Workgroup: Network Working Group
Internet-Draft: draft-trossen-rtgwg-rosa-00
Published: 24 October 2022
Intended Status: Standards Track
Expires: 27 April 2023
Authors: D. Trossen
Huawei Technologies
LM. Contreras
Telefonica

Routing on Service Addresses

Abstract

This document proposes a novel communication approach which reasons about WHAT is being communicated (and invoked) instead of WHO is communicating. Such approach is meant to transition away from locator-based addressing (and thus routing and forwarding) to an addressing scheme where the address semantics relate to services being invoked (e.g., for computational processes, and their generated information requests and responses).

The document introduces Routing on Service Addresses (ROSA), as a realization of what is referred to as 'service-based routing' (SBR). Such routing is designed to be constrained by service-specific parameters that go beyond load and latency, as in today's best effort or traffic engineering based routing, leading to an approach to steer traffic in a service-specific constraint-based manner.

Particularly, this document outlines sample ROSA use case scenarios, requirements for its design, and the ROSA system design itself.

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 27 April 2023.

Table of Contents

1. Introduction

The centralization of Internet services has been well observed, not just in IETF discussions [Huston2021] [I-D.nottingham-avoiding-internet-centralization], but also in other

efforts that aim to quantify the centralization, using methods such as the Herfindahl-Hirschman Index [HHI] or the Gini coefficient [Gini]. Dashboards of the Internet Society [ISOC2022] confirm the dominant role of CDNs in service delivery beyond just streaming services, both in centralization as well as resulting market inequality, which has been compounded through the global CV19 pandemic [CV19].

This centralization impacts the global Internet, as argued in [Huston2021], through largely replacing Internet transit with global private networks, providing optimized last mile access to services through an economy of scale that only data centres (point-of-presence) can provide. But it also runs counter the original Internet design as a peer-to-peer communication system, having replaced the destination end host through an intermediary, usually deployed in the nearest PoP.

The impact on routing can be seen in, e.g., [TIES2021], which goes as far as centralizing service requests into a single IP address behind which DC-internal mechanisms take over.

There is an inherent risk in such trend, not just at the economic level (in terms of market centralization and inequality) but also at the technological one since economic dominance may likely lead to skewing the technological enablers towards cementing the status quo that the current market represents. With it comes the danger that new use cases may be prevented in the light of the optimizations towards a centralized service provisioning capability.

Providing the backdrop to the design proposed in this document, [EI2021] proposes an Extensible Internet (EI) framework for architectural evolution atop today's Internet. Novel network services are realised within interconnected service nodes (SNs), thereby taking IP for granted, while deploying SNs within last mile providers (LMPs) or cloud providers (CPs).

The concept of limited domains [RFC8799] argues for a model of Internet technology development based on domain-specific behaviours and requirements, relying on the Internet for interconnection. The authors in [LDCU2021] show that this model has been driving innovation in the Internet since its very beginning, with well-known technologies resulting from it. ROSA aligns with the EI view of an architectural evolution through a shim layer atop IPv6. This positions ROSA as an architecture for peer-to-peer service communication in limited domains, with market opportunities for LMPs or CPs, while interconnecting via the Internet for wider reachability.

Evolving the IPv6 network layer has been part of its design from the very start. Key enabler here are Extension headers (EHs), which are part of the IPv6 specifications [RFC8200], with some observed problems, e.g., firewall traversal, in real-world deployments [SHIM2014]. Recent solutions, such as Segment routing (SR) [RFC8402], specifically SRv6 [RFC8986] build on this capability by establishing a shim layer overlay (of SR-enabled routers), utilizing an extension header to carry needed information for realizing the source routing capabilities.

In remainder of this document, we first introduce in Section 2 a terminology that provides the common language used throughout the remainder of the document. We then introduce use cases in Section 3 that drive the need for a routing on service address solution. We then outline in Section 4 the requirements for such solution before introducing its design in Section 5.

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. A composite service can be built by orchestrating a combination of monolithic services.

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 network at different network locations, thus providing service equivalence between those instances.

Service Address: An identifier for a specific service.

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 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.

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

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 Section 5.5.

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

3. Deployment and Use Case Scenarios

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 [Namespaces2022] 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].

In the following, we outline examples for use cases that exhibit the degrees of distribution in which relationship management (through explicit mapping and/or gatewaying) may become complex and a possible hinderance for service performance.

3.1. CDN Interconnect and Distribution

Video streaming has been revealed nowadays as the main contributing service to the traffic observed in operators' networks. Multiple stakeholders, including operators and third party content providers, have been deploying Content Distribution Networks (CDNs), formed by a number of cache nodes spread across the network with the purpose of serving certain regions or coverage areas. In such a deployment,

protection schemas are defined in order to ensure the delivery continuity even in the case of outages or starvation in cache nodes.

In addition to that, novel schemes of CDN interconnection [RFC6770] [SVA] are being defined allowing a given CDN to leverage the installed base of another CDN to complement its overall footprint.

As result, several caches are deployed in different Points of Presence in the network. Then for a given content requested by an end user, several of those caches could be candidate nodes for delivery. Currently, the choice of the cache node to serve the customer relies solely on the content provider logic, considering only a limited set of conditions to apply.

The performance can be improved by the consideration of further conditions in the decision on what cache node to be selected. Thus, the decision can depend of course on the requested content and the operational conditions of the cache itself, but also on the network status or any other valuable, often service-specific, semantic for reaching those nodes.

Furthermore, those decision points may be dynamic and could even change during the lifetime of the overall service, thus requiring to revisit decisions and therefore assignments to the most appropriate CDN node.

3.2. Distributed user planes for mobile and fixed access providing reachability to edge computing facilities

5G networks natively facilitate the decoupling of control and user plane. The User Plane Function (UPF) in 5G networks terminates the tunnels set carrying end user traffic permitting to route the end user traffic in the network towards its destination.

Several UPFs can be deployed in a distributed manner, not only for covering different access areas, but UPFs can also be distributed with the attempt of providing access to different services, linked with the idea of network slicing as means for tailored service differentiation. For instance, some UPFs could be deployed very close to the access for services requiring either low latency or very high bandwidth, while others could be deployed in a more centralized manner for requiring less service flows. Furthermore, multiple instances can be deployed for scaling purposes depending on the demand in a specific moment.

