Workgroup: INTAREA
Internet-Draft: draft-ietf-xml2rfc-template-06
Published: 5 September 2023
Intended Status: Standards Track
Expires: 8 March 2024
Authors: D.H. Daniel
ZTE Corporation

B.T. Bin
ZTE Corporation

D.Y. Dong
Beijing Jiaotong University

Service Aware Network Framework

Abstract

Cloud has been migrating from concentrated center sites to edge nodes with responsive and agile services to the subscribers. This industry-wide trend would be reasonably expected to continue into the future which would enjoy geographically ubiquitous services. Rather than transmitting service data streams to the stable and limited service locations such as centered cloud sites, routing and forwarding network will have to adapt to the emerging scenarios where the service instances would be highly dynamic and distributed, and further more, demand more fine-grained networking policies than the current routing and forwarding scheme unaware of service SLA requirements. This proposal is to demonstrate a framework under which the above-mentioned requirements would be satisfied.

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 8 March 2024.

Table of Contents

1. Introduction

When it comes to user data security and service responsiveness, it’s imperative to migrate the cloud services and resources to the locations with good proximity to the users who could reside anywhere and launch service requests at any time in the ongoing and upcoming industry scenarios. Therefore, the cloud services and resources has been and would continue to be deployed in such a distributed way that the services would be ubiquitous, and scheduled and requested dynamically by various subscribers. Cloud and networking services and resources operate more coherently with each other as more and more services migrate into cloud. Network has to without gap delay adapt as the cloud shift into a new distributed architecture. The same service could be instantiated at multiple locations with different networking and computing resources which would be updated dynamically. Under this circumstance, the best service quality should be guaranteed by both fine-grained networking and computing policies.

This proposal introduces a light-weight service identification label as an index in the user plane to enable the network to be highly effectively aware of the dynamic requirements of various cloud applications. The service identification label is designed to purport to the fundamental and common services for which the service qualities should be guaranteed by both fine-grained networking and computing resources. Combined with an enhanced control plane, a logical sub-layer of service function has been employed in this framework to enable the network respond to the application’s networking and computing demands in a more fine-grained and intelligent way, which would bring significant benefits to all parties involved in the network and cloud ecosystem while ensure the framework to be compatible with the ongoing network architecture.

1.1. Requirements Language

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].

2. Terminology

*SAN. Service Awareness Network

*SIL. Service Identification Label, a light-weight label designed to indicate the fundamental and common service types

*SCMS. Service Control and Management System, an entity responsible for SIL management and controlling which includes materializing networking and computing policies in terms of SIL and delivering them to the SAN forwarding and routing nodes.

*SAN ingress: routing node maintaining computing resource as well as service status from remote cloud sites, and executing the cross-site routing policies in terms of the aforementioned status as well as the identification of computing service. SAN ingress usually resides at the network edge and works as ingress of the end to end service flow.

*SAN egress: routing node maintaining computing resource as well as service status from the geographically local cloud sites and being responsible for the last hop of the service flow towards the computing service instance in the specific cloud site. SAN egress usually resides at the network edge and works as egress of the end to end service flow.

*SAN relay: routing node which is optionally aware of computing resource and service status . SAN relay usually resides between SAN ingress and SAN egress and works as ordinary routing nodes and only gets involved in computing delivery when SAN ingress fails to do so. In particular, when an end-to-end networking policy in which SAN relay would be required to identify SIL with specific routing and forwarding behavior is involved, SAN relay would decapsulate and encapsulate SIL and execute the SIL-specific policies.

*Global Service Resource and Service Status(GSRS) : General cloud site status of the service-related resources and service which consists of overall resource occupation and types of service (algorithms, functions etc.) the specific cloud site provides. GSRS is maintained at SAN ingress and expected to remain relatively stable and change in slow frequency.

*Local Service Resource and Service Status(LSRS) : fine-grained cloud site status of the service-related resource and service which consists of status of each active service instance as well as its parameters which could impact the way the instance would be selected and visited by SAN egress. LSRS is maintained at SAN egress and expected to stay quite dynamic and change in high frequency.

*Service Instance(SI): an active instance of a SIL which resides in a host usually purporting to a server, container or virtual machine.

