IETF – RTG WG

https://datatracker.ietf.org/doc/html/draft-trossen-rtgwg-rosa

Routing on Service Addresses (ROSA)

IETF 116

Dirk Trossen (Huawei), Luis Contreras (Telefonica), J. Finkhaeuser (Interpeer), P. Mendes (Airbus)

31.03.2023

Goal for this Draft

Approach is meant to provide an **alternative** (not replacement) to DNS-based service routing through an **in-band/on-path service discovery** approach with transaction **affinity** to support multi-instance, **highly dynamic** (in selection decision) and **highly distributed** service scenarios

by outlining

(i) Argumentation and motivation, (ii) use cases, (iii) benefits, (iv) requirements, (v) possible solution and (vi) possible wider body of work together with own experiences on **feasibility insights**

Structure of this Draft and Changes since IETF115

- Added Jens Finkhaeuser and Paulo Mendes as **co-authors**.
- **Restructured introduction** to improve readability and argumentation for this draft
- **Addressed IETF115 comments** in various parts of the draft, e.g., introduction, analysis (relation to other technologies), traffic steering (relation to anycast) etc
- Addressed **comments received from community** (thanks to Joel, Russ, Tommy, Daniel, Naresh, Dennis, Darren, Peng, Med)
- Added **six new use cases** (mobile applications - Section 3.4, chunk retrieval - Section 3.5, AR/VR - Section 3.6, Cloud-to-Thing - Section 3.7, Metaverse - Section 3.8, and popularity-based services - Section 3.9)
- Added separate **analysis section**, as derived from use cases (Section 4)
- Revised and **linked requirements to use cases** through additional text (Section 5)
- Discussed **possible benefits** from applying ROSA in identified use cases (Section 6)
- Revised ROSA messages figure (Figure 2)
- Added section on possible **extended capabilities to 'base' ROSA** (Section 8), including multi-homing support, namespace support.
- Added and maintaining **open issues** (Section 10)
- Added missing sections, like conclusions (Section 11) and privacy considerations (Section 13)

Use Cases (Section 3) and Analysis (Section 4)

- CDN Interconnect and distribution
- Distributed user planes for mobile and fixed access
- Multi-homed and multi-domain services
- *Micro-service Based Mobile Applications*
- *Constrained Video Delivery*
- *AR/VR through Replicated Storage*
- *Cloud-to-Thing Serverless Computing*
- *Metaverse*
- *Popularity-based Services*

Requirements (Section 5) now linked into use cases

- Separate application and routing namespace with **additional discovery** for mapping adds complexity and latency
- Instances in more than one location, i.e., **anycast** problem
- Possible **ephemeral** instance lifetime
- **Service-specific** anycast decision
- **Highly dynamic** instance selection, down to milliseconds or per request
- Completion latency often an issue for overall **latency budget**
- Often **dependency across** service invocations

Key benefits (Section 6):

- Support for **high dynamicity** of relations
- **Service-specificity** of decisions
- Reduce **DNS dependency**
- Support **higher service distribution**
- **Protocol support** for app namespace -> comm mapping

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

For this:

