Workgroup: Network Working Group
Internet-Draft:
draft-contreras-rtgwg-rosa-gaar-01
Published: 9 July 2023
Intended Status: Standards Track
Expires: 10 January 2024
Authors: LM. Contreras
Telefonica
D. Trossen
Huawei Technologies
J. Finkhaeuser
Interpeer gUG
P. Mendes
Airbus

Gap Analysis and Requirements for Routing on Service Addresses

Abstract

The term 'service-based routing' (SBR) captures the set of mechanisms for the steering of traffic in an application-level service scenario. We position this steering as an anycast problem, requiring the selection of one of the possibly many choices for service execution at the very start of a service transaction.

This document builds on the issues and pain points identified across a range of use cases, reported in [I-D.mendes-rtgwg-rosa-use-cases]. We summarize the key insights and provide a gap analysis with key technologies related to the problem of SBR, developed by the IETF over many years. We further outline the requirements to a system that would adequately close those gaps and thus address the pain points of our use cases. Those requirements will be used for outlining a suitable architecture framework in a separate document.

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 10 January 2024.

Table of Contents

1. Introduction

Virtualization and the proliferation of serverless service provisioning methods have driven the capability to dynamically deploy services in more than one network location, allowing for scaling both horizontally and vertically in a number of use cases, some of which can be found in [I-D.mendes-rtgwg-rosa-use-cases]. A key problem in such use cases is that of steering the service requests stemming from the applications, a mechanism we label as service-based routing (SBR). A key constraint in realizing solutions for such problem is the possible distribution of more than one service instance across several network locations, posing the SBR problem as an inherently anycast one.

Unlike existing methods for SBR, some of which we will survey in this document, we envision a system we call routing on service addresses (ROSA), that allows for suitable service-specific anycast decisions to be made under a possibly high frequency of change to the notion of the 'best' instance to be chosen with the expectation to yield in better performance, such as improved service completion latency, utilization, and others.

At the same time, it is important to recognize that we do not aim for replacing existing service routing capabilities, most notably the DNS as the main form of resolving a service name into routing locator; we see those capabilities working perfectly well for many Internet services. However, it is important to understand the gaps that those existing methods show in realizing the emerging use cases of high dynamicity in service relations. This document surveys key technologies, developed in the IETF over recent years, in order to identify the gaps of those technologies to deliver suitable solutions to the pain points identified in our use cases of [I-D.mendes-rtgwg-rosa-use-cases].

Complementing our gap analysis, we also formulate requirements for a solution to those pain points. We link the various requirements to observed issues in our use cases [I-D.mendes-rtgwg-rosa-use-cases] for better illustration and reasoning for their inclusion.

In the remainder of this document, we first introduce in Section 2 a terminology that provides the common language used throughout the remainder of the document; this terminology is kept in sync with the other ROSA draft. We then summarize the key observations from our use cases in [I-D.mendes-rtgwg-rosa-use-cases] as a recap for the following gap analysis in Section 4. The insights from our gap and use case analysis then leads us to the requirements in Section 5,

before outlining in Section 6 the expected benefits from realizing those requirements in a suitable system.

2. Terminology

The following terminology is used throughout the remainder of this document:

Service: A monolithic functionality that is provided according to the specification for said service.

Composite Service: A composite service can be built by orchestrating a combination of monolithic (or other composite) services. From a client perspective, a monolithic or composite nature cannot be determined, since both will be identified in the same manner for the client to access.

Service Instance: A running environment (e.g., a node, a virtual instance) that provides the expected service. One service can involve several instances running within the same ROSA network at different network locations.

Service Address: An identifier for a specific service.

Service Instance Address: A locator for a specific service instance.

Service Request: A request for a specific service, addressed to a specific service address, which is directed to at least one of possibly many service instances.

Affinity Request: A request to a specific service, following an initial service request, requiring steering to the same service instance chosen for the initial service request.

Service Transaction: A sequence of higher-layer requests for a specific service, consisting of at least one service request, addressed to the service address, and zero or more affinity requests.

Service Affinity: Preservation of a relationship between a client and one service instance, with the initial service request creating said affinity and following affinity requests utilizing said affinity.

ROSA Provider: Realizing the ROSA-based traffic steering capabilities over at least one infrastructure provider by

deploying and operating the ROSA components within its defining ROSA domain.

ROSA Domain: Domain of reachability for services supported by a single ROSA provider.

ROSA Endpoint: A node accessing or providing one or more services through one or more ROSA providers.

ROSA Client: A ROSA endpoint accessing one or more services through one or more ROSA providers, thus issuing services requests directed to one of possible many service instances that have previously announced the service address provided by the ROSA client in the service request.

Service Address Router (SAR): A node supporting the operations for steering service requests to one of possibly many service instances, following the procedures outlined in a separate architecture document.

Service Address Gateway (SAG): A node supporting the operations for steering service requests to service addresses not announced to SARs of the same ROSA domain to suitable endpoints in the Internet or within other ROSA domains.

3. Observations from Use Cases

Several observations can be drawn from the use case examples in [I-D.mendes-rtgwg-rosa-use-cases] in what concerns their technical needs:

1. Service instances for a specific service may exist in more than one network location, e.g., for replication purposes to serve localized demand, while reducing latency, as well as to increase service resilience.

2. While the deployment of service instances may follow a longer term planning cycle, e.g., based on demand/supply patterns of content usage, it may also have an ephemeral nature, e.g., through scaling in and out dynamically to cope with temporary load situations, enabled by the temporary nature of serverless functions.

3. Knowing which are the best locations to deploy a service instance is crucial and may depend on service-specific demands, realizing a specific service level agreement (with an underlying decision policy) that is tailored to the service and agreed upon between the service platform provider and the communication service provider.

4. Decisions for selecting the 'right' or 'best' service instance may be highly dynamic under the given service-specific decision policy and thus may change frequently with demand patterns driven by the use case. For instance, in our example on Distributed Mobile applications and Metaverse in Section 3.4 and 3.8 of [I-D.mendes-rtgwg-rosa-use-cases], respectively, human interaction may drive the requirement for selecting a suitable service instance down to few tens of milliseconds only, thus creating a need for high frequency updates on the to-be-chosen service instance. As a consequence, traffic following a specific network path from a client to one service instance, may need to follow another network path or even utilize an entirely different service instance as a result of re-applying the decision policy.