3. SAN framework and its chief components

An host address is request from DNS system to indicate the user’s intention of it’s service destination as well as establish a service connection under the conventional internet servivce architecture, while SAN framework proposes a refined architecture under which user’s intention is simply indicated by the service indentification label regardless of the actual service destination. Therefore, the center piece of the SAN framework is the light-weight service identification label. SIL should be confined within a limited and exhaustive service type space which only covers the indispensable and fundamental networking and computing service blocks. SIL could be component service which would be expected to be invoked by multiple application parties, or explicitly specified sub-stream of the same service data stream. A Service control and management system (SCMS) which could be a standalone or an enhanced version of the existing system. An entity with the authority of service and resource provisioning and delivery takes control of registry, publishing, authorization, authentication and policing of SIL within a closed governance domain. The life cycle of SIL runs through end user client, routing and forwarding network and cloud in terms of the end-to-end service process, and will survive in the above-mentioned closed governance domain permanently unless it’s withdrawn or updated with the new service identification label norms.

SIL is designed under this reference framework as a sophisticated interface between user and service as well as between service and network and cloud. User client requests the service through SIL encapsulated in user data packet header with customized networking and computing provisioning guaranteed by SAN routing and forwarding network which would index SIL with the corresponding networking and computing policies and resources configured from SCMS. Cloud governance system takes SIL as a public interface between user client and the cloud service provisioning system.

End user subscribes the service from SCMS and service client carries on materializing the service by instantiating an SIL into the service packet. Upon arriving at networking edge, SIL is identified and further enables the fine-grained networking service and computing-based routing and scheduling. Along the routing path the service data stream gets customized treatment in terms of SIL which is inherently defined to bo domain independent. Figure 1 is the reference framework of SAN. SAN domain reaches beyond conventional forwarding and routing network with SIL presence in both service client and server at end user devices and cloud sites respectively

which includes layer 4 and layer 3 from a network architecture perspective. Nevertheless, SAN forwarding and routing domain would be a predominant part of the entire reference framework.

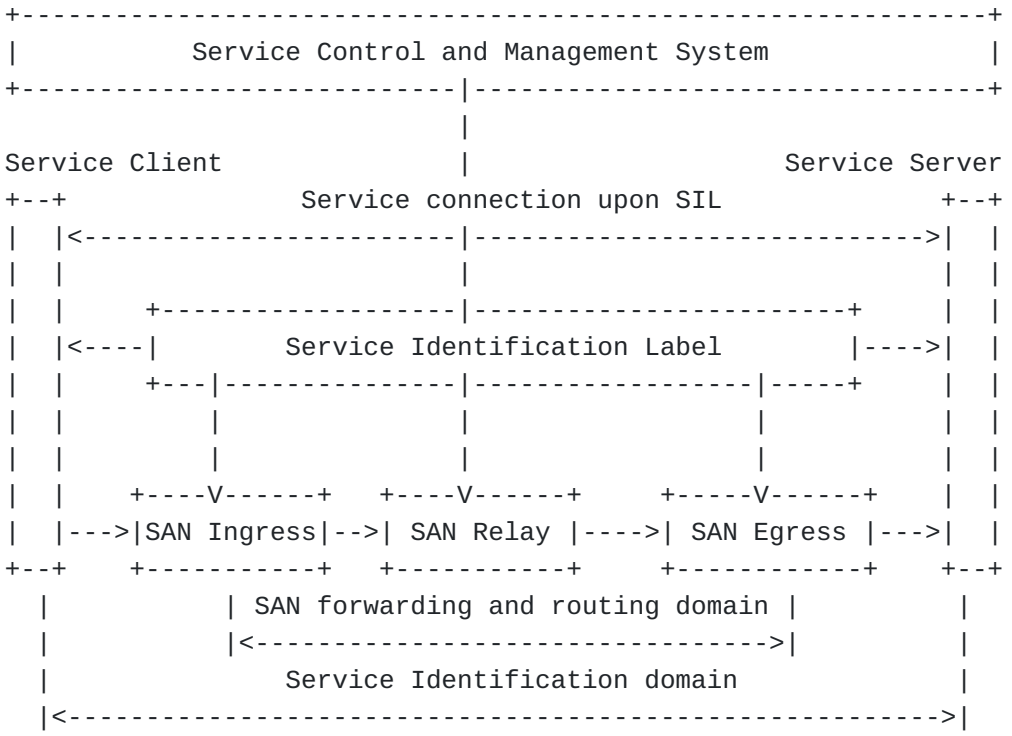

Figure 1

4. Layer 4 of SAN framework

From perspective of layer 4 which builds and maintains the service connection for the application, end user quality of experience would be significantly increased with a host-address irrelevant connection that remains stable regardless of whether or not the application server host shifts. It’s achieved by establishing the layer 4 connection with SIL which indicates only the service type without semantics of server host location.