Similarly, to what happens in mobile access, fixed access solutions are proposing schemas of separation of control and user plane for BNG elements [I-D.wadhwa-rtgwg-bng-cups] [BBF]. From the deployment point of view, different instances can be deployed based on the coverage, the temporary demand, etc, as before.

As a complement to both mobile and fixed access scenarios, edge computing capabilities are expected to complement the deployments for hosting service and applications of different purposes, for both services internal to the operator or hosting of services from third parties.

In this situation, either for both selection of the specific user plane termination instance, or from that point on, selection of the service endpoint after the user plane function, it makes sense the introduction of mechanisms enabling selection choices based on service-specific semantics.

3.3. Multi-homed and multi-domain services

Corporate services usually present exact requirements in terms of availability and resiliency. This is why multi-homing is common in order to diversify the access to services external to the premises of the corporation, or for providing interconnectivity of corporate sites (and access to internal services such as databases, etc).

The diversity of providers implies to consider service situations in a multi-domain environment, because of the interaction with multiple administrative domains.

From the service perspective, it seems necessary to ensure a common understanding of the service expectations and objectives independently of the domain traversed or the domain providing such a service. Common semantics can facilitate the assurance of the service delivery and a quick adaptation to changing conditions in the internal of a domain, or even across different domains.

3.4. Observations

Several observations can be drawn from the use case examples in this section:

1 Service instances for a specific service may exist in more than one network location, e.g., for replication purposes to serve localized demand.

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., scaling in and out dynamically to cope with temporary load situations.

3 Decisions to utilize a specific service instance may be service-specific, 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 Decision points for selecting the 'right' or 'best' service instance may be dynamic under the given service-specific decision policy. Thus, 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.

There exist a number of L4 through L7 based solutions to realize the aforementioned use cases, with [I-D.liu-can-gap-reqs] providing an initial overview into the gaps that those solutions experience in the light of the observations above.

A key takeaway from this analysis is that the explicit indirection for service discovery, realized for instance through DNS, GSLB or other solutions, poses a challenge to the dynamicity also observed in our use cases here due to the additional latency incurred but also due to the relatively static mapping of service name onto network locator that is maintained in most of those solutions. The work in [OnOff2022] investigates the impact of such off-path vs possible on-path decision making onto service performance and user experience.

In the next section, we outline requirements for a solution that would realize those use cases and address some of the gaps outlined in [I-D.liu-can-gap-reqs], with Section 5 presenting our initial design on how to address those requirements through a shim layer atop IPv6.

4. Requirements

The following requirements for a routing on service addresses (ROSA) solution (referred to as 'solution' for short) have been identified from our use cases in the previous section:

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

(a) Solution MUST provide secure association of service address to service owner.

(b) Solution SHOULD provide means to obfuscate the purpose of communication to intermediary network elements.

(c) Solution MAY provide means to obfuscate the constraint parameters used for selecting specific service instances.

REQ2: Solution 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) Solution MUST provide scalable means for route announcements.

(b) Solution MUST announce routes within a ROSA domain.

(c) Solution SHOULD provide means to delegate route announcement.

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

REQ3: Solution MUST provide means to interconnect ROSA islands.

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

(b) Solution MUST allow for announcing computational processes outside ROSA domains.

REQ4: Solution MUST provide constraint-based routing capability.

(a) Solution MUST provide means to announce routing constraints associated with specific service instances.

(b) Solution SHOULD allow for providing operation for constraint matching in announcement.

(c) Solution MUST at least provide exact constraint match during request routing.

(d) Solution MUST provide first match, if more than one match found.

(e) Solution SHOULD provide random match, if more than one match found.

(f) Solution SHOULD provide match to all, if more than one match found.

(g) Solution MAY provide partially ordered matches.

REQ5: Solution MUST provide scheduled instance selection at ROSA ingress nodes.

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

REQ6: Solution MUST support instance affinity during request routing, i.e., a request is sent from client to one dedicated service instance as part of an ongoing service transaction.

(a) Solution MUST adhere affinity to the service instance chosen in the initial service request of the service transaction.

REQ7: 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 affinity requests.

REQ8: Solution SHOULD support in-request mobility for a ROSA client.

REQ9: Solution SHOULD support transaction mobility, i.e., changing service instances during an ongoing service transaction.

REQ10: Solution SHOULD support TLS 0-RTT handshakes without the need for pre-shared certificates.

5. ROSA Design

This section outlines the design of a shim layer relying upon IPv6 to provide routing on service addresses (ROSA). It first outlines the system overview, before elaborating on various aspects of ROSA in terms of shim layer interactions, forwarding operations, needed client changes, traffic steering methods, interconnection and security considerations.

5.1. System Overview

Figure 1 illustrates a ROSA-enabled limited domain [RFC8799], interconnected to other ROSA-supporting domains via the public Internet through the Service Address Gateway (SAG). Section 5.6 provides more detail on how to achieve that interconnection. ROSA is positioned as a shim overlay atop IPv6, using Extension headers that carry the suitable information for routing and forwarding the ROSA service requests, unlike [I-D.eip-arch] which proposes to include extension processing directly into the transport network. With that in mind, a single ROSA domain may span across more than one network-level domain, thereby allowing for the multi-AS ROSA deployments.