5. Minimizing the latency from the initiating client request to the actual service response arriving back at the client is crucial in many of our scenarios. Any improvement on utilizing the best service instance as quickly as possible, thus taking into account any 'better' alternative to the currently used one, is crucial for reducing service request completion latency.

6. The namespace for services and applications is separate from that of routable identifiers used to reach the implementing endpoints, i.e., the service instances. Resolution and gateway services are often required to map between those namespace, adding management and thus complexity overhead, an observation also made in [Namespaces2022].

7. A specific service may require the execution of more than one service instance, in an intertwining way, which in turn requires the coordination of the right service instances, each of which can have more than one replica in the network.

We can conclude from our observations above that (i) distribution (of service instances), (ii) dynamicity in the availability of and choosing the 'best' service instance, and (iii) efficiency in utilizing the best possible service instance are crucial for our use cases.

4. Gap Analysis

We now discuss observations and suitability of existing technologies for realizing the use cases in [I-D.mendes-rtgwg-rosa-use-cases]. We first survey technologies that possibly provide similar SBR functionality to our use cases. Here, we have currently identified

the DNS (and solutions based on it), CATS, LISP, and ALTO as such technologies.

We then outline works that are related to certain aspects of SBR only for the purpose of explaining differences and relations for possible future integration or touching points in solutions to ROSA. Here, we currently include technologies such as SFC, SPRING, and TVR. Future discussions and work may extend on both of those areas for a more comprehensive analysis.

4.1. Domain Name System (DNS)

The Domain Name System (DNS) is the most prevalent method being used for service-based routing in that it supports the resolution of a domain name, such as foo.com, to an IP address, which is then used for subsequent message transfer between sender and receiver. We see, thus, the DNS and methods extending but basing themselves on the DNS, such as Global Server Load Balancing, as the baseline for SBR. In the following, we provide insights into the main technology and the gaps identified towards ROSA objectives.

4.1.1. Technology Overview

The DNS [RFC1035] provides an explicit method for mapping domain names onto an IP locator, often referred to as 'early binding'. Those mappings are provided based on previous DNS registrations of IP locators to certain domain names.

There are many extensions to this basic lookup mechanism, some of which are relevant to our discussion. For instance, DNS extensions may be used to base the decision on which IP address of several to pick based on, e.g., geo-location or load information. For the latter, load balancing is provided alongside the DNS resolver, e.g., in the form of Global Server Load Balancing (GSLB) [GSLB] solutions in CDNs. Furthermore, a health check functionality may be provided to resolve IP address failures, providing alternatives to detected failures of reachability.

4.1.2. Relation to ROSA

As mentioned upfront, the explicit resolution provided by the DNS is our baseline for comparison due to its widespread use in the Internet. Albeit its rather static nature of assigning IP addresses to domain names, it is sufficient for many of the use cases of the Internet, where the initial selection of a suitable server address suffices. We thus see the DNS to continue being a vital component of the Internet and thus only focus in our following gap analysis on those shortcomings in relatin to our identified use cases.

4.1.3. Gaps

There are number of key differences and gaps to the desired properties of a ROSA system. Several of those gaps have already been identified in [I-D.yao-cats-gap-reqs] and also apply here:

1. Resolution latency: The explicit resolution for a DNS name takes additional time that adds to the overall following data transfer with the selected IP address. It thus adds to the completion time of the high layer request that is being made. Many measurements exist for such latency but its extend heavily depends on the provisioning for the underlying resource that exposes the selected IP address. [OnOff2022], for instance, outlines latencies ranging from 15 to 45 milliseconds where the used DNS-based systems range from local ISP provided DNS to more complex CN-provided GSLB [GSLB] solutions, while resolutions that require several DNS resolver steps may easily require 100ms and more. For many of our use cases in [I-D.mendes-rtgwg-rosa-use-cases], such latency is prohibitive since it may either heavily contribute or even exceed the available delay budget of the application. But resolution latency may also be cummulative, e.g., for web browsing, as discussed in [OnOff2022], particularly when needing to resolve a larger number of distinct (in terms of domain names) objects within a given meta-object (such as a webpage). DNS latencies may still become a decisive factor, negatively impacting the end user experience. Through the in-band selection method in ROSA, this explicit resolution latency is entirely avoided, therefore also reducing the sending of 4 messages (2 for resolution and 2 for the initial data transfer) across the client access, often being the bottleneck in Internet access, to merely two messages for the in-band discover instead.

2. Acting on stale information: DNS applies a local caching model to remove the burden on the DNS system when subsequently the same request is issued again by the application. This can, however, lead to acting on stale information for those cases where the mapping has changed, more so for services where the mapping is meant to change frequently. Applications may flush the local DNS cache after every lookup, which may however lead to overburdening the DNS with the number of renewed requests, possibly being perceived as a denial-of-service attack by the DNS. ROSA aims at avoiding any stale information or at least minimizing stale information through more reactive routing or entirely local scheduling selection methods.

3. Supporting dynamic resolution changes: Updating a mapping of a domain name to an IP locator takes time to propagate. Unlike in local environments, where extensions such as DNS-SD [RFC6763]

and DNS-multicast [RFC6762] may be used for a limited number of local services, the propagation of renewed mappings need to propagate the hierarchy of DNS servers in the system. Even, e.g., CDN-local, mapping updates do not happen frequently although concrete numbers depend on the various providers using those systems. With that, even if resolving the domain name frequently, flushing the cache at the client to avoid using the stale information and ignoring any possible rate limitation of client request in first hop DNS resolver, the mapping update may not propagate to the client before seconds or even longer have passed. For many of our use cases, such as for the multi-domain/homed use case in Section 3.3, the micro-service based applications in Section 3.4 or the video-related ones in Section 3.5 and 3.6 in [I-D.mendes-rtgwg-rosa-use-cases], this level of dynamicity does not suffice.

4. Supporting arbitrary application identifiers: As the name suggests, domain names are the primary naming scheme for the DNS. Any other application identifier scheme would utilize its own resolution scheme, possibly mimicing the workings of the DNS. This requires a per-application support for its own identifier scheme, such as done in the QUICr [I-D.jennings-moq-quicr-arch] work discussed in our use case of Section 3.5 in [I-D.mendes-rtgwg-rosa-use-cases]. This is unlike ROSA, which aims at supporting application identifiers rather than a one size fits all scheme only. With that, ROSA also provides the ability to support own naming schemes that may want to explicitly avoid the use of a centrally governed namespace as well as the use of a central name resolution scheme that may reveal service usage patterns to the resolver system itself, as discussed in our use case of Section 3.10 in [I-D.mendes-rtgwg-rosa-use-cases].