In particular, SIL could be encapsulated in layer 4 protocol-intact way and layer 4 protocol-updated way. The former entails informing the app client of an SIL when it comes to the process of informing it of a destination address such as DNS inquiry process. Therefore, layer 4 protocol remains intact by taking SIL as destination address and proceeds as usual. SIL would be identified and treated with fine-grained networking and computing policies of which an addressable server host address would be selected and encapsulated in the out layer routing header by SAN ingress. A mapping between SIL and server host address could be made either by SAN egress or app client and server. So far as the latter is concerned, a

dedicated sub-layer between layer 4 and layer 3 should be employed to encapsulate SIL which is to be used by layer 4 protocol to establish the service connection. The dedicated SIL sub-layer survives through end user, SAN network and the cloud, and SAN nodes would index the SIL from this sub-layer to execute the fine-grained service treatment in terms of both networking and computing policies.

SIL structure designing as well as the dedicated sub-layer protocol specification are out of scope of this proposal.

5. Layer 3 of SAN framework

Layer 3 SAN domain is responsible for the fine-grained treatment of service flow referenced by SIL in terms of both networking and computing requirements.

5.1. SAN ingress

SAN ingress extracts SIL from either the destination address field to be encapsulated with SIL or the dedicated SIL sub-layer as illustrated in section 4, and determines its fine-grained networking policy as well as its next hop in light of which cloud site hosts the satisfying service node. Under this particular circumstance, the next hop could be either an SAN egress which connects directly to the selected remote cloud site or the selected service server within the local cloud site.

Meanwhile, SCMS delivers both fine-grained networking and computing policies in terms of SIL to SAN ingress which maps the extracted SIL to the corresponding policies and executes them upon the forwarding plane. SCMS could be the combination of computing and networking orchestrator and controller which logically should be two standalone entities. The orchestrator part is responsible for the general scheduling policies of computing and networking resources, such as at which threshold a policy is triggered, while the controller generates and delivers the SIL-indexed policies according to the scheduling policies from the orchestrator. Under distributed routing scheme, the computing and networking policies generated by the protocols (such as BGP) within the SAN forwarding and routing node could also be part of SCMS logically.

5.2. SAN relay

SAN relay is optionally designed to identify SIL and execute the according service routing as well as networking policies when SAN ingress alone could not do this. Particularly, SIL-indexed routing status would not necessarily be maintained by SAN ingress once and all, so SAN relay could play the role of recursive routing table query. Also, SAN relay could coordinate with SAN ingress to execute

the networking policy in terms of SIL when necessary. Nevertheless, SAN relay could be reduced to an ordinary forwarding and routing node without knowledge of SIL when SAN ingress completes the service and networking service policies.

5.3. SAN egress

When it comes to SAN egress, SIL could be extracted from the destination address field when there is a tunnel encapsulation as an outside header such as SRH (SRv6 Header), or from the dedicated SIL header when the original SIL in the destination address field has been replaced with a selected SAN egress address by SAN ingress. SAN egress maps the SIL with the networking and computing policies from SCMS and executes them upon the forwarding plane, the next hop would be selected through the computing policy as a server hosting the service or a proxy of the service. SAN egress always removes the outside tunnel header and terminates the networking policy.

The networking and computing policy exchange with SCMS follows the same process of SAN ingress other than the difference of particularity of the policies generated and delivered.

5.4. SAN control plane consideration

As the conventional IP routing control plane scheme in place, SAN control plane could also be deployed by centralized and distributed scheme or combination of both with additional service status as well as the polices. Service-related control plane only has impacts upon SAN ingress and egress, and could logically be decoupled from the conventional IP control plane.