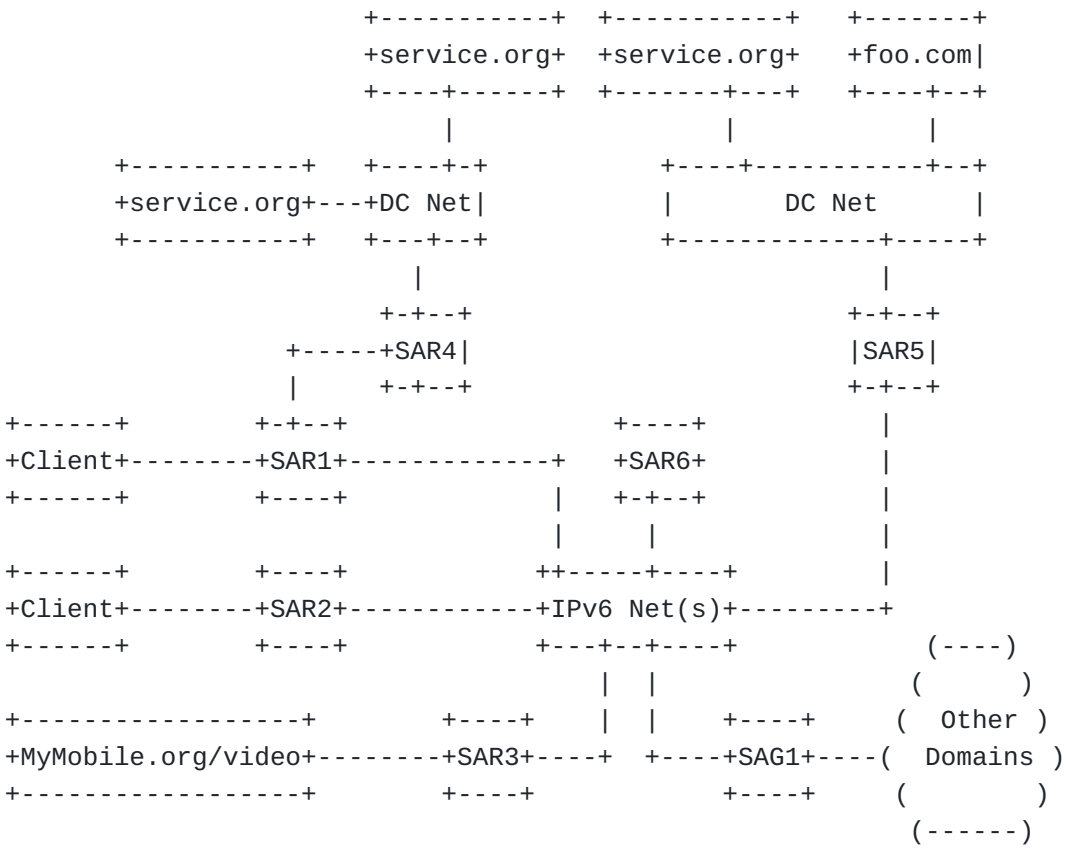

Figure 1: ROSA System Overview

ROSA endpoints start with discovering their ingress Service Address Router (SAR), e.g., through DHCP extensions or through utilizing the Session Management Function (SMF) in 5G networks [_3.501]. An endpoint may discover several ingress SARs for different categories of services, each SAR being part of, e.g., a category-specific ROSA overlay, which in turn may be governed by different routing policies and differ in deployment (size and capacity). The category discovery mechanism may be subject to specific deployments of ROSA and thus is likely outside the scope of this document.

Services are realized by service instances, possibly at different network locations. Those instances expose their availability to serve requests through announcing the service address of their service to their ingress SAR.

To invoke a service, a ROSA client sends an initial request, addressed to a service, to its ingress SAR, which in turn steers the request (possibly via other SARs) to one of possibly many service

instances. See Section 5.3 for the required SAR-local forwarding operations and end-to-end message exchange and Section 5.4 for the needed changes to ROSA clients.

We refer to initial requests as 'service requests'. If an overall service transaction creates ephemeral state, the client may send additional requests to the service instance chosen in the (preceding) service request; we refer to those as 'affinity requests'. With this, routing service requests (over the ROSA network) can be positioned as on-path service discovery, contrasted against explicit, often off-path solutions such as the DNS.

In order to support transactions across different service instances, e.g., within a single DC, a sessionID may be used, as suggested in [SOI2020]. Unlike [SOI2020], discovery does not include mapping abstract service classes onto specific service addresses, avoiding semantic knowledge to exist in the ROSA shim layer for doing so.

With the above, we can outline the following design principles that guide the development for the solutions described next:

- Service addresses have unique meaning only in the overlay network.

- Service instance IP addresses have meaning only in the underlay networks, over which the ROSA domain operates.

- SARs map service addresses to the IP addresses for the next hop to send the service request to, finally directed to the service instance IP address.

- Within the underlay network, service instance IP addresses have both locator and identifier semantics.

- A service address within a ROSA domain carries both identifier and locator semantics to other nodes within that domain but also other ROSA domains (through the interconnection methods shown in Section 5.6).

- Affinity requests directly utilize the underlay networks, based on the relationships build during the service request handling phase.

We can recognize similarities of these principles with those outlined for the Locator Identifier Separation Protocol (LISP) in [I-D.ietf-lisp-introduction] albeit extended with using direct IP communication for longer service transactions.

5.2. Message Types

Apart from affinity requests, which utilize standard IPv6 packet exchange between the client and the service instance selected through the initial service request, ROSA introduces three new message types, shown in Figure 2.

NOTE: more detailed IP header style notation will be added in later versions.