4.2. Compute-aware Traffic Steering (CATS)

The Compute-aware Traffic Steering (CATS) WG is a newly established working group in the IETF, which aims at supporting the selection of one of possibly many service instances for a particular service. This similarity in objectives makes us draw out the main concepts and gaps to the objectives for ROSA in the following.

4.2.1. Technology Overview

Let us provide a brief overview of LISP and its main concepts - for more detail, we refer to, e.g., [I-D.ldbc-cats-framework].

CATS proposes compute-aware decisions in sending traffic between a client and a set of possible egress sites or directly Internet-connected service hosts. For this, CATS introduces the CS-ID as the

CATS service identifier, which is mapped onto the CB-ID as the CATS binding identifier. The exact nature of those identifiers is still work-in-progress with proposals currently being presented to the CATS WG.

CATS proposes to use an ingress-egress tunneling approach, where ingress CATS routers use metrics to decide upon the CB-ID to be used for an incoming request to a CS-ID. The tunneling method is currently still under discussion with SRv6, MPLS and other technologies being considered.

As the name suggests, the basis for the aforementioned selection at the ingress CATS router are compute metrics that are being distributed to the ingress CATS routers through suitable methods, which are still under investigation together with the nature and extend of the metrics themselves.

To support the steering of longer service transactions, CATS proposes a CATS traffic classifier component, which associates several packets to such longer service transaction to ensure the steering of those packets to the same selection made for the initial packet.

4.2.2. Relation to ROSA

CATS proposes a similar anycast type of addressing and as well as separation of service from routing identifier as done by ROSA. Furthermore, the ingress CATS router performs a traffic steering decision among the set of possible service instances albeit with a focus on such decisions to be compute-aware.

4.2.3. Gaps

There are number of key differences and gaps to the desired properties of a ROSA system:

1. Focus on compute-awareness: In contrast to CATS (considering the arch and solutions currently discussed), ROSA does not specifically consider compute-awareness. This does not prevent using the CATS steering framework (and later solutions) to be used outside compute-aware metrics. For this, the extensibility to general service-specific metrics in the future metric distribution solutions for CATS will need to be studied for that purpose.

2. Tunneling all traffic: As mentioned above, CATS proposes an ingress-egress tunneling of ALL traffic, which is contrary to ROSA which merely initially selects the service instance through the ROSA overlay, while all following packets will be directly sent to the service instance IP address, thus not using the ROSA overlay anymore and not tunneling any traffic

either; it thus siginificantly more lightweight on the ROSA overlay.

3. Network- vs endpoint-controlled affinity: The aforementioned tunneling of all traffic through the CATS overlay makes it necessary to support affinity through functionality provided by the CATS overlay network. Specifically, [I-D.ldbc-cats-framework] proposes use of the CATS Traffic Classifier for this purpose, interfacing with the ingress CATS router to convey the suitable information for detecting those packets belonging to a previously tunneled CATS flow. ROSA instead proposes a purely endpoint-based method where the initiation of another endpoint selection message signals the beginning of a new transaction, possibly being sent to a different choice of service instance than the previous one. This removes not just state management from the network but also the need for explicitly supporting future types of transactions and their associated transport/network-level identification.

4. Dynamicity of selection changes: CATS does foresee changes in service instance selections based on the metrics being distributed to the ingress CATS router via the CATS Service Metric Agent (C-SMA) and he CATS Network Metric Agent (C-NMA). Currently, necessary routing protocols (and their possible use and/or extension) are actively discussed. ROSA does foresee see use of ingress-based scheduling of selection messages, not requiring frequent metric updates to the ingress point and therefore allowing for higher frequencies of changes, such as prescribed in the AR/VR use case in Section 3.6 of [I-D.mendes-rtgwg-rosa-use-cases]. Relying on routing-based approaches to metric changes makes the realization of such high frequency changes difficult or impossible due to the associated routing overhead and latency for propagation of updated metrics.

5. Adherence to underlay routing policy: ROSA performs endpoint selection (from a set of possible choices), either routing- or ingress-based, where any subsequent message(s) that follows the selection message will traverse the network provider(s) defined IPv6 path. Here we see ROSA more aligned (conceptually) with existing SBR methods, such as DNS+IP, where selection precedes the subsequent network provider policy defined data transfer. CATS, instead, is currently looking into methods for active path (selection) control for ALL tunnelled CATS messages, e.g., using SRv6 or MPLS. However, a purely IP-in-IP tunneling at the ingress CATS router would align CATS with ROSA in this respect. Conversely, ROSA may provide such overlay path steering methods

by providing SRv6 path information as the result of the endpoint selection message.

4.3. Locator-ID Separation Protocol (LISP)

The Locator-ID Separation Protocol (LISP) WG has been in existence for many years, aiming at separation endpoint identifiers (called EIDs) and routing locators (called RLOCs) for better scalability of adjusting to changes in their relation. This similarity in focusing on in-band dynamic assignments of EIDs to RLOCs positions LISP as a possible technology to address the pain points identified in our use case draft. Let us draw out the LISP concepts and the gaps to ROSA objectives in the following.

4.3.1. Technology Overview

Let us provide a brief overview of LISP and its main concepts - for more detail, we refer to, e.g., [RFC9299].

LISP introduces two namespaces, separating endpoint identifiers (EID) from routing locator (RLOC) for a device realizing the service or resource represented by the EID. The EID may be determined from mapping services such as the DNS, resolved from other application-specific identifiers (such as a URL).

Endpoints communicate through their EIDs, sent domain-locally through an intra-domain routing protocol either to a locally present EID or to the ingress tunnel router (ITR) of their local domain. The ITR in turn consults a mapping service [RFC9301] to resolve the EID to an RLOC of an egress tunnel router (ETR), to which the incoming request is then sent, while the ETR domain-locally forwards the packet to the destination EID. LISP uses UDP for ITR-ETR tunnelling as well as for access the mapping service.

Mapping service resolutions are usually cached at the ITR after initially being resolved due to an incoming packet request. In addition to this DNS-like pull operation, a pub/sub extension may proactively pull EID->RLOC mappings from the mapping service (e.g., for planned handovers) or update previously resolved mappings in the future.