5.4.1. Centralized control plane consideration

SIL-centered networking and computing resource and policy generation and maintenance is the key feature of SAN control plane. Networking resource and policy generally aligns with the existing scheme other than the SIL-centered networking policy is differentiated in a fine-grained granularity. As far as computing resource and policy is concerned, LSRS’s volatility makes it infeasible to be maintained and controlled in a centralized entity, GSRS is the chief computing resource and service status information to be collected and managed in the controller with regard to service stream delivery in routing network architecture. Routing and forwarding policies from GSRS calculated in the centralized controller apply only to the segment between SAN ingress and egress, while the second segment routing policy from SAN egress to the selected service instance in the cloud site is determined by LSRS at SAN egress.

Hierarchically centralized control plane architecture would be strongly recommended under the circumstances of nationwide networking and computing management and scheduling.

5.4.2. Distributed control plane consideration

Networking resource is notified and updated through existing distributed protocols (BGP/IGP etc.) and the SIL-centered networking policy would be formulated as well. When it comes to computing resource, GSRS is updated among the SAN edge routers which have been connected in a mesh way that each pair of edge routers could exchange GSRS to each other, while LSRS will be unidirectionally updated from cloud site to the associated SAN edge router in which LSRS is maintained and its update process is terminated.

Protocol consideration upon which GSRS and LSRS is updated is out of the scope of this proposal and will be illustrated in forthcoming draft.

5.4.3. Hybrid control plane consideration

In terms of the particularity of service-related resource updating and notification, it would be more efficient to update the GSRS by a distributed way than a centralized way in terms of routing request and response in a limited network and cloud domain, but would be the opposite case in a nationwide circumstance. This is how hybrid control plane could be deployed in such a scheme that overall optimization could be achieved.

5.5. SAN user plane consideration

5.5.1. SIL encapsulation

Service identification label is the predominant index across the entire SAN framework ranging through user terminal, forwarding and routing network and cloud with SIL working as the virtual destination. Data plane determines the routing and forwarding orientation with SIL by inquiring GSRS and LSRS at SAN ingress and SAN egress respectively. SIL encapsulation could be achieved by extending the existing packet header and also achieved by designing a dedicated SIL sub-layer, which along with the specific structure of SIL are out of the scope of this proposal and will be illustrated in forthcoming draft.

5.5.2. SIL in forwarding and routing network

SAN ingress obtains SIL from either the destination address field or the dedicated SIL sub-layer of the user packet header explicitly or mapping from the traditional 5 tuples implicitly. Either way SIL starts being indexed for networking and computing policies until it

arrives at SAN egress. SIL in the dedicated sub-layer would remain intact in the user packet header, while SIL in the destination address field would remain intact if SAN ingress employs a tunnel header or could be replaced with a selected SAN egress address with a dedicated SIL encapsulation in extension headers such as DoH, SRH, HBH. In the case of absence of SIL in user packet header, SAN ingress would generate and encapsulate SIL in extension headers by combination of existing parameters from user packet header such as 5 tuples etc. Apart from the networking and computing policy execution at SAN ingress and egress, SIL could be ignored by SAN relay and without computing state maintenance unless the mapping between SIL and GSRS fails or SIL fine-grained networking policy is involved.

5.5.3. SIL-based routing

As illustrated in section 3, SIL encapsulated in the headers and maintained in GSRS and LSRS indicates an abstract service type rather than a geographically explicit destination label, thus the routing scheme based upon SIL is actually a two-part and two-layer process in which SIL only indicates the routing intention of user’s requested service type while routing does not actually materialize in forwarding plane and the explicit routing destination of the two segments would be determined by GSRS and LSRS respectively. Therefore the actual routing falls within the traditional routing scheme which remains as they are.

Apart from the indication of service routing intention, SIL could also indicates a specific network serivice requirements by associating the networking service policy in GSRS which would therefore schedule the network resources such as an SR tunnel, guaranteed bandwidth etc. at SAN ingress.

Therefore, GSRS and LSRS in control plane along with SIL encapsulation in user plane enables an logical service routing sub-layer which is able to be aware of the computing status from cloud sites and forward the service flow in terms of networking services as well as computing resources. Nevertheless, this logical sub-layer remains predominantly at SAN ingress and egress nodes. There're drafts such as [I-D.liu-dyncast-ps-usecases] and [I-D.li-dyncast-architecture] which analyze the benefits of computing-based routing and demonstrate an anycast-as-service-identification solution.

