IETF – RTG WG

https://www.ietf.org/id/draft-trossen-rtgwg-rosa-00.html

Routing on Service Addresses (ROSA)

IETF 115

Dirk Trossen (Huawei), Luis Contreras (Telefonica)

10.10.2022

Goal for this Draft

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).

Structure of this Draft

To be added in minor upgrade

Terminology

- **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 or 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 node 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.

Use Cases

- CDN Interconnect and distribution (Section 3.1)
  - Key aspects
    - multi-site **replication**
    - **Dynamic** decision making
    - **Reducing latency var**iance through multi-site retrieval
- Distributed user planes for mobile and fixed access (Section 3.2)
  - Key aspects
    - **Distributed** user plane functions (UPFs)
    - UPFs may be **service- and thus policy-specific**
    - Supporting edge **compute** capabilities
- Multi-homed and multi-domain services (Section 3.3)
  - Key aspects
    - Services often deployed **across administrative do**mains (e.g., enterprise scenarios)
    - **Multi-homing** often used at both client and service side
    - Consideration of **instantaneously 'best'** conditions to select service instance

How to efficiently steer traffic across possibly significantly distributed networks locations, enabling dynamicity in the selection of the 'best' service instance?

Main Idea

Replace DNS+IP sequence, i.e., the off-path discovery of service name to IP locator mapping with *on-path discovery* of suitable service instance location

<u>For this:</u>

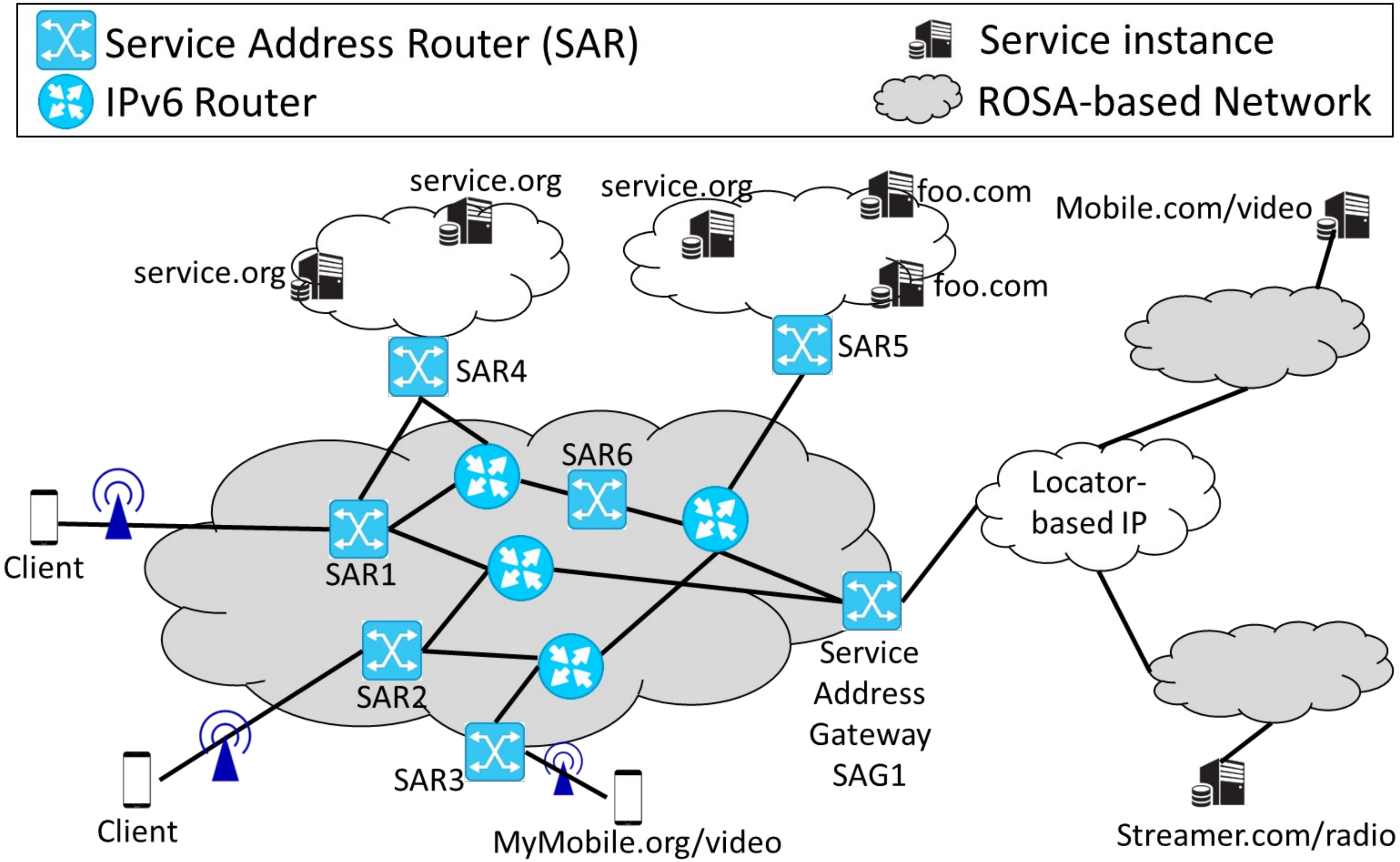
1. Send initial IP packet, “directed” to service address *S* to a special shim overlay that routes the packet based on the service name, not service instance locator
   - Use mappings, replacing the role of DNS records, between service name and possible service instance location(s)
2. Deliver chosen service instance location *SI* in response to initial packet back to client
3. Now use *SI* in native IPv6 packets to direct send subsequent packets to the chosen service instance
   - This is to support possible ephemeral state created at service instance as consequence of initial IP packet