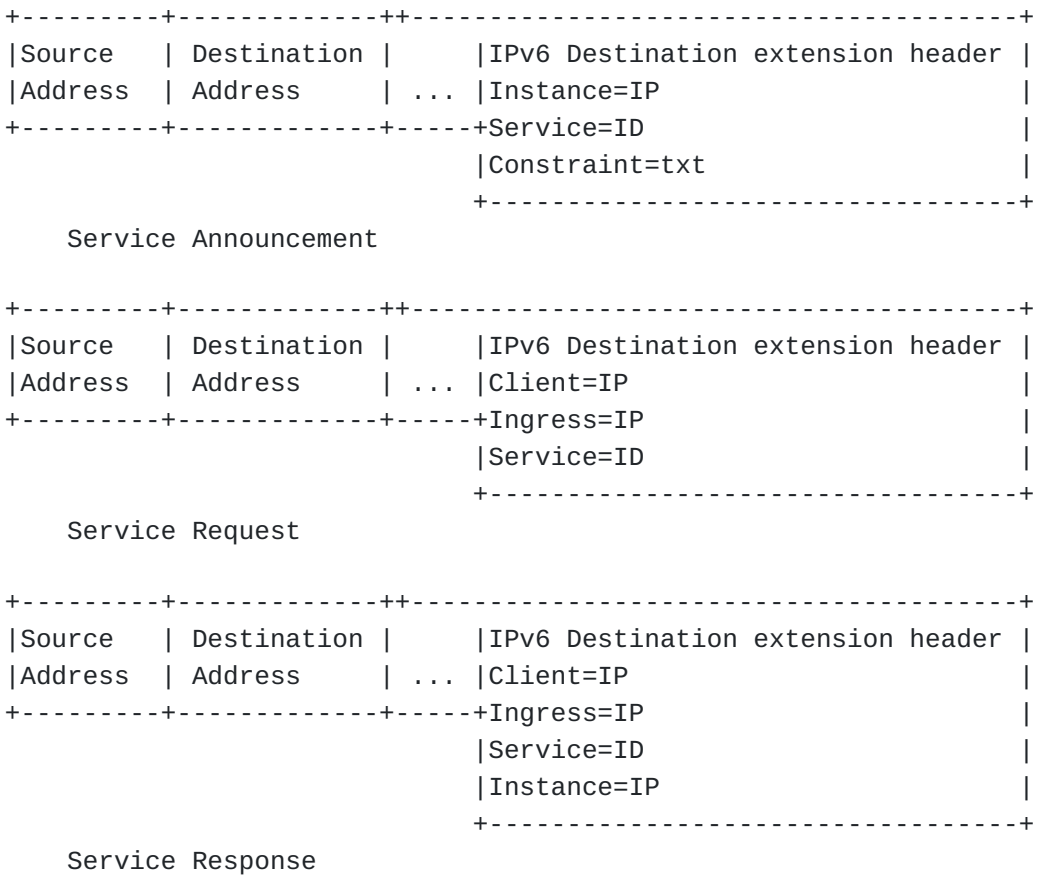

Figure 2: ROSA message types

Given the overlay nature of ROSA, clients, SARs, and service instances are destinations in the IPv6 underlay of the network domains that the overlay spans across. For this reason, we use the destination option EH [RFC8200], where Figure 2 highlights only the entries needed for the specific purpose of the message, omitting other IPv6 packet header information for simplicity. The initial prototype uses a TLV format for the extension header with Concise Binary Object Representation (CBOR) [RFC8949] being studied as an

alternative. The EH entries shown are populated at the client and service instance, while read at traversing SARs.

A service address is encoded through a hierarchical naming scheme, e.g., using [RFC8609]. Here, service addresses consist of components, mapping existing naming hierarchies in the Internet onto those over which to forward packets, illustrated in the forwarding information base (FIB) of Figure 3 as illustrative URLs. With components treated as binary objects, the hierarchical structure allows for prefix-based grouping of addresses, reducing routing table size, while the explicit structure allows for efficient hash-based lookup during forwarding operations, unlike IP addresses which require either log(n) radix tree search software or expensive TCAM hardware solutions.

Note that other encoding approaches could be used, such as hashing the service name at the ROSA endpoint or assigning a service address through a mapping system, such as the DNS, but this would require either additional methods, e.g., for hash conflict management or name-address mapping management, which lead to more complexity.

With the service announcement message, a service instance signals towards its ingress SAR its ability to serve requests for a specific service address. Section 5.5 outlines the use of this message in routing or scheduling-based traffic steering methods.

The service request message is originally sent by a client to its ingress SAR, which in turn uses the service address provided in the extension header to forward the request, while the selected service instance provides its own IP locator as an extension header entry in the service response. The next section describes the SAR-local forwarding operations and the end-to-end message exchange that uses the extension header information for traversing the ROSA network, while Section 5.6 outlines the handling of service addresses that have not been previously announced within the client-local ROSA domain.

5.3. SAR Forwarding Engine

The SAR operations are typical for an EH-based IPv6 forwarding node: an incoming service request or response is delivered to the SAR forwarding engine, parsing the EH for relevant information for the forwarding decision, followed by a lookup on previously announced service addresses, and ending with the forwarding action.

Figure 3 shows a schematic overview of the forwarding engine with the forwarding information base (FIB) and the next hop information base (NHIB) as main data structures. The NHIB is managed through a

routing protocol, see Section 5.5, with entries leading to announced services.

The FIB is dynamically populated by service announcements, with the FIB including only one entry into the NHIB when using routing-based methods (rows 0 to 3 in Figure 3), described in Section 5.5.2. Scheduling-based solutions (see Section 5.5.1), however, may yield several dynamically created entries into the NHIB (items 0, 4 and 5 in Figure 3, where SI1 and SI2 represent the IPv6 address announced by the respective service instances) as well as additional information needed for the scheduling decision; those dynamic NHIB entries directly identify service instances locations (or their egress as in item 0) and only exist at ingress SARs towards ROSA clients.

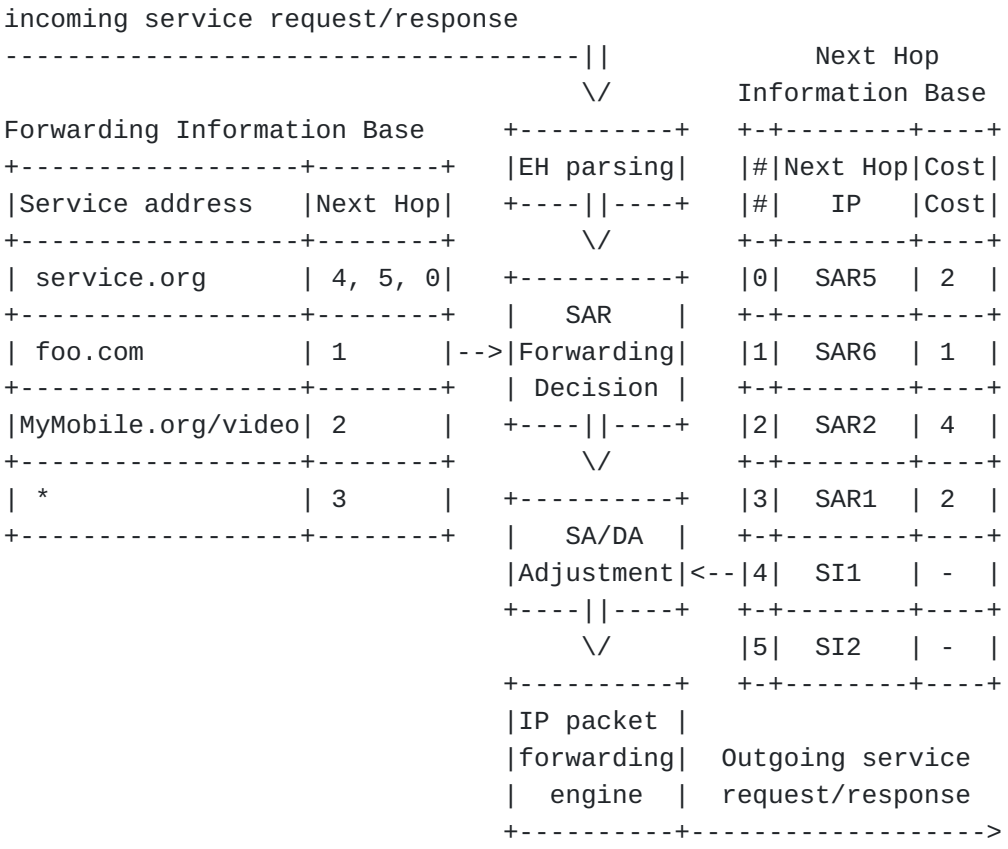

Figure 3: SAR forwarding engine model

For a service request, a hash-based service address lookup (using the Service EH entry) is performed, leading to next hop (NH) information for the IPv6 destination address to forward to (the final destination address at the last hop SAR will be the instance serving the service request).