5.6. Hierarchical service routing architecture

In addition to the existing networking resource and status sensing scheme, SAN routing and forwarding network is designed specifically to enable sensing the service-related resource and service status from the cloud sites and routing the service flow according to both

network and computing metrics as illustrated in figure 2. The architecture is a horizontal convergence of cloud and network, while the latter maintains the converged resource status and thus is able to achieve an end to end routing and forwarding policy from a perspective of cloud and network resource. PE1 maintains GSRS with a whole picture of the multiple cloud sites, and executes the routing policy for the network segment between PE1 and PE2 or PE3, namely between SAN ingress and egress, while PE2 maintains LSRS with a focus picture of the cloud site where S1 resides, and establishes a connection towards S1. S1 is an active instance of a specific service type. On top of the role of SAN egress which maintains LSRS, PE2 and PE3 also fulfill the role of SAN ingress which maintains GSRS from neighboring cloud sites. P provides traditional routing and forwarding functionality for computing service flow, and optionally remains unaware of service-related status.

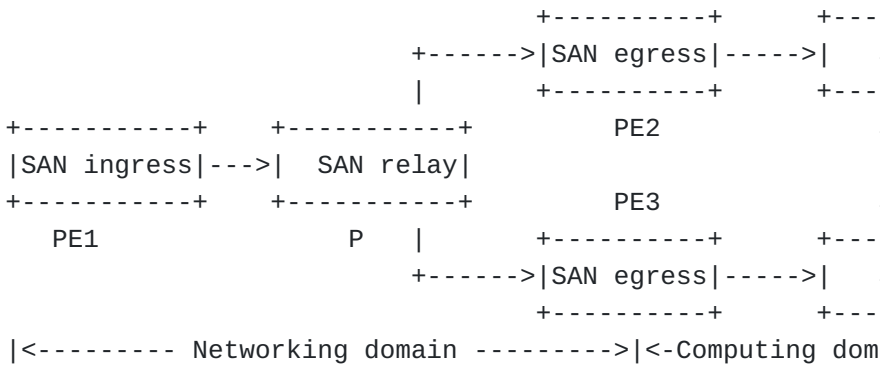

Figure 2

5.6.1. Routing scheme in line with multiple service-related resources granularity

Status updates of service-related resource and service in the cloud sites extend in a quite broad range from relatively stable service types and overall resource occupation to extremely dynamic capacity changes as well as busy and idle cycle of service instances. It would be a disaster to build all of the status updates in the network layer which would bring overburdened and volatile routing tables.

It should be reasonable to divide the wide range of service-related resource and services into different categories with differentiated characteristics from routing perspective. GSRS and LSRS correspond to cross-site domain and local site domain respectively, and GSRS aggregates the service-related resource and service status with low update frequency from multiple cloud sites while LSRS focuses only upon the status with high frequency in the local sites. Under this two-granularity scheme, service-related routing table of GSRS in the

SAN ingress remains in a position roughly as stable as the traditional routing table, and the LSRS in the SAN egress maintains a near synchronized state table of the highly dynamic updates of service instances in the local cloud site. Nonetheless, LSRS focusing upon a single and local cloud site is the normal case while upon multiple sites should be exception if not impossible.

5.6.2. Two-segment routing and forwarding

When it comes to end to end service flow routing and forwarding, there is an status information gap between GSRS and LSRS, therefore a two-segment mechanism has to be in place in line with the two-granularity routing scheme demonstrated in 5.6.1. As is illustrated in figure 3, R1 as ingress determines the specific service flow's egress which turns out to be R2 according to policy calculation from GSRS. In particular, the SIL from both in-band (user plane) and out-band (control plane) is the only index for R1 to calculate and determine the egress, it's highly possible to make this egress calculation in terms of both networking (bandwidth, latency etc) and computing requirements. Nevertheless, the two SLA routing optimization could be decoupled to such a degree that the traditional routing algorithms could remain as they are. The convergence of the SLA policies as well as the methods to make SAN ingress aware of the two SLA would be illustrated by the example work flow of section 7. Nevertheless, the specific solution is out of scope of this proposal.