4.3.2. Relation to ROSA

One could position an EID as a service address in ROSA, where the mapping process in the ITR resembles the endpoint selection. The proactive pub/sub mapping resolution would allow for changing RLOC assignments and thus direct EID requests to other ETRs.

4.3.3. Gaps

There are number of key differences and gaps to the desired properties of a ROSA system:

1. Resolution latency: In its explicit resolution mode, as described in [RFC9299], LISP is to experience similar latencies as in other resolution systems. Unlike DNS, the resolution is done, however, at the ITR, thus not requiring explicit resolution at the client with subsequent data transfer, therefore reducing the needed client access link operations. Results from [LISPmon2017] show early deployment insights for LISP, with resolvers replying to EID mappings between 400ms and 1400ms. However, pub/sub extensions to the mapping service [RFC9301] also allow for reducing those latencies, e.g., proactively placing EID mappings in ITRs in anticipation of future resolution requests, although this is subject to suitable management and planning methods to exist. Equally, for EID mapping updates to previously resolved EID mappings, the pub/sub extensions may reduce the latency of future resolution requests. However, scenarios such as those outlined in Section 3.5 and 3.6 of [I-D.mendes-rtgwg-rosa-use-cases] are difficult to realize even with those methods since the frequency of update for per transaction changes of EID mappings may be achieve through notification updates to EID mappings due to the network latencies experienced for the traversal of the EID mapping update to the respective ITR(s). We can therefore expect that the support for high dynamicity of service instance changes is likely less in LISP than what is required in some of our use cases, thus limiting required the SBR capabilities, while the scheduled mode of service instance selection in ROSA is expected to allow per transaction changes.

2. Lack of affinity support: LISP does not have a notion of affinity to EID selections made for a service transaction, meaning that an EID->RLOC mapping may change independent from any notion of a service transaction. This is in contrast to ROSA, where affinity is signalled directly by the originating endpoint through issuing a new endpoint selection message, possibly resulting in a different service instance being selected, with which the endpoint continues to communicate through the transaction. Through this, any client and/or flow-specific state is avoided to exist in the ROSA network elements.

3. Tunnelling all traffic: LISP is a network-level overlay to separate the EID from RLOCs. As a consequence, ALL traffic from an originating endpoint to an EID must be tunnelled via the ITR to the resolved ETR. This is unlike the simpler problem of

identifying a service instance in ROSA, followed by any subsequent traffic (of a transaction) being sent directly via the underlying (possibly multi-domain) IP networks, similar to explicit resolution SBR solutions like DNS. This simplicity is reflected in less load on the ROSA elements (since only endpoint selection messages need treatment while no direct endpoint-instance message will traverse the ROSA element), while also removing any tunnelling overhead.

4. Deployment as network-independent SBR overlay: LISP extends the network-level routing capabilities through its separation of address spaces. It does so, however, by requiring the ITR as a border gateway to be part of the domain-local network deployment, turning the otherwise 'LISP unaware' network into a 'LISP-aware' one, consequently allowing LISP endpoints in this domain to communicate with other LISP-aware domains. It thus requires the participation of the local domain in the overall LISP deployment, still allowing for gradual deployment (through traversing non-LISP-aware domains through tunnelling) but nonetheless requiring the endpoint-local domain to be LISP-enabled for using LISP-enabled services. Proxy-xTRs allow, however, for the internetworking of LISP-unaware with LISP-aware sites but still require involvement of the provider edge network and need careful deployment considerations on EID announcement (to the global routing system) and placement in the network. This is unlike ROSA, which is positioned as a L3.5 overlay, thus not requiring that endpoint-connected domains to participate in the ROSA service. From a local network perspective, a client sends an endpoint selection message to what looks like an IP endpoint to the local domain. Those endpoint selection messages are routed as true overlay messages, until arriving at an IP-enabled endpoint that represents the selected service instance, followed by direct client-instance exchanges for subsequent messages for the service transaction. Thus, the burden of deployment in local networks or the need for proxies does not exist here.

5. Service specificity of EID selections: The current methods of selecting one of possible several EID->RLOC mappings foresee a priority and weighted mechanism, where those priorities and weights are driven by the announcer of the EID mapping, with a direct consequence on how traffic is being steered through the network. Thus, the objective of those mapping policies are more focused on traffic distribution although RLOC priorities could also be driven by service-specific policies. This is unlike the explicit service specificity of the foreseen ROSA overlay routing decision, where either a routed or scheduled endpoint selection process is realized to disconnect the choice of service instance selection from the network-level policy of

steering traffic to it, as linked to the routing locator of the service instance.

4.4. Application-Layer Traffic Optimization (ALTO)

ALTO, as defined in [RFC7285], provides the ability to select suitable application-level servers for a client requesting it. It is thus seemingly aligned with the ROSA anycast problem but there are, however, very fundamental differences when looking closer:

4.4.1. Technology Overview

ALTO follows other SBR methods in employing an explicit server discovery step, defined in [RFC7286], thus conceptually aligning with methods like DNS in that it employs an off-path method.

ALTO also follows more of a recommendation model, where the final decision is being made by the ALTO client, which of the possible choices to utilize in the data transfer, while ROSA advocates a ROSA overlay driven decision.

Moreover, ALTO operates at the application level, currently supporting HTTP/1, while ROSA advocates the use of any application (and transport) protocol similar to using the DNS for resolution.

ALTO provides insights into server selection criteria through metric work, as outlined in [RFC9274] [RFC9241][RFC8895]; work that is already considered as input to the CATS WG. This consideration equally applies to ROSA where metrics as well as metric distribution are not in scope.

4.4.2. Relation to ROSA

Similar to the DNS, detailed in Section 4.1, ALTO provides an explicit resolution step for selecting HTTP/1-based service instances from a set of available servers. It thus provides a solution for an anycast selection albeit limited to HTTP/1-based services. It also allows for service-specific selection of the final server to be used through a recommendation model, i.e., providing choices of suitable servers to the client, which ultimately selects the server. With this, it differs from the DNS model, where the DNS resolver makes the ultimate selection.

4.4.3. Gaps

There are number of key differences and gaps to the desired properties of a ROSA system. Several of those gaps are similar to those that have already been identified in Section 4.1.3 and also thus presented only briefly again here:

1. Resolution latency: Similar to other explicit resolution solutions, ALTO experiences a discovery latency through the procedures defined in [RFC7285], leading to similar issues outlined already for the DNS.