Forwarding the response utilizes the Client and Ingress EH fields, where the latter is used by the service instance's ingress SAR to forward the response to the client ingress SAR, while the former is used to eventually deliver the response to the client by the client's ingress SAR, ensuring proper firewall traversal of the response back to the client. We have shown in prototype realizations of ROSA that the operations in Figure 3 can be performed using eBPF [eBPF] extensions to Linux SW routers, while [SarNet2021] showed the possibility a realizing a similar design using P4-based platforms.

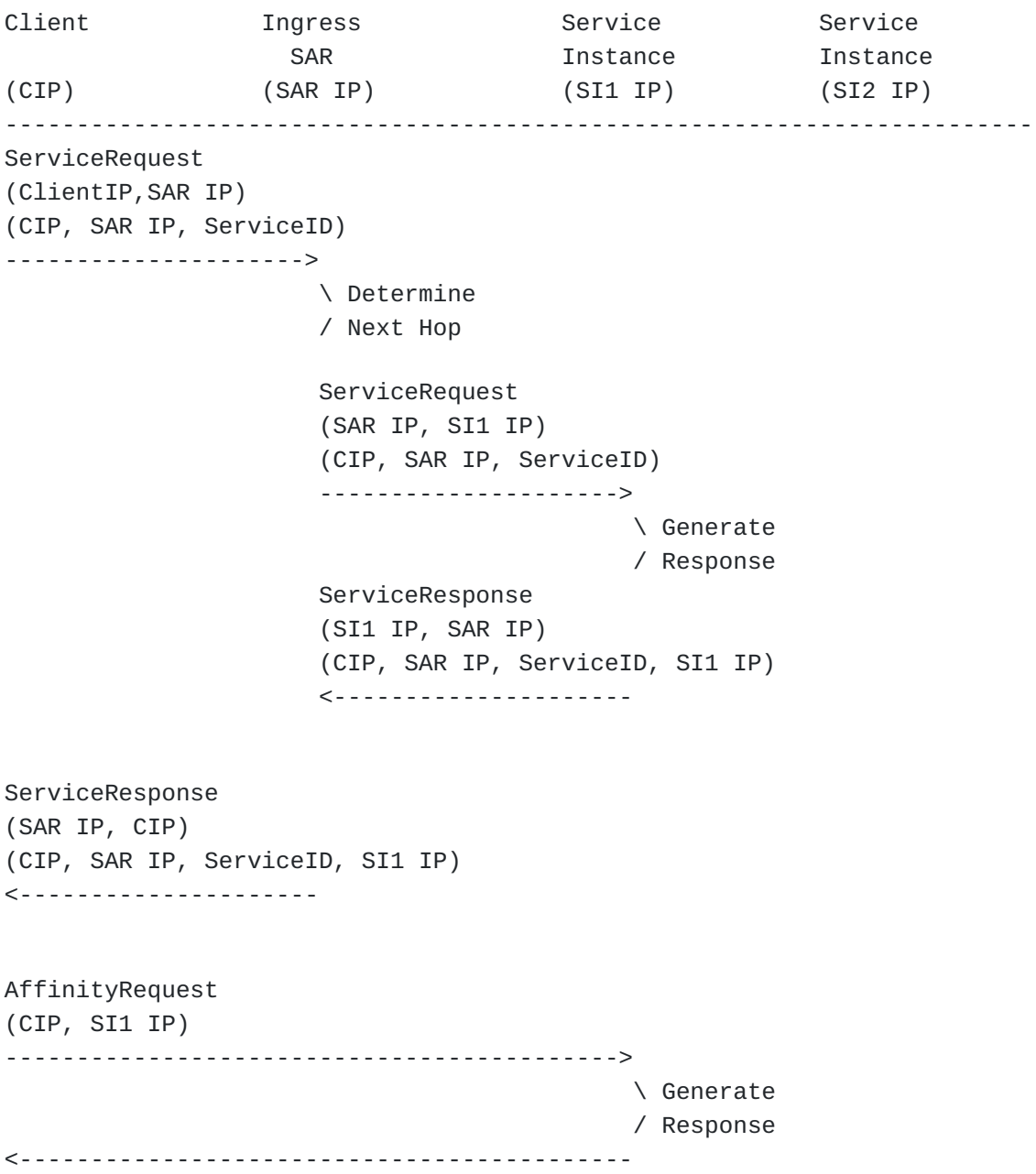

Figure 4: ROSA message exchanges

Figure 4 shows the resulting end-to-end message exchange, using the aforementioned SAR-local forwarding decisions. We here show the IP source and destination addresses in the first brackets and the extension header information in the second bracket.

We can recognize two key aspects. First, the SA/DA re-writing happens at the SARs, using the EH-provided information on service address, initial ingress SAR and client IP locators, as described above. Second, the selection of the service instance is signalled back to the client through the additional Instance EH field, which is used for sending subsequent (affinity) requests via the IPv6 network. As noted in the figure, when using transport layer security, the service request and response will relate to the security handshake, thereby being rather small in size, while the likely larger HTTP transaction is sent in affinity requests. As discussed in Section 9, 0-RTT handshakes may result in transactions being performed in service request/response exchanges only.

5.4. Changes to Clients to Support ROSA

Within endpoints, the ROSA functionality is realized as a shim layer atop IPv6 and below transport protocols. For this, endpoints need the following adjustments to support ROSA:

* Adapting network layer interface: Introducing service addresses requires changes to the current socket interface for discovering the ingress SAR and issuing service requests as well as maintaining affinity to a particular service instance, i.e. mapping a service instance IP address to the initial service address. This could be achieved through providing a new address type (e.g., ADDR_SA) during socket creation, assigning the service address to the returned handle, while utilizing socket options to assign constraints to receiving sockets, utilized in the announcement of the service address. Alternatively, supporting service addresses could be integrated with efforts such as [POSTSOCK2017] to redefine the transport interface towards applications. Any OS-level client changes, as required by introducing new sockets, could be avoided by relying on, e.g., UDP-based, encapsulation of client traffic to the ingress SAR.