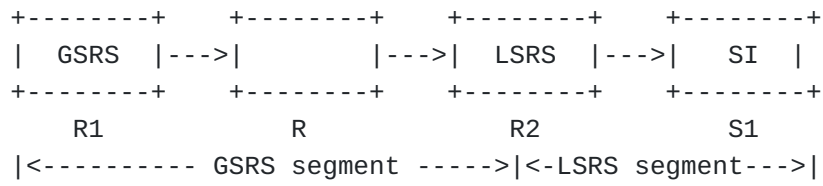

Figure 3

When the service flow arrives at R2 which terminates the GSRS segment routing and determines S1 which is the service instance selected according to LSRS maintained at R2. Again SIL is the only index for LSRS segment routing process.

5.6.3. Cross-domain computing routing and forwarding

Co-ordinated computing resource scheduling among multiple regions which are usually connected by multiple network domains, is an important part of intended scenarios with regard to why computing-based scheduling and routing is proposed in the first place. The two-segment routing and forwarding scheme illustrated in5.6.2 is a typical use case of cross-domain service routing and forwarding and a good building block for the full-domain scenario solution. Service

status information is brought into network domain to enable the latter scheduling routing policies beyond network. However, a particular scheme has to be put in place to ensure mild and acceptable impacts upon the ongoing IP routing scheme. A consistent SIL across terminal, network (multiple domains) and cloud along with hierarchical SIL-indexed service-related resource and service status which corresponds with different network domains, is the enhanced full-domain routing and forwarding solution. Each domain maintains a corresponding ervice-related resource and service status at its edge node and makes the service-based routing for the domain-specific segment which should be connected by the neighboring segments.

5.6.4. Service traffic affinity

SIL holds the only semantics of the service type that could be deployed as multiple instances within specific cloud site or across multiple cloud sites, SIL in the destination field is not explicit enough for all of the service flow packets to be forwarded to a specific host. Traffic affinity has to be guaranteed at both ingress and egress. Once the egress is determined at SAN ingress, the binding relationship between the egress and the service flow’s unique identification (5-tuple or other specifically designed labels) is maintained and the subsequent flow could be forwarded upon this binding table. Likewise SAN egress maintains the binding relationship between the service flow identification and the selected service instance.

Traffic affinity could be guaranteed by mechanisms beyond routing layer, but they will not be in the scope of this proposal.

5.7. Logical sub-layer of service routing in forwarding and routing network

A SIL-indexed networking and computing policy state from SCMS is maintained at the control plane in SAN forwarding and routing network. The policy state regulates and guides the forwarding and routing behaviors of SAN nodes on the basis of SIL as well as the identification of the service stream. Namely, the fine-grained networking and computing policies bring new abilities for forwarding and routing network to be highly efficiently aware of the service from perspectives of both networking and computing requirements.

Either SIL is originated from user client or SAN ingress, SIL’s presence in the data plane is guaranteed by various encapsulation solutions and could be extracted and indexed by any node for execution of either networking policy, computing policy or both.

As for the networking connection services for which only network-domain resources would be involved, SIL provides a fine-grained

interface between network and application which would be unavailable otherwise. Under this particular circumstance, SIL-indexed networking policy state maintenance actually aligns perfectly with the existing scheme.

When it comes to the computing services for which both networking and computing resources would be involved, SIL is a light-weight index for the fine-grained policies, and the particular computing policies are clearly decoupled with that of networking policies .

Therefore, a logical SIL sub-layer with both control and data plane has been conceptually employed within SAN forwarding and routing architecture. The sub-layer simply delivers the SIL specific policies but would leave them for the existing routing layer to execute the policies on a basis of service stream and packet.

6. Governance and life cycle of service identification label

6.1. Originality and governance of SIL

SIL is designated to indicate the fundamental and common service types, and could be registered from both networking and computing domain with regard to networking connection services and comprehensive computing services respectively. The SIL templates should be specified by the entity which is both technically able to coordinate the SIL-indexed networking and computing resources and services, or by public standardization organizations. SCMS publishes the SIL which has been authenticated, authorized and configured with networking and computing policies.

Application developer and operator subscribes SIL from SCMS and integrates it into its application system, and the service client initiates the service by encapsulating SIL into its IP protocol headers. SIL template specification is out of scope of this proposal.

