

# draft-eckert-anima-grasp-dnssd-00

IETF'100 Singapore, November 2017  
DNSSD WG, version 2

Toerless Eckert, Huawei (Futurewei Technologies USA)  
tte+ietf@cs.fau.de

# ANIMA WG solution

- ANI/ACP: Autonomic - Network Infrastructure / Autonomic Control Plane
  - Automatically built, hop-by-hop encrypted VRF (VRF-lite) – to manage/control networks:
    - Autonomic: between autonomic software in network
      - Management/service agents, distributed protocol instances, ...
    - Classical/SDN: network-devices <-> NOC (NMS/controller/orchestrator/provisioning)
- ACP/ANI: Only IPv6 unicast routing/forwarding
  - No definition/requirement for DNS names – AND NO IP multicast
- GRASP: protocol for discovery, synchronization, negotiation
  - Discovery: Announcement and Request messages: (M\_FLOOD, M\_DISCOVERY)
    - Network wide hop-by-hop reliable flooded across ANI/ACP
  - Header: objective-name (string), hop-count, sequence number (loop-prevention)
    - objective-name” is the “address” for logical endpoints sending/receiving flooded GRASP messages..
  - Payload: undefined – up to each application (per “objective”)
    - Encoding mandatory though: CBOR (message header and payload)

# Use cases

- Autonomic, required by ANI/ACP itself:
  - Bootstrap servers (“BRSKI”).
    - ANI network devices provide bootstrap helper functionality. Connect to bootstrap servers.
  - Servers for renewal of keying material used to protect ANI/ACP (“EST” – RFC7030).
  - ... more in future
- Traditional OAM model:
  - Variety of services in NOC to be discovered by (all or many) network devices to auto-configure those services – and provide automatic server failover:
    - Syslog
    - Time (NTP)
    - Netconf (call-home)
    - DHCP server, DNS server
    - Radius, Diameter, Tacacs (authentication servers)
    - WiFi controller (for AP devices)
    - IPfix, any other data collection export
    - tftp (“yuck”)
    - List goes on...

# OK, sounds great... why this draft again ?

- Do not re-invent for ANI/ACP GRASP aspects of service discovery we already know (and love) from DNS-SD
  - Page 1: Payload: undefined – up to each application (per “objective”)
    - *Not every flooded GRASP objective is a service (announce/request), but if it is, then reuse common definitions!*
  - But keep as simple as possible
    - Do not inherit historic crud (e.g.: `_udp`, `_tcp`)
    - Difference in functionality to unicast DNS – GRASP flooding more like mDNS and `.local`
- So, what do we have with DNS-SD ?
  - Service name space – RFC6335 (e.g.: “brski”, “est”, ...)
  - Service selection criteria (prio, weight)
  - Service instance names (for browsing... how useful without human selection)
  - “Host-names” of service instances – not needed ?!
  - ... (draft-00 has hopefully mapped the 90% important parameters)

# DNS-SD compatible GRASP objectives (CDDL):

```
objective-name    // = SRV.<rfc6335-service-name>
objective-name    // = NAME.<hostname>          ..... If we ever need them
service-element  = {
    ?( &(private:0)      => any), ..... Non-standardized extensions
    ?( &(msg-type:1)     => msg-type
    ?( &(service:2)      => tstr), ..... Service Name ("printer")
    *( &(instance:3)    => tstr), ..... Instance Name („my-kitchen-printer“)
    ?( &(domain:4)      => tstr), ..... Empty = ANI/ACP (like .local), else VRF name
    ?( &(priority:5)    => 0..65535 ), ..... As in DNS-SD
    ?( &(weight:6)      => 0..65535 ), ..... As in DNS-SD
    *( &(kvpairs:7)     => { *(tstr: any) }, .. Key Value pairs – as in DNS-SD
    ?( &(range:8)       => 0..255 ), ..... For distance based service selection
    *( &(clocator:9)    => clocator), ..... GRASP locators with context indicator ("VRF")
}

clocator = [ context, locator-option ] ..... Permit locators to be in data plane
context = tstr ..... Empty: ACP, „0“ = „VRF0“, else name of VRF
locator-option = <unchanged> ..... from GRASP specification – IPv4/IPv6addr/port

msg-type = &( describe: 0, describe-request:1, enumerate:2, enumerate-request:3 ).
```

# Target system

- Simpler, more consistent specification of ANI/ACP services:
  - “service has <name> in IANA registry, parameters according to DNSSD/GRASP” - DONE
- Common service announce/discover API/SDK for servers/clients:
  - Could map to DNS-SD and/or GRASP based on context
- mDNS <-> GRASP services gateway function in NOC router
  - NOC servers could use mDNS (more and more supported already, free SDKs, proven)
  - gateway converts to GRASP,
  - ANI/ACP clients use GRASP
- Outside ANIMA scope:
  - Gateway function for any service announce/discovery of non-OAM services
    - Produced consumed directly by non-network infra (clients)
  - Not limitation of technology but of ANIMA goals:
    - GRASP flooding (with ANI or modified underlying transport) could be alternative to draft-ietf-dnssd-hybrid
    - Technically: Flooding in GRASP makes solution more applicable to services with dense client distribution (everybody needs to know about these services).

# References / details

- Presentation of DNS-SD draft in ANIMA on Monday
- draft-ietf-anima-grasp
- draft-ietf-anima-autonomic-control-plane (ACP)
- draft-ietf-anima-bootstrap-keyinfra (BRSKI)
- draft-ietf-anima-stable-connectivity (NOC services framework)