* Transport protocol integration: We see our design aligned with existing transport protocols, like TCP or QUIC, albeit with changes required to utilize the aforementioned new address type. For the application (protocol), the opening and closing of a transport connection would then signal the affinity to a specific instance, where the semantic of the 'connection' changes from an IP locator to a service address associated to that specific

service instance. With this, a new service transaction is started, akin to a fresh DNS resolution with IP-level exchange.

* Changes to application protocols: The most notable change for application protocols, like HTTP, would be in bypassing the DNS for resolving service names, using instead the aforementioned different (service) socket type. These adaptions are, however, entirely internal to the protocol implementation. Given the ROSA deployment alongside existing IP protocols, those changes to clients can happen gradually or driven through (e.g., edge SW) platforms.

5.5. Traffic Steering

Traffic steering in ROSA is applied to service requests for selecting the service instance that may serve the request, while affinity requests use existing IPv6 routing and any policies constraining traffic steering in this part of the overall system. At receiving service endpoints, service provisioning platforms may use additional methods to schedule incoming service requests to suitable resources with the ingress point to the service provisioning platform being the service endpoint for ROSA.

In the following, we outline two approaches for traffic steering. The first uses ingress-based scheduling decisions to steer traffic to one of the possible service instances for a given service address. The second follows a routing-based model, determining a single destination for a given service address using a routing protocol.

We envision that some services may be steered through scheduling methods, while others use routing approaches. The indication which one to apply may be derived from the number of next hop entries for a service address. In Figure 3, service.org uses a scheduling method (with instances connected to SAR5 being exposed as a single instance to ROSA, using DC-internal methods for scheduling incoming requests), while the other services are routed via SARs.

Important here is that traffic steering is limited to a single ROSA domain, i.e., traffic steering is not provided across instances of the same service in different ROSA domains; traffic will always be steered to (ROSA) domain-local instances only.

5.5.1. Ingress Request Scheduling

Traffic steering through explicit request scheduling follows an approach similar to application- or transport-level solutions, such as GSLB [GSLB], DNS over HTTPS [RFC8484], HTTP indirection [RFC7231] or QUIC-LB [I-D.ietf-quic-load-balancers]: Traffic is routed to an

indirection point which directs the traffic towards one of several possible destinations.

In ROSA, this indirection point is the client's ingress SAR. However, unlike application or transport methods, scheduling is realized in-band when forwarding service requests in the ingress SAR, i.e., the original request is forwarded directly (not returned with indirection information upon which the client will act), while adhering to the affinity of a transaction by routing subsequent requests in a transaction using the instance's IP address. Scheduling commences to a possibly different instance with the start of a new transaction.

For this, the ingress SAR's NHIB needs to hold information to ALL announced service instances for a service address. Furthermore, any required information, e.g., capabilities or metric information, that is used for the scheduling decision is signalled via the service announcement, with (frequent) updates to existing announcements possible. Announcements for services following a scheduling- rather than a routing-based steering approach carry suitably encoded information in the Constraint field of the announcement's EH, leading to announcements forwarded to client-facing ingress SARs without NHIB entries stored in intermediary SARs.

In addition, a scheduling decision needs to be realized in the SAR forwarding decision step of Figure 3. This may require additional information to be maintained, such as instance-specific state, further increasing the additional NHIB data to be maintained. Examples for scheduling decisions are:

* Random selection of one of the service instances for a given service address, not requiring any additional state information per service address. Announcing the service instance is required once.

* Round robin, i.e., cycling through service instance choices with every incoming service request, requiring to keep an internal counter for the current position in the NHIB for the service address. Announcing the service instance is only required once.

* Capability-based round robin: Cycle through service instances in weighted round robin fashion with the weight (as additional information in each NHIB entry) representing a capability, e.g., number of (normalized) compute resources committed to a service instance. Announcing the service instance requires an update when capabilities change (e.g., during re-orchestration). Weights could be expressed as numerals, limiting the needed semantic exposure of service provider knowledge and thereby supporting the possible separation of service and communication network

provider. The solution in [CArDS2022] realises a compute-aware selection through such decision.

* Metric-based selection: Select service instance with lowest or highest reported metric, such as load, requiring to keep additional metric information per service instance entry in the NHIB. Frequent signalling of the metric is required to keep this information updated.

Although each method yields specific performance benefits, e.g., reduced latency or smooth load distribution, [OnOff2022] outlines simulation-based insights into benefits for realising the compute-aware solution of [CArDS2022] in ROSA.

5.5.2. Routing Across Multiple SARs

In order to send a service request to the `best' service instance (among all announced ones) using a routing-based approach, we build NHIB routing entries by disseminating a service instance's announcement for a given service address S, arriving at its ingress SAR. This distribution may be realized via a routing protocol or a central routing controller, an option suitable for smaller scale deployments.

If no particular constraint is given in the announcement's EH Constraint field, shortest path will be realized as a default policy for selecting the `best' instance, routing any client's request to S the nearest service instance available.

Alternatively, selecting a service instance may use service-specific policies (encoded in the Constraint field of the EH, with the specific encoding details being left for future work). Here, multiple constraints may be used, with [Multi2020] providing a framework to determine optimal paths for such cases, while also conventional traffic engineering methods may be used.

Through utilizing the work in [Multi2020], a number of multi-criteria examples can be modelled through a dominant path model, relying on a partial order only, as long as isotonicity is observed. Typical examples here are widest-shortest path or shortest-widest routing (see [Multi2020]), which allow for performance metrics such as capacity, load, rate of requests, and others. However, metrics such as failure rate or request completion time cannot directly be captured and need formulation as a max metric. Furthermore, metrics may not be isotonic, with Section 3.4 of [Multi2020] supporting those cases through computing a set of dominant attributes according to the largest reduction. [Multi2020] furthermore shows that non-restarting or restarting vectoring protocols may be used to compute

dominant paths and to distribute the routing state throughout the network.

However, the framework in [Multi2020] is limited to unicast vectoring protocols, while the routing problem in ROSA requires selecting the 'best' path to the 'best' instance, i.e., as an anycast routing problem. To capture this, [Multi2020] could be extended through introducing a (anycast) virtual node, placed at the end of a logical path that extends from each service instance to the virtual node. Selecting the best path (over the announced attributes of each service instance) to the virtual node will now select the best service instance (over which to reach the virtual node in the logically extended topology).