2. Acting on stale information: Due to the explicit resolution, the client, in re-using a previous choice, may in fact act on stale information in that the previously used server does not represent the 'best' choice anymore. Only frequent repetition of the discovery step would avoid this, with similar issues than those outlined for the DNS.

3. Support dynamic resolution changes: ALTO defines methods for cost-based selection of (ALTO) servers [RFC9274] as well as advertising capabilities [RFC9241] and sending server events impacting the selection [RFC8895]. However, apart from the latencies involved in updating this information for a renewed and thus dynamic resolution result, such renewed result can only be considered in a renewed resolution step, leading back the latency incurred for doing so; both of which combined does not suffice in terms of dynamicity, e.g., in the video-related use cases of Section 3.5 and 3.6 as well as for the mobile application scenario in Section 3.4 of [I-D.mendes-rtgwg-rosa-use-cases].

4. Support for arbitrary application identifiers (and protocols): As mentioned before, ALTO supports HTTP/1 only, thus limiting both application identifiers and protocols to the specific HTTP-based file sharing, media delivery and real-time comms scenarios that are outlined in the ALTO problem statement [RFC5693], thus providing no support for use cases outside the use of HTTP/1.

5. Multi-domain operation: Before the service-level communication commences, an ALTO client discovers a suitable ALTO server, which in turn provides guidance on the possible servers (for a particular service) that may suit the client requirements, provided as a recommendation to the ALTO client for its ultimate choosing of the server. As outlined in [RFC7286], the discovery of the ALTO server is domain-local, while explicit procedures as defined in [RFC8686] are required for discovering an ALTO server beyond the current domain. As outlined in the appendix A of [RFC8686], a possibly multi-domain ALTO deployment would require steps for discovering (and using) other ALTO servers so as to enrich the information available to the locally discovered ALTO server, much akin to the working of the DNS. The approach taken by ROSA is that of an overlay, employing routing-based methods to support those services

advertised to it (akin to all those services advertised to the overall ALTO system), while interconnecting to other ROSA domains and the wider Internet through an explicit gateway; a capability missing in ALTO.

4.5. Technologies related to SBR

Unlike the solutions in the previous sections, which provide capabilities to address service-based routing overall, the works in the next subsections relate to the SBR problem but often only in parts, which may still be relevant to the wider discussion of identifying works that may feed into the toolbox for ROSA solutions.

Most of the items on this list were suggested throughout discussions with community members and they aim at answering their questions on the relation to ROSA. As such, the list here may or may not increase in the future.

4.5.1. Service Function Chaining (SFC)

SFC as defined in [RFC7665] allows for chaining the execution of services at L2 or L3 level, targeting scenarios such as carrier-grade NAT and others. The work in [RFC8677] extends the chaining onto the name level, using service names to identify the individual services of the chain, even allowing combinations of name and L2/L3-based chains. However, [RFC8677] is tied into a realization of the SFF (service function forwarder) using a path-based forwarding approach, thus still relying on an explicit resolution process and therefore experiencing similar latency and dynamicity issues as DNS, ALTO, and LISP. The ROSA architecture framework draft includes an early discussion on how to possible realize name-based SFC without the need for such explicit resolution, extending the basic functionality of ROSA to invoke a single chain service.

4.5.2. Multiplexed Application Substrate over QUIC Encryption (MASQUE)

The work in the MASQUE WG aims at developing techniques for stream- or datagram-based flow multiplexing in a single HTTP connection. For this, the notion of a 'proxy' [I-D.schinazi-masque-proxy] is proposed together with CONNECT-UDP and CONNECT-IP primitives to enable this multiplexing. Typical use cases are tunnelling for increased privacy or additional encryption. Although QUIC is assumed as the underlying transport protocol, the WG will consider the working of its primitives over TCP.

We can foresee the linkage to the proposed ROSA work in utilizing MASQUE primitives for the in-band signalling of resolution request, utilizing the CONNECT-IP primitive. This effectively tunnels the ROSA overlay over MASQUE, possibly improving on deployability. One key aspect to consider, however, is the support for affinity, i.e., only

utilizing the MASQUE proxy for initial endpoint selection requests, then 'transferring' the client-endpoint relation onto a direct relation, thus removing the proxy from the middle of the connection for performance improvements and to adhere to the initial routing policies defined for reaching the locator of the selection service instance.

4.5.3. Time-Variant Routing (TVR)

The work in the newly established TVR WG addresses the problem of scheduled, thus predictable changes in routing state within the network. It plans on utilizing the exposure of agenda information to feed into the routing protocols for accommodating such predictable changes.

We can foresee two key linkages to the proposed ROSA work

1. The use of agenda information not just for maintaining route but possibly also endpoint availability information, which in turn may feed into the endpoint selection message handling in ROSA.

2. The use of a TVR solution as ROSA overlay routing solutions where the forwarding of ROSA messages (i.e., the endpoint selection message), may underlie scheduled and thus predictable changes; this could even be the case in the use cases currently identified for TVR (e.g., satellite, mobile devices etc) where those use cases may experience an anycast semantic for the endpoint selection.

4.5.4. Source Packet Routing in Networking (SPRING)

Source routing solutions, such as developed in the SPRING WG, allow for influencing the path across which a packet may traverse to a final destination. Unlike ROSA, the destination selection itself is not within scope of such consideration, thus SPRING and similar work may complement the endpoint selection process of ROSA in that it provides tools for further determining the path over which a packet is sent.

5. Requirements

The following requirements for a routing on service addresses (ROSA) solution (referred to as 'solution' for short) have been identified from the analysis in the previous section of the use cases provided in [I-D.mendes-rtgwg-rosa-use-cases].

One commonality of all use cases is the communication with a 'service', realized at one or more network locations as equivalent 'service instances'. Associating the service to an 'owner' is key to

avoid services being announced by fake entities, thus misdirecting the client's traffic, while obfuscating the purpose of communication (e.g., leaked through the specific name of a service) but also any possible policy to select one over another service instance may want to be kept private; this is likely the case across all of our use cases. Hence, any solution

REQ1: MUST provide means to associate service instances with a single service address.

- **(a)** MUST provide secure association of service address to service owner.
- **(b)** SHOULD provide means to obfuscate the purpose of communication to intermediary network elements.
- **(c)** MAY provide means to obfuscate the constraint parameters used for selecting specific service instances.