Repeat steps 1 and 3 for every service transaction, allowing those transactions now to be served at any of the available service instance albeit keeping one transaction at one chosen service instance!

- For stateless services, only steps 1 and 2 are executed!

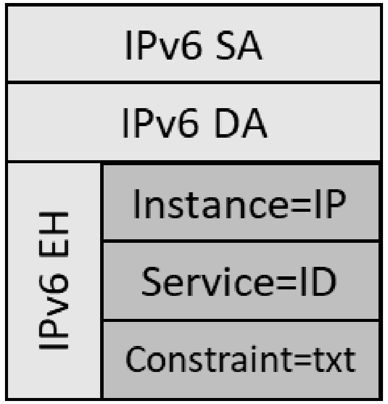
Key point: in-band/on-path discovery is performed at IP packet level, NOT application level!

System Overview



- Located at **L3.5** with ROSA-specific IPv6 destination extension headers
  - Deployed in network or in edge site
- **On-path** routing for initial service requests rather than off-path indirection in DNS+IP, GSLB, QUIC_LB
- **Flexible traffic steering** with service-specific policies
  - Can rely purely on ingress-based selection or use intra-SAR routing
- **Instance affinity** over native IPv6
- **Routing table size limited** to services supported by ROSA domain
- **No client-awareness needed**, i.e., Clients may access ROSA-internal, inter-ROSA and Internet services through SAG