Alternatively, ROSA routing may rely on methods for anycast routing, but formulated for service instead of anycast addresses. For instance, AnyOpt [AnyOpt2021] uses a measurement-based approach to predict the best (in terms of latency) anycast (i.e. service) instance for a particular client. Alternatively, approaches using regular expressions may be extended towards spanning a set of destinations rather than a single one. Realizations in a routing controller would likely improve on convergence time compared to a distributed vector protocol; an aspect for further work to explore.

5.6. Interconnection

There are two cases for interconnection: access to (i) non-ROSA services in the public Internet and (ii) ROSA services not domain-locally announced but existing in other domains.

For both cases, we utilize a reserved wildcard service address '*' that points to a default route for any service address that is not being advertised in the local domain. This default route is the service address gateway (see Figure 1), ultimately receiving the service request to the locally unknown service.

Upon arriving at the SAG, it searches its local routing table for any information. If none is found, it consults the DNS to retrieve an IP address where the service is hosted; those mappings could be cached for improving future requests or being pre-populated for popular services.

For case (i), the resolution returns a server's IP address to which the SAG sends the service request with its own IP address as source address. The service response is routed back via the SAG, which in turn uses the Ingress EH information to return the response to the client via its ingress SAR.

For case (ii), the IP address would be that of the SAG of the ROSA domain in which the service is hosted. For this, a domain-local

service instance would have exposed its service, e.g., Mobile.com/video Figure 1, by registering its domain-local SAG IP address with the mapping service. To suitably forward the request, the SAG adds its own IP address as the value to an additional SAG label into the extension header. At the destination SAG, the service address information, extracted from the extension header, is used to forward the service request based on ROSA mechanisms. For the service response, the destination SAG uses the SAG entry in the EH to return the response to the originating ROSA domain's SAG, which in turn uses the Ingress information of the EH to return the response via the ingress to the client.

Given the EH deployment issues pointed out in [SHIM2014], a UDP-based encapsulation may overcome the observed issues, not relying on the EH being properly observed during the traversal over the public Internet. Furthermore, while Figure 1 shows the SAG as an independent component, we foresee deployments in existing PoPs. This would allow combining provisioning through frontloaded PoP-based services and ROSA services. Any service not explicitly announced in the ROSA system would lead to being routed to the PoP-based SAG, which may use any locally deployed services before forwarding the request to the public Internet.

6. Open Issues

7. Relation to IETF/IRTF Efforts

8. Conclusions

TBD

9. Security Considerations

Aligned with security considerations in existing service provisioning systems, we address aspects related to authenticity, i.e., preventing fake service announcements, confidentiality, both in securing relationship as well as payload information, and operational integrity.

* Announcement security: A key exchange between service and network provider may be used to secure the service announcement for ensuring an authorized announcement of services. Self-certifying identifiers could be used for this purpose

* Relationship security: Using service addresses at the routing layer poses not just a privacy but possibly also a net neutrality problem, allowing for non-ROSA elements to discriminate against specific service addresses. Similar to [I-D.per-app-networking-considerations], service addresses could reflect service categories, not services themselves. Service

endpoints to those category-level services can use information in the secured payload (e.g., the URL in an HTTP-based service invocation) to direct the traffic accordingly. The downside of such model is a possible convergence towards a PoP-like model of service provisioning, since exposing an entire service category naturally requires provisioning many possible services under that category, likely favouring large-scale providers over smaller ones; an imbalance that ROSA intends to change, not favour. Work on identity privacy in ILNP [ILNP2021] has shown that ephemeral identifiers may increase the private nature of the communication relation; a direction that needs further exploration in the context of our work. Also, the service address in the extension header could be encrypted, based on a key exchange during the SAR discovery. However, the impact of such mechanism would need further study.

*Transport-level security: Given the often sensitive nature of service requests, payload security is key. We adopt techniques used in TLSV1.3 [RFC8446], providing a 1-RTT handshake for communication between formerly untrusted parties. While the initial 'Client Hello' is sent as a service request, the subsequent communication uses the topological address of the responding server in an affinity request. Using pre-shared keys may allow for communication between trusted client and service instances, e.g., where the client is provided by the service authority and preconfigured with a pre-shared key. This results in a 0-RTT handshake with the 'Client Hello' including the initial service data, encrypted with the pre-shared key. This comes with known forward-secrecy issues and should be avoided in networks with untrusted intermediary nodes. Alternatively, the service's public key could encrypt the initial security handshake, akin to the solutions proposed for Encrypted Client Hello (ECH), using the DNS for obtaining the public key.

*Bandwidth DoS: We assume network provider level mechanisms to restrict traffic injected both by the service provider and client, including for the number of service advertisements in order to control the routing traffic.

*Denying routing service: A SAR could maliciously deny forwarding of client requests, which is no different from denying IP packet forwarding. In both cases, we assume an existing commercial relationship that avoids such situation.

10. IANA Considerations

This draft does not request any IANA action.

11. Acknowledgements

Many thanks go to Mohamed Boucadair for his comments to the text to clarify several aspects of the technical details of ROSA.

12. Informative References

[AnyOpt2021]
Zhang, Z., April, T., Chandrasekaran, B., Choffnes, D., Maggs, B. M., Shen, H., Sitaraman, R. K., Yang, X., Zhang, X., and T. Sen, "AnyOpt: predicting and optimizing IP Anycast performance", Paper ACM SIGCOMM, 2021.

[BBF] ""Control and User Plane Separation for a disaggregated BNG"", Technical Report-459 Broadband Forum (BBF), 2020.

[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.

[CV19]
Feldmann, A., Gasser, O., Lichtblau, F., Pujol, E., Poese, I., Dietzel, C., Wagner, D., Wichtlhuber, M., Tapiador, J., Vallina-Rodriguez, N., Hohlfeld, O., and G. Smaragdakis, "A Year in Lockdown: How the Waves of COVID-19 Impact Internet Traffic", Paper Communications of ACM 64, 7 (2021), 101-108, 2021.

[eBPF] "What is eBPF?", Technical Report eBPF Foundation, 2022, <https://ebpf.io/what-is-ebpf/>.