Across all our use cases, the knowledge of where service instances (realizing specific services) reside within the network, i.e., possibly at different network locations, is crucial for the communication to happen, at least for the ROSA domain with which the service has an association with. Such knowledge may be created by a service management platform, e.g., as part of the overall service deployment, and thus may not be initiated by the deployed service instance itself, such as in the example of mobile distributed applications of Section 3.4 in [I-D.mendes-rtgwg-rosa-use-cases]. Furthermore, service deployment may be delegated to service or CDN platforms, e.g., in the CDN, AR/VR and video distribution examples of [I-D.mendes-rtgwg-rosa-use-cases], albeit with linkages needed to the service routing capabilities of ROSA. Crucially, however, is that a solution ought to use proactive pushing of suitable reachability information to service instances into the ROSA system, i.e., pursuing a routing-based approach, allowing for faster availability of information to make suitable decisions on which service instance to choose among those available. Hence, any solution

REQ2: MUST provide means to announce route(s) to specific instances realizing a specific service address, thus enabling service equivalence for this set of service instances.

(a) MUST provide scalable means to route announcements.

(b) MUST announce routes within a ROSA domain.

(c) SHOULD provide means to delegate route announcement.

(d) SHOULD provide means to announce routes at other than the network attachment point realizing the announced service address.

(e) MUST allow for removing service instances that are intermittently available, i.e., revoking their service announcement after a defined timeframe.

A client application may not just invoke services within a single ROSA domain. While associating with different ROSA domain may be possible, clients may simply invoke services through their existing ROSA domain, e.g., for utilizing helper services in examples like distributed mobile applications (Section 3.4 in [I-D.mendes-rtgwg-rosa-use-cases]), expecting the service transaction to be realized regardless. The same goes for invoking services that may reside in the public Internet, without requiring an explicit awareness of the client to which ROSA domain (or the public Internet) to direct the invocation. Thus, any solution

REQ3: MUST provide means to interconnect ROSA islands.

(a) MUST allow for announcing services across ROSA domains.

(b) MUST allow for announcing services outside ROSA domains.

Use cases like distributed mobile applications (Section 3.4 in [I-D.mendes-rtgwg-rosa-use-cases]) but also video delivery ones such as for replicated chunk retrieval or AR/VR (Sections 3.5 and 3.6 in [I-D.mendes-rtgwg-rosa-use-cases], respectively) or the selection of an appropriate UPF (user plane functions) within a cellular sub-system (Section 3.2 in [I-D.mendes-rtgwg-rosa-use-cases]), may want to constrain the selection of 'suitable' service instances through service-specific constraints, such as the computing load (on the deployed service instances or their host platforms), service-level latency, but also, e.g., HW or SW, capabilities. This may also be the case for multi-homed deployments (see Section 3.3 in [I-D.mendes-rtgwg-rosa-use-cases]), where constraints on the multi-connectivity of the service instance may constrain the suitability for specific clients. Thus any solution

REQ4: Solution MUST provide constraint-based routing capability.

(a)

MUST provide means to announce routing constraints associated with specific service instances and their realizing networking, computing and storaged resources.

(b) SHOULD allow for providing constraints in the service (address) announcement.

The work in [OnOff2022] has shown the potential gains in making runtime decisions for every incoming service transaction, where transaction lengths may be as small as single (application-level) requests. For use cases such as for replicated chunk retrieval (Section 3.5 in [I-D.mendes-rtgwg-rosa-use-cases]) or AR/VR (Section 3.6 in [I-D.mendes-rtgwg-rosa-use-cases]), this may lead to significant smoothening of the request completion latency, i.e., reducing the latency variance, thus enabling a better, smoother experience at the client. However, the specific mechanism may vary and, more importantly, may be highly service-specific, with solutions such as [CArDS2022] providing a simple weighted round robin, while other methods may rely on regular (service) metric reporting. Thus any solution

REQ5: MUST provide an instance selection at ROSA domain ingress nodes only.

(a) MUST allow for signalling selection mechanism and necessary input parameters for selection to the ROSA domain ingress nodes.

Explicit resolution steps, such as those in DNS, GSLB, or Alto, suffer from the need for an explicit control plane exchange. This causes additional latency before the data transfer to the chosen service instance may start. In-band data, i.e., the inclusion of application-level data in the control messages, is not supported due to the layering of such solutions at the application level itself. It is desirable, however, to already allow for the exchange of application data, including that needed for establishing secure connections, in the process that determines the most suitable service instance to further reduce any latency for completing a given application-level service transaction. Thus any solution

REQ6: MUST provide an in-band data transfer capability in the process of determining the suitable service instance for any following data transfer within the same service transaction.

While video delivery use cases like replicated chunk retrieval (Section 3.5 in [I-D.mendes-rtgwg-rosa-use-cases]) or AR/VR (Section 3.6 in [I-D.mendes-rtgwg-rosa-use-cases]) may exhibit short lived transactions of just one (service-level) request, due to the replicated nature of the video content in each service instance,

service transactions may last many requests after the initial one has been sent. Ephemeral state may be created during this transaction, which would require that a change of the (initial) service instance during a transaction would share such ephemeral state with any new service instance being used. While service platforms, like K8S, provide such ability through 'shared data layer' capabilities, those are often limited to single site deployments. Any support across sites would incur additional costs or even possibly latencies for such state sharing, thus often leading to completing an ongoing service transaction with the service instance that has been originally been used (note that a service instance in ROSA may use internal methods for serving incoming requests across which state sharing would be applied - from a ROSA perspective, however, only one service instance is being used). We call the capability to retain an initial selection of a service instance for the length of a service transaction 'affinity'. Thus, any solution

REQ7: MUST adhere to the affinity towards the service instance chosen in the initial service request of the service transaction, thus directing all subsequent service transaction requests to the same instance.

All of our use cases are likely being deployed over existing network infrastructure, which makes a consideration to use its existing solutions in any realization of ROSA very important. Specifically, any solution

REQ8: Solution SHOULD use IPv6 for the routing and forwarding of service and affinity requests.

(a) Solution MAY use IPv4 for the routing and forwarding of service and affinity requests.

Most of our use cases, specifically on distributed mobile applications (Section 3.4 in [I-D.mendes-rtgwg-rosa-use-cases]) but also our video delivery examples, may be realized in inherently mobile settings with clients moving about for their experience. While mobile IP solutions exist, the service initialization in ROSA needs to be equally supported in order to allow for invoking ROSA services on the move. Thus, any solution

REQ9: SHOULD support in-request mobility for a ROSA client.