Message Types and Layering



IPv6 SA

IPv6 DA

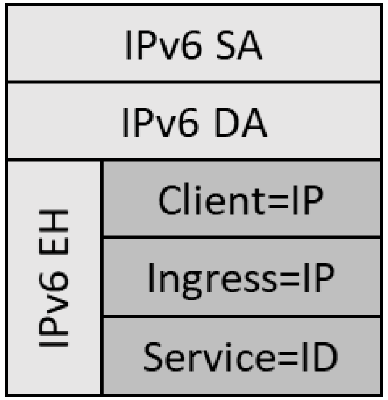
IPv6 EH

Instance=IP

Service=ID

Constraint=txt

Service Announcement



IPv6 SA

IPv6 DA

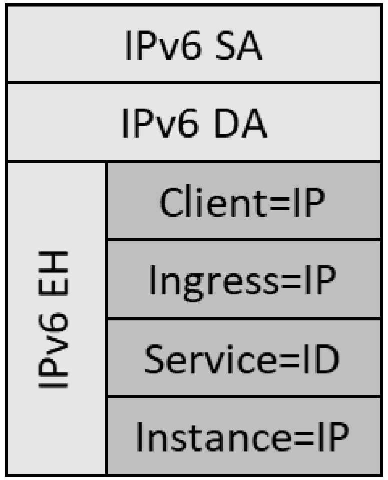
IPv6 EH

Client=IP

Ingress=IP

Service=ID

Service Request



IPv6 SA

IPv6 DA

IPv6 EH

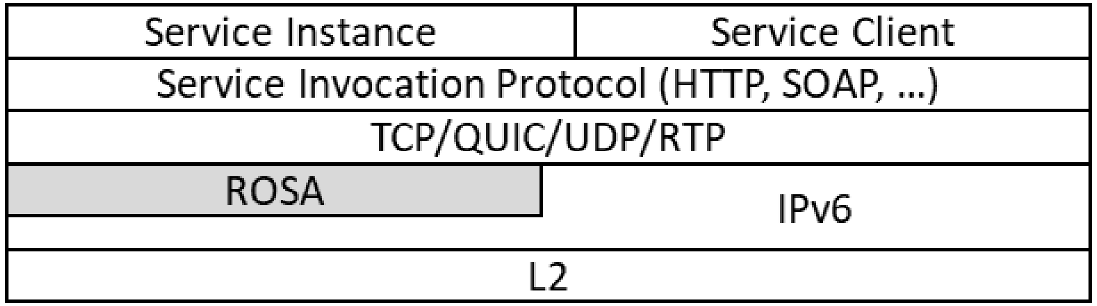
Client=IP

Ingress=IP

Service=ID

Instance=IP

Service Response



Service Instance | Service Client

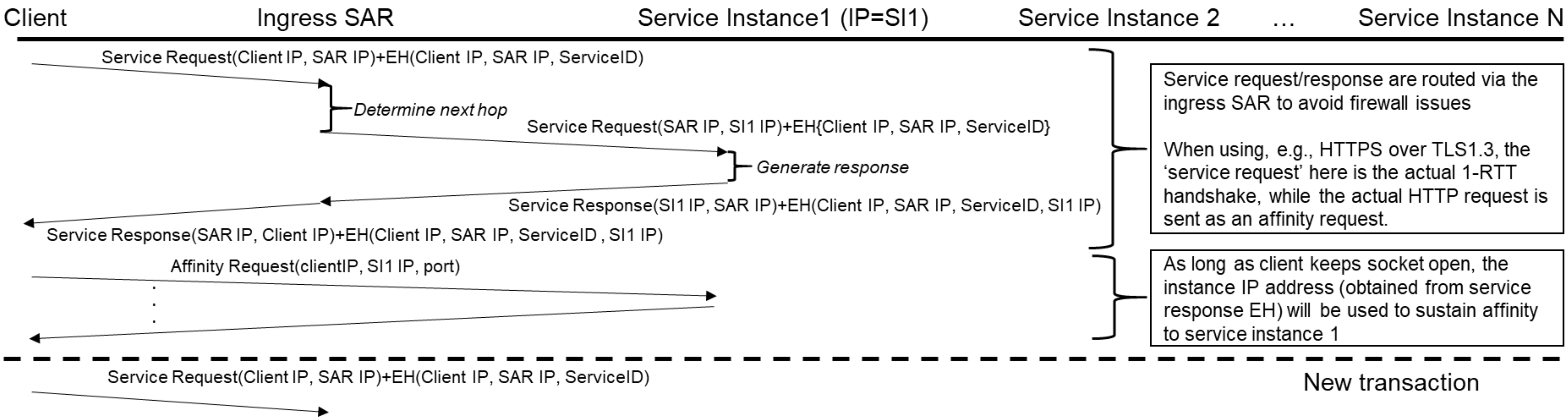
Service Invocation Protocol (HTTP, SOAP, …)

TCP/QUIC/UDP/RTP

ROSA | IPv6

L2

Message Flow



Client

Ingress SAR

Service Instance1 (IP=SI1)

Service Instance 2

…

Service Instance N

Service Request(Client IP, SAR IP)+EH(Client IP, SAR IP, ServiceID)

Determine next hop

Service Request(SAR IP, SI1 IP)+EH{Client IP, SAR IP, ServiceID}

Generate response

Service Response(SI1 IP, SAR IP)+EH(Client IP, SAR IP, ServiceID, SI1 IP)

Service Response(SAR IP, Client IP)+EH(Client IP, SAR IP, ServiceID , SI1 IP)

Affinity Request(clientIP, SI1 IP, port)

Service request/response are routed via the ingress SAR to avoid firewall issues

When using, e.g., HTTPS over TLS1.3, the 'service request' here is the actual 1-RTT handshake, while the actual HTTP request is sent as an affinity request.

