An Autonomic Control Plane

draft-ietf-anima-autonomic-control-plane update for ietf99:08 – ietf100:12

100th IETF, November 2017, Singapore

Michael Behringer (editor), Toerless Eckert (editor),
Steinthor Bjarnasson

Activity since prague (ietf99)

- Passed WG last call.
  - Thorough reviews from Brian Carpenter, Sheng Jiang (shepherd), Michael Richardson.
    - Thanks!
  - Discussion on mailing list.
  - Waiting for revised shepherd writeup to get into IETF/IESG review.

Changes since prague (ietf99)

- Introduction:
  - Mentions non-normative section content
  - Defines ANI as BRSKI + ACP + ANI and how ANI itself is not a ”full” autonomic network, but enables it. But ANI also enables also non-autonomic network for stable connectivity.
- Terminology lots of improvements, e.g.:
  - Replaced Autonomic/AN FOOBAR with ACP FOOBAR whenever appropriate
    - Goal: make clear ACP is/can-be standalone, refer to AN/Autonomic only when referring to elements beyond ACP, eg: ANI (BRSKI), other ASA, Intent.
  - device / host / .. -> node. Includes core terms such as Device-ID -> Node-ID.
  - “physical” interface -> “native” interface (also defined). This is uncommon in IETF documents, but IMHO (toerless) there are no physcial interfaces on virtual nodes and ACP can run there too.
  - RFC2119 text (MUST/SHOULD/..)
  - „loopback interface“ – for the interface holding the ACP address(es)
  - „virtual interface“ – for the interface mapped to „secure channels“ (to other ACP nodes)

Changes since prague (ietf99)