[EI2021]
Cidon, I., Culler, D., Estrin, D., Katz-Bassett, E., Krishnamurthy, A., McCauley, M., McKeown, N., Panda, A., Ratnasamy, S., Rexford, J., Schapira, M., Shenker, S., Stoica, I., Tennenhouse, D., Vahdat, A., Zegura, E., Balakrishnan, H., and S. Banerjee, "Revitalizing the public internet by making it extensible", Paper ACM

Computer Communication Review, Vol. 51. 18-24. Issue 2, 2021.

[Gini] "Gini Coefficient", Technical Report Wikipedia, 2022, <https://en.wikipedia.org/wiki/Gini_coefficient>.

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

[HHI] "Herfindahl-Hirschman index", Technical Report Wikipedia, 2022, <https://en.wikipedia.org/wiki/Herfindahl-Hirschman_index>.

[Huston2021] Huston, G., "Internet Centrality and its Impact on Routing", Technical Report IETF side meeting on 'service routing and addressing', 2021, <https://github.com/danielkinguk/sarah/blob/main/conferences/ietf-112/materials/Huston-2021-11-10-centrality.pdf>.

[I-D.eip-arch] Salsano, S., ElBakoury, H., and R. Diego Lopez, "Extensible In-band Processing (EIP) Architecture and Framework", Work in Progress, Internet-Draft, draft-eip-arch-00, 15 June 2022, <https://www.ietf.org/archive/id/draft-eip-arch-00.txt>.

[I-D.ietf-lisp-introduction] Cabellos, A. and D. S. (Ed.), "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-introduction-15, 20 September 2021, <https://www.ietf.org/archive/id/draft-ietf-lisp-introduction-15.txt>.

[I-D.ietf-quic-load-balancers] Duke, M., Banks, N., and C. Huitema, "QUIC-LB: Generating Routable QUIC Connection IDs", Work in Progress, Internet-Draft, draft-ietf-quic-load-balancers-14, 11 July 2022, <https://www.ietf.org/archive/id/draft-ietf-quic-load-balancers-14.txt>.

[I-D.liu-can-gap-reqs] Liu, P., Jiang, T., Eardley, P., Trossen, D., Li, C., and D. Huang, "Computing-Aware Networking (CAN) Gap Analysis and Requirements", Work in Progress, Internet-Draft, draft-liu-can-gap-reqs-00, 23 October 2022, <https://www.ietf.org/archive/id/draft-liu-can-gap-reqs-00.txt>.

[I-D.nottingham-avoiding-internet-centralization] Nottingham, M., "Centralization, Decentralization, and Internet Standards", Work in Progress, Internet-Draft, draft-nottingham-avoiding-internet-centralization-05, 9

July 2022, <https://www.ietf.org/archive/id/draft-nottingham-avoiding-internet-centralization-05.txt>.

[I-D.per-app-networking-considerations] Colitti, L. and T. Pauly, "Per-Application Networking Considerations", Work in Progress, Internet-Draft, draft-per-app-networking-considerations-00, 15 November 2020, <https://www.ietf.org/archive/id/draft-per-app-networking-considerations-00.txt>.

[I-D.wadhwa-rtgwg-bng-cups] Wadhwa, S., Shinde, R., Newton, J., Hoffman, R., Muley, P., and S. Pani, "Architecture for Control and User Plane Separation on BNG", Work in Progress, Internet-Draft, draft-wadhwa-rtgwg-bng-cups-03, 11 March 2019, <https://www.ietf.org/archive/id/draft-wadhwa-rtgwg-bng-cups-03.txt>.

[ILNP2021] Yanagida, R., Bhatti, S., and G. Haywood, "End-to-end privacy for identity and location with IP", Paper 2nd Workshop on New Internetworking Protocols, Architecture and Algorithms, 29th IEEE International Conference on Network Protocols, 2021.

[ISOC2022] "Internet Centralization", Technical Report ISOC Dashboard, 2022, <https://pulse.internetsociety.org/centralization>.

[LDCU2021] Carpenter, B., Crowcroft, C., and D. Trossen, "Limited domains considered useful", Paper ACM Computer Communication Review, Vol. 51. 22-28. Issue 3, 2021.

[Multi2020] Ferreira, M. A. and J. L. Sobrinho, "Routing on Multi Optimality Criteria", Paper ACM SIGCOMM, 2020.

[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.

[POSTSOCK2017] Kuehlewind, M., Trammell, B., and C. Perkins, "Post sockets: Towards an evolvable network transport

interface", Paper IFIP Networking Conference (IFIP Networking) and Workshops, 2017.

[RFC6770] Bertrand, G., Ed., Stephan, E., Burbridge, T., Eardley, P., Ma, K., and G. Watson, "Use Cases for Content Delivery Network Interconnection", RFC 6770, DOI 10.17487/RFC6770, November 2012, <https://www.rfc-editor.org/info/rfc6770>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

[RFC8609] Mosko, M., Solis, I., and C. Wood, "Content-Centric Networking (CCNx) Messages in TLV Format", RFC 8609, DOI 10.17487/RFC8609, July 2019, <https://www.rfc-editor.org/info/rfc8609>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/

RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[SarNet2021] Glebke, R., Trossen, D., Kunze, I., Lou, Z., Rueth, J., Stoffers, M., and K. Wehrle, "Service-based Forwarding via Programmable Dataplanes", Paper 1st Intl Workshop on Semantic Addressing and Routing for Future Networks, 2021.

[SHIM2014] Naderi, H. and B. Carpenter, "Putting SHIM6 into practice", Paper 2014 Australasian Telecommunication Networks and Applications Conference (ATNAC), 2014.

[SOI2020] Jiang, S., Li, G., and B. Carpenter, "A New Approach to a Service Oriented Internet Protocol", Paper IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020.

[SVA] ""Optimizing Video Delivery With The Open Caching Network"", Technical Report Streaming Video Alliance, 2018.

[TIES2021] Giotsas, V., Kerola, S., Majkowski, M., Odinstov, P., Sitnicki, J., Chung, T., Levin, D., Mislove, A., Wood, C. A., Sullivan, N., Fayed, M., and L. Bauer, "The Ties that un-Bind: Decoupling IP from web services and sockets for robust addressing agility at CDN-scale", Paper ACM SIGCOMM, 2021.

[_3.501] "System architecture for the 5G System (5GS); Stage 2 (Release 16)", Technical Report 3GPP TS 23.501 V16.11.0 (2021-12), 2021, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>.

Authors' Addresses

Dirk Trossen
Huawei Technologies
Munich
Germany

Email: dirk.trossen@huawei.com

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/