Autoconfiguration of NOC services in ACP networks via GRASP and more

draft-eckert-anima-noc-autoconfig-00

draft-eckert-anima-grasp-dnssd-01

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

v1.0

Summary / Status

- No update to drafts
  - Seeking feedback from DNS experts in IETF
  - Unfortunately ANIMA conflicts with DNSOP @ IETF103

- Re-summarizing idea / scope:
  - Goal: key pieces to create minimum version of RFC8368

- *draft-eckert-anima-noc-autoconfig-00*
  - Use-case draft: candidate normative document to build standardized, autoconfiguring model for RFC8368
    - ANI/ACP network that automatically uses NOC services
    - E.g.: for full automatic workflow ANI + SDN-controller configuring data-plane

- *draft-eckert-anima-grasp-dnssd-01*
  - *Underlying technology to enable NOC autoconfig (and more)*
  - *Use GRASP as transport for DNS-SD (and some subset of other DNS)*
  - *More use-cases: any ASA announcing/discovering services*
    - *Goal: Do not reinvent DNS-SD service for ANIMA, but map it over GRASP*

Problem and resolution

- ANI Pledge gets enrolled with BRSKI zero-touch

- Now what ?

- We wanted to create more zero-touch "autonomic" nodes

- This draft proposes the smallest possible incremental step
    - For ANI nodes used in conjunction with existing, centrally managed network
    - Minimum standards work to make RFC8368 work better
        - RFC8366 = ANIMA stable connectivity draft: Use ACP for management from NOC
        - RFC8366 is informational, deals with all possible migration scenarios, etc. pp.
        - Does not specify any of the really good, simple automations possible
        - Those automations need standards work

- This draft is extending ANI standard framework to support RFC8366 neworks

So, what do we need ?

- A1: TIME
  - Node needs accurate time. Find NTP server, sync time from it.

- A2: Logging
  - Node needs to automatically syslog, so NOC knows what is going on
  - Including "I am new Pledge, enrolled but unconfigured"

- A3a: secure access of SDN-controller/operator to nodes
  - SSH, Netconf (Netconf typically on top of SSH – or TLS)
  - Automatically discover Radius/Diameter server, enable SSH/Netconf servers, allow access to node SH/Netconf VIA ACP ONLY
- A3b: Alternative or additional option (not in current draft release)
  - "Call-home" – server announces itself (e.g.: "NetConf Call Home Server") and ACP nodes connect to it
  - Model was built for cases/networks where A3a would not work
  - Q: How does SDN controller discover nodes ?
    - A: some discover of nodes needed, but this could come from Topology-discovery of ACP-nodes on RPL root – open

- Good minimum ? Open to expand/change the minimum set of automated service
  - Also want to use this document as showcase for vendors to understand how to easily automatey any more services (standards based or not)

Great! How does it work ?

- ## draft-eckert-anima-grasp-dnssd
  - DNS-SD service discovery across ACP via GRASP
  - GRASP allows to flood objective, this draft proposed a standard to announce/disc over DNS-SD services

- ## draft-eckert-anima-noc-autoconfig
  - Defines required services that ACP nodes (supporting this document) must support, and what to do then:
    - Syslog -> log
    - ntp -> sync time
    - Radius/Diameter -> enable SSH/Netconf
    - Call-home-server

```
[M_FLOOD, 12340815,
    h'fd89b714f3db0000200000064000001', 210000,
  ["SRV.syslog", 4, 255,
     { rfcXXXX: {
         &(sender-loop-count:1) => 255,
         &(srv-element:2) => {
             &(msg-type:1)  => &(describe: 0),
             &(service:2)   => "syslog",
             &(instance:3)  => "east-coast-primary",
             &(priority:5)  => 0,
             &(weight:6)    => 65535,
             &(kvpairs:7)   => { "replicate" => 2 },
             &(range:8)     => 2,
         }
     } }
  ],
  [O_IPv6_LOCATOR,
     h'fd89b714f3db0000200000064000001', TLS12,
514]
]
```

How does it compare to "normal" DNS-SD ?

- Multi-hop flooding of DNS-SD has no good solution ?!
  - Proxy solutions for more constrained network
- Ability to flood allows to measure hop-distance, select nearest server
  - Part of spec draft-eckert-anima-grasp-dnssd
- DNS-SD for GRASP allows kvpairs like unicast/mDNS encoding
  - Added one kvpair "replication" to indicate #servers to connect to simultaneously
  - For redundancy in services: Eg: send syslog messages to 3 best found servers
  - In other services it does not make sense:
    - Connect to always only one Radius server, switch over quickly when the server is unresponsive.

More: How about more autonomic networks ?

- Most ASA will need to auto-discover each other

- Model of service-providing ASA and service-consuming ASA quite generic
    - In this model it makes a lot of sense to utilize the DNS-SD data-mode
        - Service-Name, Service-Parameters
    - Allows to build AF / ASA that are not constrained to run ONLY in ANI
        - Could equally run across other networks
        - Just need some working DNS-SD system

- Aka: ANI components are build re-useable
    - ACP, GRASP, BRSKI can be reused individually

- Relying on DNS-SD for announce/discover make ASA also more independent from ANI
    - Just DNS-SD model for discovery
        - Via GRASP when using ANI, otherwise via mDNS or unicast-DNS/other)
    - GRASP for following p2p negotiations
        - works across any infra, not only ANI

Summary:

- *draft-eckert-anima-noc-autoconfig-00*
  - Make ACP/ANI network automatically use NOC services
  - NOC services announce themselves via DNS-SD
    - ACP/ANI nodes discovery serves, autoconfigure themselves accordingly

  - GRASP/ACP as flooding transport, DNS-SD message encoding.

  - NTP ssh/Netconf, syslog, call-home-server? most key core services ?
    - If not, happy to modify list of key services

  - Should be easy to individually add more and more services specs
- *draft-eckert-anima-grasp-dnssd-01*
  - Enables NOC autoconfig
  - Enables to build ANI independent AF / ASA when relying only on DNS-SD data-model for mutual discovery of ASA (client/server) model
    - (not only option for ASA, but an interesting one!)

Thank You!