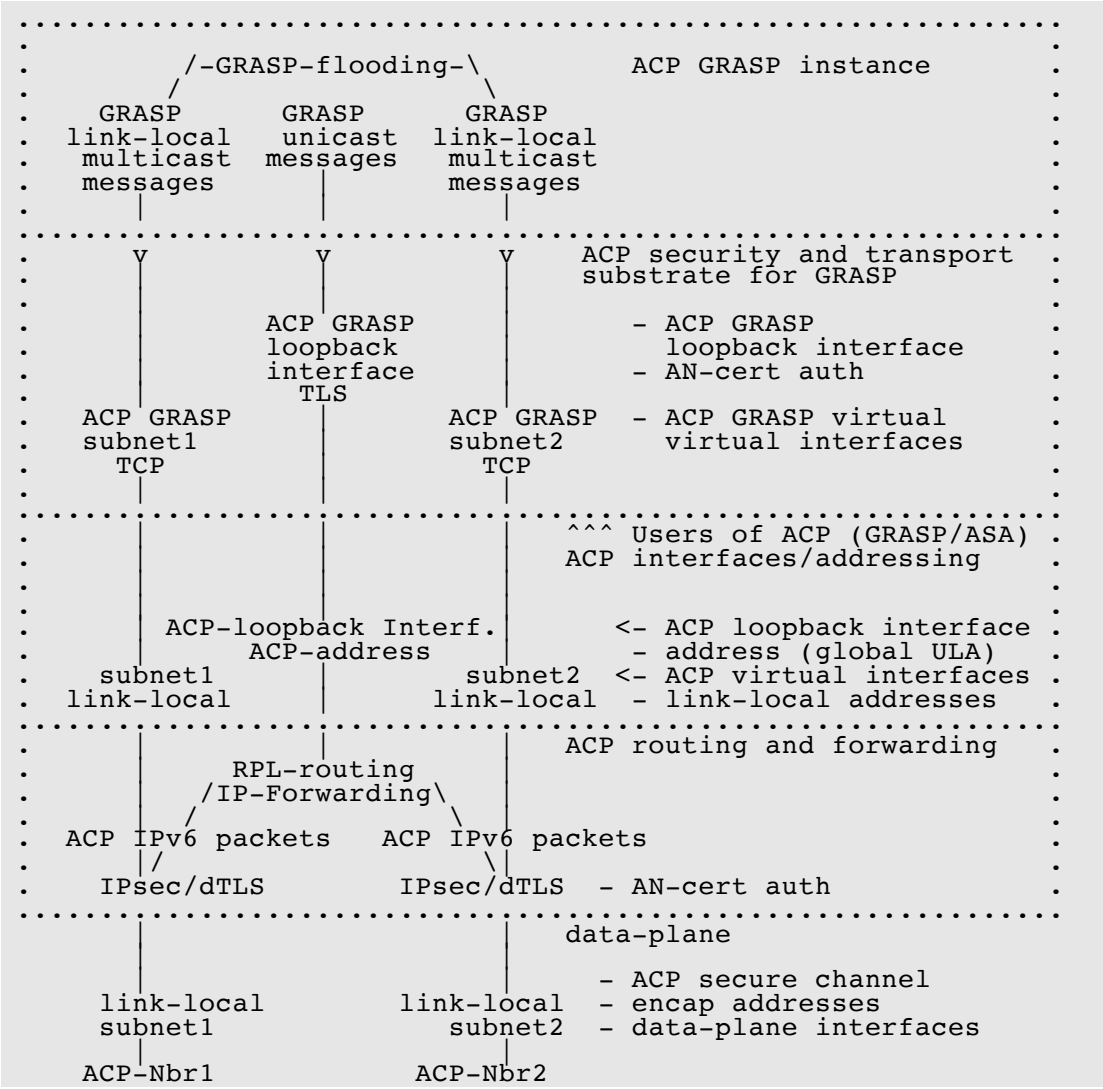
- 6.1 domain certificate / keying material
  - More explanation. relationship to rfc7575. Use of ACP without full autonomic network, Use Domain Cert for any domain authentication (not only ACP secure channels), …
  - Define ACP information field in cert earlier to make text easier readible.
  - 6.1.2 – ACP domain membership
    - Was previously in section 6.6 (candidate ACP Neighbor verification) for authentication of ACP secure channels, moved here because it applies to any domain membership authentication (GRASP TLS connections or future outside ACP domain authentications by ASA).
  - Certificate maintenance (aka.: Cert Renewal)
    - ACP nodes must support cert renewal via EST (rfc7030).
    - GRASP objective name for EST server: SRV.est
    - Removed option for “distance” based server selection. Now in separate draft as future update to GRASPs service discovery
- 6.3 Neighbor Discovery with DULL GRASP
  - Explains need to use MLD for GRASP group (often not mentioned in other IETF RFCs using link-local multicast. No idea how often this leads to bad implementations. Mandated by MLD RFC).

Changes since prague (ietf99)

- 6.7 Security association protocols
  - IPsec tunnel mode required because ACP forwards packets from other nodes (was transport mode in 08). With GRE, IPsec transport mode is used because only the “locally generated” GRE packet (outer header) needs to be Ipsec encapsulated.
  - Immediately terminate/re-negotiate ACP channels when neighbor/own certificate expire.
- 6.8 GRASP in the ACP
  - Explanation why ACP relies on GRASP and does not use IP multicast (toerless pet peeve)
    - Service discovery required as core service for Autonomic network/ANI
      - Provided by ACP via ACP GRASP. lightweight and fully autonomous/distributed
    - IP multicast with PIM-SM or PIM-DM would be horrible and nobody has made this autnomous yet.
      - Chicken & egg problem
    - Flooding via IGPs is alternative to GRASP but ACP choose RPL because it is more lightweight (less state flooded)
    - Could in future ACP update define how to more efficiently do ACP GRASP M_FLOOD by relying on RPL DODAGs.
      - ACP GRASP flooding right now floods on every link, not necessary to reach every node.