As long as client keeps socket open, the instance IP address (obtained from service response EH) will be used to sustain affinity to service instance 1

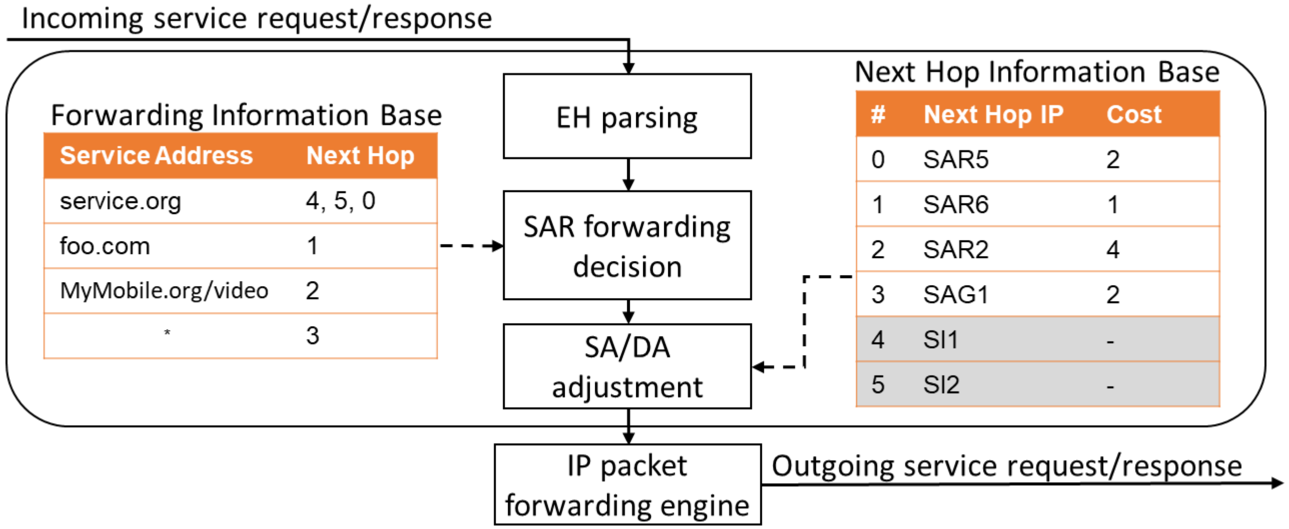
Service Request(Client IP, SAR IP)+EH(Client IP, SAR IP, ServiceID)

New transaction

- **Key point**: only service requests are sent via ROSA shim layer, while affinity requests follow the direct client-service instance IPv6 path
- New socket interface maintains the mapping of service address to instance IP address for subsequent requests to ensure affinity of service transaction

-> **Code looks like any socket-based app, changing the address type to AF_ROSA (instead of AF_INET)!**

SAR Forwarding Engine



Incoming service request/response
EH parsing
SAR forwarding decision
SA/DA adjustment
IP packet forwarding engine
Outgoing service request/response

Forwarding Information Base

Service Address	Next Hop
service.org	4, 5, 0
foo.com	1
MyMobile.org/video	2
*	3

Next Hop Information Base

#	Next Hop IP	Cost
0	SAR5	2
1	SAR6	1
2	SAR2	4
3	SAG1	2
4	SI1	-
5	SI2	-

- FIBs are populated by routing protocol
- Traffic steering based on
  - **Request scheduling**, i.e., sending service request to one of possibly many instance at runtime
  - Multi-optimality routing, i.e., sending service request to the ‘best’ service instance according to constraint-based routing metric
- ROSA supports **any** traffic steering mechanism
  - Not necessarily compute-aware!
- **Special wildcard** service address points at SAG to interconnect with other ROSA domain & existing Internet services
  - Could be deployed at CDN to serve PoP-based request first, before directing to Internet

Plans Moving Forward

- More **details** on design and realization (Section 5) & incorporating **feedback** received
  - Proper header descriptions
  - Support for multi-homed service instances (thanks Jens for suggestion!)
  - Support for different naming schemes (harking back to ICN provenance)

- **Use case** insights
  - Find more applications for ROSA with clear benefits
  - Outline those benefits more clearly (if possible with quantifiable insights)

- **Implementation** insights
  - Plans on open sourcing eBPF-based SW SAR
  - Performance results (forwarding and use case driven)
  - Planning demo or hackathon for IETF116

IETF – RTG WG

https://www.ietf.org/id/draft-trossen-rtgwg-rosa-00.html

We seek feedback on

- Problem space and motivation
- Architectural approach and its realization
- **Interested parties and possible contributors**
- Way forward

THANKS!

QUESTIONS? / COMMENTS?

10.10.2022