6.2. Life cycle of SIL

Upon the registration and publication of the service, SIL is active and available through the ecosystem of terminal, network and cloud until it’s withdrawn and terminated by SCMS. Life cycle of SIL would not terminate with a specific end of user service, and the same SIL could be instantiated by multiple users simultaneously. SIL renders the services as an effective interface between service and network which is inherently absent in the decoupled internet protocol system.

7. An example of end-to-end SAN work flow

SIL is controlled and managed by SCMS as illustrated in section 6.1, therefore SIL subscription and service agreement process between service client and SCMS should be finished before the service client initiates a service request and starts sending service data stream while SIL would be encapsulated in layer 3 header and lives on through the service terminal, SAN routing and forwarding network, and service server in the cloud site. An example work flow of cloud game application will be demonstrated under SAN reference framework. From networking and computing resource perspectives, rendering algorithm of real-time game situation data should be the dominant sub-service of cloud game application. The work flow instantiates the mechanism under which the rendering algorithm would be materialized by SAN.

7.1. Initiation and maintenance of rendering algorithm in SAN system

Rendering algorithm is identified as a fundamental and common service which could be invoked by multiple parties, and at least one provider has registered to SCMS and has been authenticated and verified by SCMS for the service availability. An SIL might be allocated or activated as SIL-RA which should be structured service identification specifically and uniquely allocated for the rendering algorithm.

When it comes to the networking requirements of rendering algorithm, SCMS verifies the corresponding networking resource availability as well as capability and establishes a comprehensive SLA state for rendering algorithm indexed by SIL-RA.

When a rendering algorithm provider registers/withdraws the rendering algorithm service and the corresponding networking resources update to a degree the service would not be able to be rendered as promised, the computing and networking SLA state of SIL-RA should be updated accordingly. The signal as well as data stream of SIL-RA is illustrated as an example work flow in figure 4.

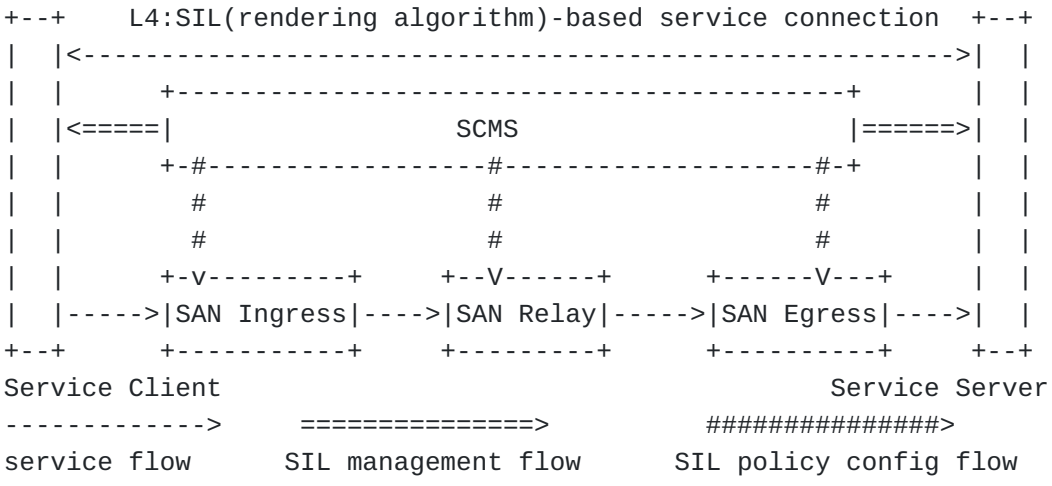

Figure 4

The service client launches a cloud game application request to the designated server in a centered cloud site, and a cloud game service connection thus would be established, while the rendering algorithm might be deployed in the edge cloud nodes for the sake of responsiveness of the game service.

7.2. Configuration of a rendering algorithm (as SIL-RA) in SAN forwarding and routing network

End-to-end latency determines both the user experience and the feasibility of the service deployed in the cloud site. The latency resulting from the service server and the network path depends on the rendering algorithm capability along with the computing resources and the networking delay respectively. Therefore, SCMS maps a networking connection policy with 10 mini-seconds of latency and computing policy with GPU to SIL-RA, and makes the networking and computing policies configuration to the SAN forwarding and routing nodes such as SAN ingress, SAN relay and SAN egress through network management and control interfaces.