Changes since prague (ietf99)

- 6.8.3 ACP as security and transport substrate for GRASP
  - Graphic (requested by sec AD review for GRASP security)
  - Use TLS for GRASP in ACP (was TCP):
  - Provides “some” protection against onpath rogue ACP member.
  - Without TLS, IETF sec recommendations could be that ASA using ACP GRASP would need to encrypt sensitive data negotiated via GRASP between them. ?!
  - But also describe that protection against rogue ACP members is difficult when ASA peer was discovered via GRASP M_FLOOD
    - No unique secure identities of individual nodes currently provided / considered in selecting a GRASP peer



Changes since prague (ietf99)

- 6.10 Addressing inside ACP
  - 6.10.4 Added “Manual” addressing sub-scheme. Primarily for “ACP connect” interfaces
    - Can be used in ACP certs provided to devices that can not participate in ACP secure channels.
  - ACP V8 -> ACP Vlong scheme
    - Now allows to give ACP node 8 or 16 bits of addresses it can use. Eg: Brian/Michael discus that required many addresses
- 6.11 RPL / routing in ACP
  - Explain how profile uses no data-plane artefacts. Results in just “single” DODAG, non-ideal routing when multiple NOCs are used. But data-plane artefacts would require a lot more novel forwarding plane support for all nodes.
  - Establishment of black-hole route for unassigned addresses on each node
  - Logging of packets to unknown destinations at RPL root (NOC).
- 6.12.1 No performance requirements defined
  - Because of wide range of possible deployment options..

Changes since prague (ietf99)

- 6.12.5 ACP interfaces
  - ACP loopback interfaces – hold the ACP addresses
  - ACP virtual interfaces: p2p and multiaccess – map ACP secure channels to ACP VRF (for forwarding of ACP packets)
    - Explains how to do link-local addressing , IPv6 ND on multi-access ACP virtual interfaces
    - Could not find any good RFC reference describing how to map multiple p2p “tunnels” to a multpoint interface
    - Explains how multipoint interface provides more efficient operations (e.g.: flooding).
- 8.1 ACP connect
  - Added sub-sections explaining how it can not only be used as a short term workaround when ACP secure channels are not supported, but also as a way to modular build next-gen NOC devices: Add VM/container supporting ACP, ACP connect interface is virtual internal to the device.
  - Auto-configuration across ACP connect: use SLAAC. Use RFC4191 prefix announcement so that ACP node on ACP connect interface will not become default router but only route ACP prefix.

Changes since prague (ietf99)

- 10.2 New: Diagnostics (informational)
  - Overview of diagnostic (operational Yang model of all components)
  - Proposal to support easier root cause analysis by having step-by-step data-model elements allowing to easier find first problem.
  - Discuss that future work should add more diagnostics to neighbor discovery because as currently defined it is very secure but difficult to diagnose. Argues that certificate is not secret. If announced for diagnostics via DULL GRASP it could easier help diagnostics. (This discussion may not be ideal in ACP but maybe more so in BRSKI.. ?)
- 10.3 enabling/disabling ACP (informational)
  - Filter/drop non-ACP packets by default (make device protected until configured differently)
  - Introduce “admin down” != “physcial down” state to permit running ACP even if data plane is “admin down”
    - Arguing that filtering+admin down is good replacement for physcial down with better survivability/diagnostics
    - Discussing impact on various esablished diagnostics (physcial down to detect remote device)
    - Power level impacts,…
  - Brownfield vs. Greenfield node discussion
    - Global enabling of ACP via explicit config required for brownfield devices
    - Greenfield node: real interesting ANI nodes: only case where ACP globally enabled automatically
  - Interface level ACP enable/disable:
    - Auto-enable “native” (aka: physcial interface) only “automatically”
    - Tunnel interfaces etc. should have explicit “ACP enable” config. Does not introduce another operational step because tunnel interface needs to be created/configured by operator anyhow.

Changes since prague (ietf99)

- 10.8 Adopting ACP for other environments (informational)
  - Discusses how environments where some aspects of ACP are not desirable could create variations with different approaches:
    - Existing auto-addressing schemes for nodes (eg: from existing global unique device IDs).
    - No separation between Data-Plane and ACP. Make ACP the data plane – for new networks ?!
    - Use different routing protocols
    - Use different encapsulation.
      - E.g.: not via link local IPv6 but across L2 to remove data-plane dependency against link-local IPv6.
- More security considerations
- IANA considerations
- 30 new references (thanks Brian ;-)