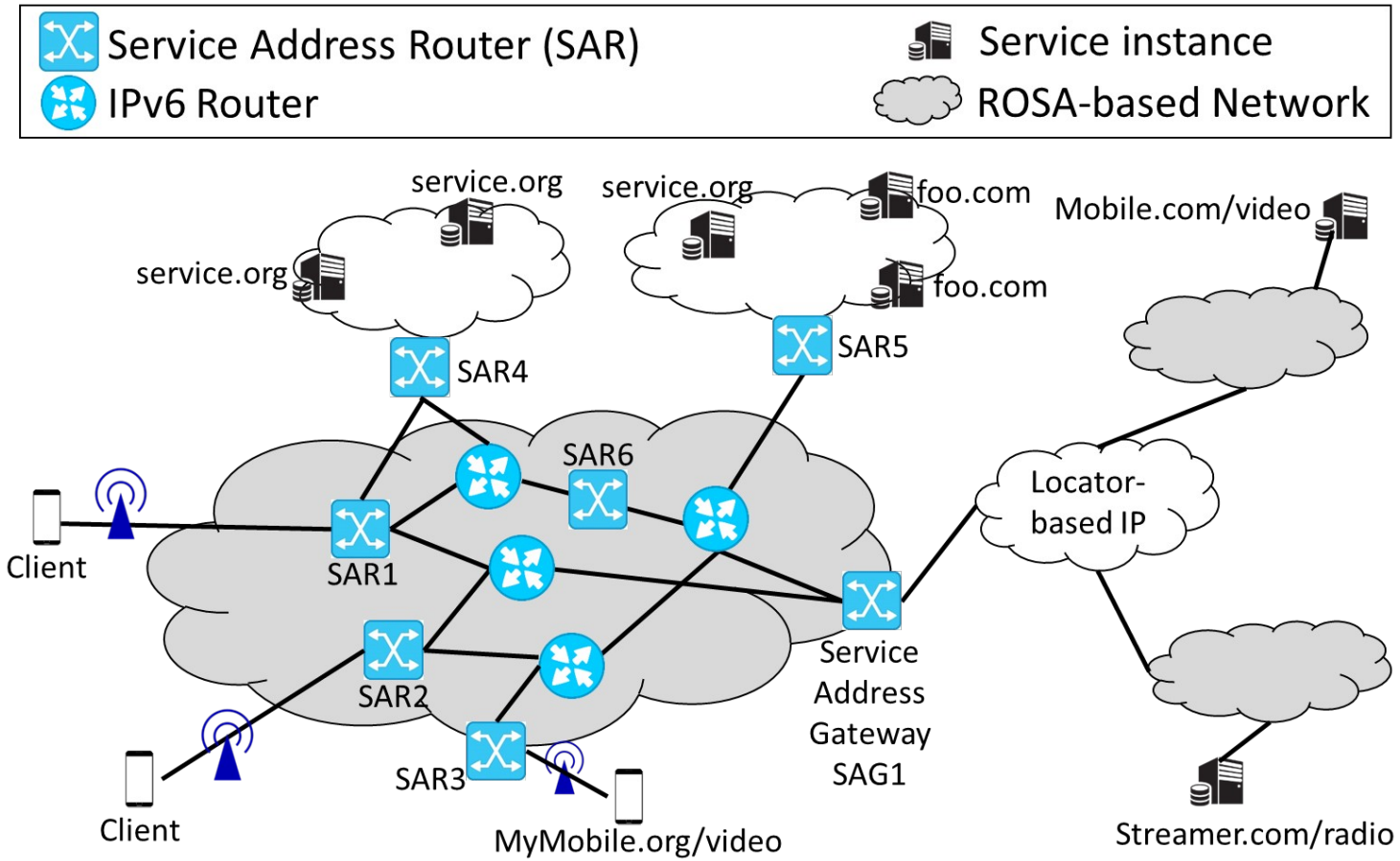
1. Send initial IP packet, “directed” to service address *S* to a special shim overlay
2. The shim overlay routes the packet based on the service address to one of the possibly many service instances for S over one or more existing IP networks.
   - Use mappings, replacing the role of DNS records, between service name and possible service instance location(s)
3. Deliver chosen service instance location *SI* in response to initial packet back to client
4. 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 through 4 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, 2, and 3 are executed!

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

RECAP: System Overview



- Located at **L3.5** with ROSA-specific IPv6 destination extension headers
- **On-path** routing for initial service requests rather than off-path indirection in DNS+IP, GSLB, QUIC_LB
- **In-band discovery** since data already carried with initial service request/response
- **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

Interworking with, not replacing DNS!

Possible Extensions Beyond ‘Base’ ROSA Capability (Section 8)

- Different namespace encodings
  - RFC8066, DNS-based, or hash-based -> different impact on governance, conflicts, etc.
- Multi-homing of service instances
  - Provided by Jens with possible instanceID in addition to instanceIP as a possible solution
- 0-RTT TLS
  - Enabling 0-RTT TLS for short affinity scenarios
- Transaction Mobility
  - Change service instance mid-transaction -> support from ROSA overlay, aiding app-level context transfer
- Service Function Chaining
  - Chaining at app/service name level instead of L2/L3 -> realize name-based SFF in RFC8677 using ROSA
- Privacy-Compliant Communication
  - Avoid exposure of service address in EH -> routing privacy in addition to service categories

Idea is to extend on these possible extensions upon progress of ‘base work’
(ROSA as body of work, not just one specific solution)

Plans Moving Forward

- Tighten **argumentation** for work
  - Discussion with wider community and individuals
  - Position against other, related work

- More **details** on design and realization (Section 7)
  - More header descriptions
  - More details on steering mechanisms, i.e., how to ‘plug-in’ and support any secondary forwarding decision

- **Implementation** insights (fill in Section 9)
  - Plans on open sourcing eBPF-based SW SAR
  - Performance results (forwarding and use case driven)
  - Planning demo or hackathon for IETF117 (sorry, 116 did not work out)

IETF – RTG WG

https://datatracker.ietf.org/doc/html/draft-trossen-rtgwg-rosa

We seek feedback on

- Problem space, **argumentation**, and motivation
- **Positioning of work** within IETF and RTG area
- Architectural **approach** and its realization
- **Interested parties and possible contributors**
- Way forward

THANKS!

QUESTIONS? / COMMENTS?

31.03.2023