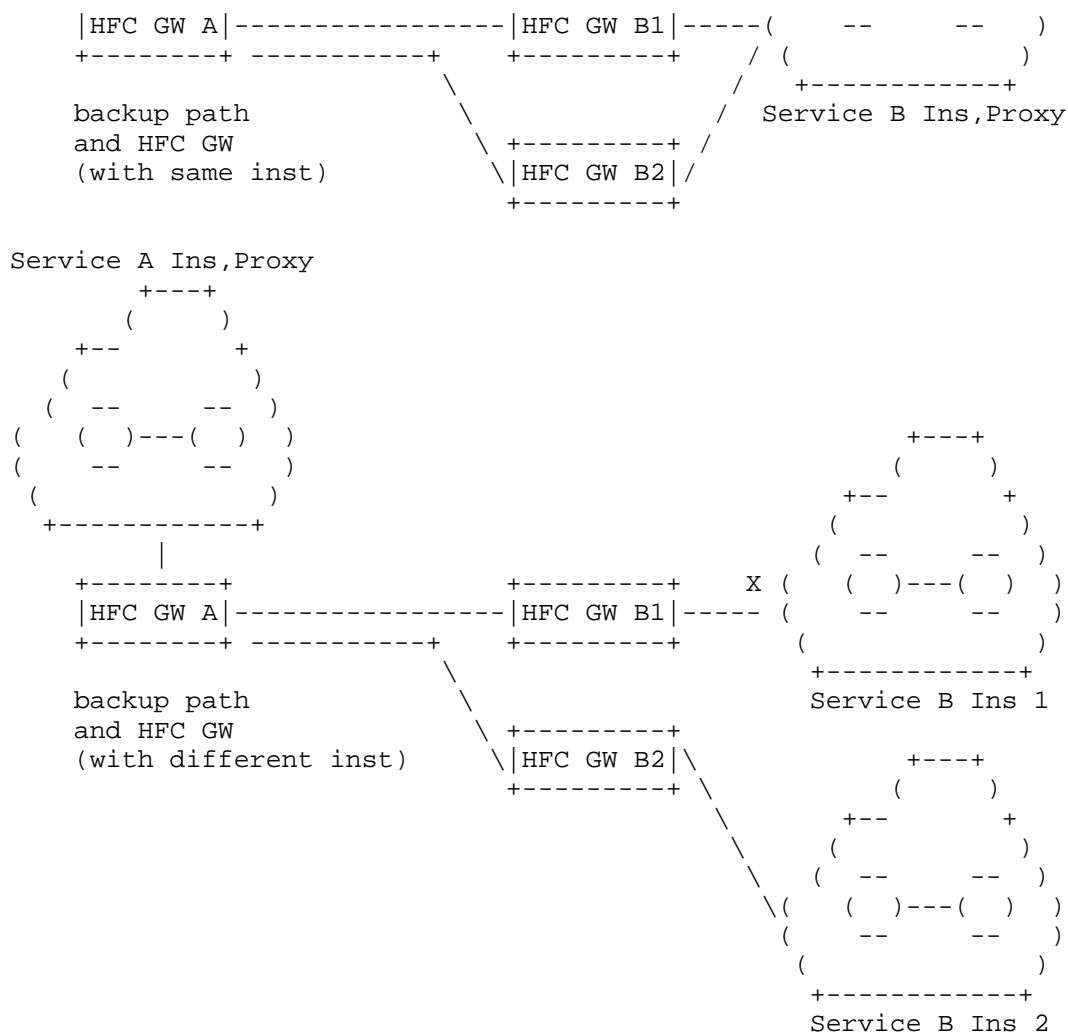


Figure 6: HFC Protection

Furthermore, SRv6 based HFC SHOULD support other features. For instance, backup paths would be orchestrated and accordingly configured at HFC GWs. End-to-end service observability would be achieved by distributed tracing and relevant schemes. More detailed implementation designs would be discussed in future works.

6. Security Considerations

TBA.

7. Acknowledgements

TBA.

8. IANA Considerations

TBA.

9. Normative References

[I-D.huang-rtgwg-us-standalone-sid]

Huang, D., Liang, J., Yang, F., Zhang, Y., and D. Yang,
"Use Cases-Standalone Service ID in Routing Network", Work
in Progress, Internet-Draft, draft-huang-rtgwg-us-
standalone-sid-01, 1 July 2024,
<<https://datatracker.ietf.org/doc/html/draft-huang-rtgwg-us-standalone-sid-01>>.

[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>>.

[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>>.

Authors' Addresses

Dongyu Yuan
ZTE Corporation
Nanjing
China
Phone: +86 13776623784
Email: yuan.dongyu@zte.com.cn

Yan Zhang
China Unicom
China
Email: zhangy1156@chinaunicom.cn

Daniel Huang
ZTE Corporation
Nanjing
China
Phone: +86 13770311052
Email: huang.guangping@zte.com.cn

Chuanyang Miao
ZTE Corporation
Nanjing
China
Email: miao.chuanyang@zte.com.cn