Mobility of clients, but also varying loads in scenarios of no client mobility, may also lead to situations where moving on ongoing service transaction to another service instance may be beneficial, termed 'transaction mobility'. In other words, service instances may be replaced mid-transaction, in order to ensure the service level agreement. This may happen if, for instance, the local node where the

service instance was initially installed is running out of resources, or its accessibility is reduced (which be periodically). Thus, any solution

REQ10: SHOULD support transaction mobility, i.e., changing service instances during an ongoing service transaction.

With most service transactions likely being encrypted for privacy and security reasons, supporting the appropriate transport layer methods is crucial in all our scenarios in [I-D.mendes-rtgwg-rosa-use-cases]. While work in [OnOff2022] has shown that small service transactions in scenarios like replicated chunk retrieval (Section 3.5 in [I-D.mendes-rtgwg-rosa-use-cases]) or AR/VR (Section 3.6 in [I-D.mendes-rtgwg-rosa-use-cases]) may be beneficial for significantly reducing the service-level latency, the challenge lies in initiating suitable transport layer security associations with frequently changing service instances. Pre-shared certificates may address this to allow for 0-RTT handshakes being realized but come with well-known forward secrecy problems. Thus, any solution

REQ11: SHOULD support TLS 0-RTT handshakes without the need for pre-shared certificates.

We envision the ROSA layer in ROSA endpoints to be transparently integrated in the operation of transport protocols, and thus applications, by provuding suitable interfaces to accessing the ROSA services of a specific ROSA domain. Thus, any solution

REQ12: SHOULD be transparent to applications in order to ensure a smooth deployment.

6. Benefits from Addressing the SBR Problem

We expect the following benefits to be realized through providing a solution to the problem statement presented in [I-D.mendes-rtgwg-rosa-use-cases]:

* Remove explicit resolution latency: Current service-based routing utilises a an explicit resolution step with explicit off-path operations before being able to utilise a specific service, thus incurring an additional latency for requesting the resolution and receiving its result. We aim at significantly reducing, even removing this latency. The work in [OnOff2022] outlines the possible impact of such reduction, while also evaluating the capabilities enabled by a flexible (small affinity) traffic steering under the constraint of a given latency budget that is now been enabled by the smaller endpoint selection latency.

* Dynamicity: Decisions to select one out of possibly many service instance can be highly dynamic, done per service transaction,

including for single service requests even. This is enabled by the move from an explicit off-path resolution step to an in-band mapping of a service address to its realizing service instance. Such dynamicity aims at improving transaction completion latency and variance, balancing load across service instances, as well as possibly deal with temporary network conditions. The work in [OnOff2022] evaluates the impact of performing traffic steering decisions through such in-based rather than explicit resolution approaches.

*Service-specificity: The constraints for selecting a suitable service instance should not be limited to network metrics like delay or bandwidth. Instead, services should be able to define service-specific constraints, allowing for either multi-optimality routing or realising request-level and possibly compute-aware request scheduling for selecting one of possibly several service endpoints. The mechanism in [CArDS2022] outlines an example for such steering decisions, taking into account service-specific compute information. However, to avoid embedding full path information into the service-based routing itself, the consideration of service-specific constraints should be limited to the selection of service instances, while the forwarding of transaction data (in the form of subsequent affinity requests) solely follows the routing policies defined by the underlay network, similar to the workings of the DNS today.

*Avoiding in-network state: Mimicking the workings of the DNS, ROSA seeks to keep any transaction state management entirely at the endpoint, i.e., it is the endpoint that explicitly invokes the (now in-band) endpoint selection, followed by end-to-end data transfer throughout the transaction. This avoids the need for any in-network or edge component to manage client- and flow/ transaction- specific state, such as envisioned in the CATS architecture framework [I-D.ldbc-cats-framework] when relying on explicit tunnel endpoints. This creates a deployment dependency only for the endpoint selection itself, much like when using the existing DNS, while any subsequent data transfer (within the transaction) runs directly over the (possibly many) IP networks that the IP packets will traverse, likely easing deployment of any ROSA solution.

*Efficiently support higher degree of service distribution: Typical application or also L4-level solutions, such as GSLB, QUIC-based indirection, and others, lead effectively to egress hopping when performed in a multi-site deployment scenario in that the client request will be routed first to an egress as defined either through the DNS resolution or the indirection through a central server, from which the request is now resolved or redirected to the most appropriate DC site. In deployments with a high degree of

distribution across many (e.g., smaller edge computing) sites, this leads to inefficiencies through path stretch and additional signalling that will increase the request completion time. Instead, direct or on-path solutions such as ROSA are expected to lead to a more direct traffic towards the site where the service will eventually be executed, while also allowing for application data to be already carried as part of the service instance selection process, thus keeping the request completion time close to its optimum in respect to the best site being used for execution of the request.

*Bring application namespace closer to communication relations: Reid et al [Namespaces2022] outline insights into the aspects and pain points experienced when deploying existing intra-DC service platforms in multi-site settings, i.e., networked over the Internet. The main takeaway in is the lacking protocol support for routing requests of microservices that would allow for mapping application onto network address spaces without the need for explicitly managed mapping and gateway services. While this results in management overhead and thus costs, efficiency of such additional mapping and gateway services is also seen as a hinderance in scenarios with highly dynamic relationships between distributed microservices, an observation aligned with the findings in [OnOff2022]. The use cases presented in [I-D.mendes-rtgwg-rosa-use-cases], among others, exhibit the degrees of distribution in which relationship management (through explicit mapping and/or gatewaying) may become complex and a possible hinderance for service deployment and suitable performance.

7. Conclusions

This draft provided a gap analysis of existing methods for service-based routing in relation to the issues and pain points identified in [I-D.mendes-rtgwg-rosa-use-cases].

Furthermore, we outlined requirements to fill those gaps in possible realizations, a first of which is being described in a companion document as the ROSA architecture.

8. Security Considerations

To facilitate the decision between service information (i.e., the service address) and the IP locator of the selected service instance, information needs to be provided to the ROSA service address routers. This is similar to the process of resolving domain names to IP locators in today's solutions, such as the DNS. Similar to the latter techniques, the preservation of privacy in terms of which services the initiating client is communicating with, needs to be preserved

against the traversing underlay networks. For this, suitable encryption of sensitive information needs to be provided as an option. Furthermore, we assume that the choice of ROSA overlay to use for the service to locator mapping is similar to that of choosing the client-facing DNS server, thus we assume it being configurable by the client, including to fall back using the DNS for those cases where services may be announced to ROSA methods and DNS-like solutions alike.