7.3. Publication/Subscription of SIL-RA

Upon completion of SIL-RA initiation and configuration, SIL-RA is ready to be published to and subscribed by all of the interested parties. The SIL-RA publish/subscribe process is more of a commercial agreement than a technical exchange process. The rendering algorithm service client initiates the service by sending the service requests as well as data stream with SIL-RA encapsulated in layer 3 header. The rendering algorithm service connection would be based soly upon SIL-RA which could be indexed and mapped to a computing node located in other edge cloud site, and thus the actual

service connection could be different from that of cloud game application connection as illustrated in section 7.1.

7.4. SIL-RA service data stream treatment at SAN forwarding and routing network

In the ongoing process of the real-time cloud game service, the rendering algorithm service would be invoked simultaneously during the entire life-cycle of the game service. Therefore, the rendering algorithm would not be invoked as the initiation of the game service and would also not reside in a same cloud site of the master game service. The rendering algorithm service should be specifically guaranteed by SAN with comprehensive networking and computing policies configured for it. The game service client would initiate an independent rendering algorithm service by encapsulating SIL-RA in the data packet headers.

7.4.1. SIL-RA service data stream treatment at SAN ingress

Upon arrival of SIL-RA service data stream, SAN ingress extracts SIL-RA from the designated field of layer 3 header and identifies the type of the SIL-RA. SIL-RA availability as well as the supporting computing resources among multiple servers has been maintained in the service routing table, the SAN egress connecting to a proper SIL-RA server would be selected.

As configured in 7.3, SIL-RA service stream should be transmitted by a forwarding and routing policy with 10 mini-seconds latency between SAN ingress and the selected SAN egress. The SIL-RA service data stream would be guided into a designated networking path. As far as bounded latency of routing network is concerned, a path-specific flow label could be employed in line with the underling networking technologies such as Detnet.

7.4.2. SIL-RA service data stream treatment at SAN relay

SIL-RA service data stream treatment with SIL-RA involved is actually optional, it’s only necessary when SAN ingress relays the SIL-RA service routing request to a SAN relay where the SIL-RA indexed service routing table is homed or one or more SAN relay nodes has to be involved to provide SIL-RA specific networking connection service.

7.4.3. SIL-RA service data stream treatment at SAN egress

As illustrated in section 5.5, SIL-RA indexed service routing table would be aggregated with two class granularity at SAN ingress and SAN egress respectively. So SIL-RA would be extracted from the designated field and mapped with the LSRS table. The final hop purporting to the selected rendering algorithm service server would

be determined by SAN egress and the service data stream would be forwarded to the server or the server’s proxy accordingly.

SAN egress should be the terminating node of the networking connection path and the the associated path header would be removed.

7.5. SIL-RA service data stream treatment at cloud site

SIL-RA encapsulation in layer 3 header would remain intact until the data packet arrives at the cloud site. SIL-RA would be identified and treated as both a service connection ID by the rendering algorithm server and a rendering algorithm to be guided and scheduled by the cloud system respectively .

8. Acknowledgements

To be added upon contributions, comments and suggestions.

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

SIL employment in SAN framework would bring security challenges for user and application in the cloud sites as well as SAN forwarding and routing network. SCMS is responsible for registry, authorization and authentication of SIL and acts as the first check point for SIL. The detailed specification of the security process of SCMS is out of scope of this proposal. When it comes to SAN forwarding and routing network, it’s imperative for the service gateway to execute security policies with regard to SIL and the specific security solution would be illustrated in a dedicated proposal.

11. Informative References

[I-D.li-dyncast-architecture] Li, Y., "Dynamic-Anycast Architecture", February 2021, <https://datatracker.ietf.org/doc/draft-li-dyncast-architecture/>.

[I-D.liu-dyncast-ps-usecases] Liu, Peng., "Dynamic-Anycast (Dyncast) Use Cases and Problem Statement", February 2021, <https://datatracker.ietf.org/doc/draft-liu-dyncast-ps-usecases/>.

[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>.

Authors' Addresses

Daniel Huang
ZTE Corporation
Nanjing

Phone: +86 13770311052
Email: huang.guangping@zte.com.cn

Bin Tan
ZTE Corporation
Nanjing

Phone: +86 13918622159
Email: tan.bin@zte.com.cn

Dong Yang
Beijing Jiaotong University
Beijing

Email: dyang@bjtu.edu.cn