9. IANA Considerations

This draft does not request any IANA action.

10. Acknowledgements

Many thanks go to Ben Schwartz, Luigi Iannone, Mohamed Boucadair, Tommy Pauly, Joel Halpern, Daniel Huang, and Russ White for their comments to the text to clarify several aspects of the motivation for and technical details of ROSA.

11. Informative References

[CArDS2022] Khandaker, K., Trossen, D., Khalili, R., Despotovic, Z., Hecker, A., and G. Carle, "CArDS:Dealing a New Hand in Reducing Service Request Completion Times", Paper IFIP Networking, 2022.

[GSLB] "What is GSLB?", Technical Report Efficient IP, 2022, <https://www.efficientip.com/what-is-gslb/>.

[I-D.jennings-moq-quicr-arch] Jennings, C. F. and S. Nandakumar, "QuicR - Media Delivery Protocol over QUIC", Work in Progress, Internet-Draft, draft-jennings-moq-quicr-arch-01, 11 July 2022, <https://datatracker.ietf.org/doc/html/draft-jennings-moq-quicr-arch-01>.

[I-D.ldbc-cats-framework] Li, C., Du, Z., Boucadair, M., Contreras, L. M., Drake, J., Huang, D., and G. S. Mishra, "A Framework for Computing-Aware Traffic Steering (CATS)", Work in Progress, Internet-Draft, draft-ldbc-cats-framework-02, 22 June 2023, <https://datatracker.ietf.org/doc/html/draft-ldbc-cats-framework-02>.

[I-D.mendes-rtgwg-rosa-use-cases] Mendes, P., Finkhäuser, J., Contreras, L. M., and D. Trossen, "Use Cases and Problem Statement for Routing on Service Addresses", Work in Progress, Internet-Draft, draft-mendes-rtgwg-rosa-use-cases-00, 26 June 2023, <https://datatracker.ietf.org/doc/html/draft-mendes-rtgwg-rosa-use-cases-00>.

[I-D.schinazi-masque-proxy]
Schinazi, D., "The MASQUE Proxy", Work in Progress, Internet-Draft, draft-schinazi-masque-proxy-00, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-schinazi-masque-proxy-00>.

[I-D.yao-cats-gap-reqs] Yao, K., Jiang, T., Eardley, P., Trossen, D., Li, C., and D. Huang, "Computing-Aware Traffic Steering (CATS) Gap Analysis and Requirements", Work in Progress, Internet-Draft, draft-yao-cats-gap-reqs-00, 3 March 2023, <https://datatracker.ietf.org/doc/html/draft-yao-cats-gap-reqs-00>.

[LISPmon2017] Li, Y., Iannone, L., and D. Saucez, "LISP-Views: Monitoring LISP at Large Scale", Paper 29th International Teletraffic Congress (ITC 29), 2017.

[Namespaces2022] Reid, A., Eardley, P., and D. Kutscher, "Namespaces, Security, and Network Addresses", Paper ACM SIGCOMM workshop on Future of Internet Routing and Addressing (FIRA), 2022.

[OnOff2022] Khandaker, K., Trossen, D., Yang, J., Despotovic, Z., and G. Carle, "On-path vs Off-path Traffic Steering, That Is The Question", Paper ACM SIGCOMM workshop on Future of Internet Routing and Addressing (FIRA), 2022.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <https://www.rfc-editor.org/info/rfc5693>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <https://www.rfc-editor.org/info/rfc7285>.

[RFC7286] Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., and H. Song, "Application-Layer Traffic Optimization (ALTO) Server Discovery", RFC 7286, DOI 10.17487/RFC7286, November 2014, <https://www.rfc-editor.org/info/rfc7286>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC8677] Trossen, D., Purkayastha, D., and A. Rahman, "Name-Based Service Function Forwarder (nSFF) Component within a Service Function Chaining (SFC) Framework", RFC 8677, DOI 10.17487/RFC8677, November 2019, <https://www.rfc-editor.org/info/rfc8677>.

[RFC8686] Kiesel, S. and M. Stiemerling, "Application-Layer Traffic Optimization (ALTO) Cross-Domain Server Discovery", RFC 8686, DOI 10.17487/RFC8686, February 2020, <https://www.rfc-editor.org/info/rfc8686>.

[RFC8895] Roome, W. and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)", RFC 8895, DOI 10.17487/RFC8895, November 2020, <https://www.rfc-editor.org/info/rfc8895>.

[RFC9241] Seedorf, J., Yang, Y., Ma, K., Peterson, J., and J. Zhang, "Content Delivery Network Interconnection (CDNI) Footprint and Capabilities Advertisement Using Application-Layer Traffic Optimization (ALTO)", RFC 9241, DOI 10.17487/RFC9241, July 2022, <https://www.rfc-editor.org/info/rfc9241>.

[RFC9274] Boucadair, M. and Q. Wu, "A Cost Mode Registry for the Application-Layer Traffic Optimization (ALTO) Protocol", RFC 9274, DOI 10.17487/RFC9274, July 2022, <https://www.rfc-editor.org/info/rfc9274>.

[RFC9299] Cabellos, A. and D. Saucez, Ed., "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", RFC 9299, DOI 10.17487/RFC9299, October 2022, <https://www.rfc-editor.org/info/rfc9299>.

[RFC9301] Farinacci, D., Maino, F., Fuller, V., and A. Cabellos, Ed., "Locator/ID Separation Protocol (LISP) Control Plane", RFC 9301, DOI 10.17487/RFC9301, October 2022, <https://www.rfc-editor.org/info/rfc9301>.

Authors' Addresses

Luis M. Contreras
Telefonica
Ronda de la Comunicacion, s/n
Sur-3 building, 1st floor
28050 Madrid
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://lmcontreras.com/

Dirk Trossen
Huawei Technologies
80992 Munich
Germany

Email: dirk.trossen@huawei.com
URI: https://www.dirk-trossen.de

Jens Finkhaeuser
Interpeer gUG
86926 Greifenberg
Germany

Email: ietf@interpeer.io
URI: https://interpeer.io/

Paulo Mendes
Airbus
82024 Taufkirchen
Germany

Email: paulo.mendes@airbus.com
URI: http://www.airbus.com