Internet-Draft Huawei
Intended status: Informational M. Behringer
Expires: February 3, 2018 August 2, 2017

Using Autonomic Control Plane for Stable Connectivity of Network OAM
draft-ietf-anima-stable-connectivity-05

Abstract

OAM (Operations, Administration and Maintenance - as per BCP161, [RFC6291]) processes for data networks are often subject to the problem of circular dependencies when relying on connectivity provided by the network to be managed for the OAM purposes.

Provisioning while bringing up devices and networks tends to be more difficult to automate than service provisioning later on, changes in core network functions impacting reachability cannot be automated because of ongoing connectivity requirements for the OAM equipment itself, and widely used OAM protocols are not secure enough to be carried across the network without security concerns.

This document describes how to integrate OAM processes with the autonomic control plane (ACP) in Autonomic Networks (AN) in order to provide stable and secure connectivity for those OAM processes. This connectivity is not subject to aforementioned circular dependencies.

Table of Contents

1. Introduction

1.1. Self dependent OAM Connectivity

OAM (Operations, Administration and Maintenance - as per BCP161, [RFC6291]) for data networks is often subject to the problem of circular dependencies when relying on the connectivity service

provided by the network to be managed. OAM can easily but unintentionally break the connectivity required for its own operations. Avoiding these problems can lead to complexity in OAM. This document describes this problem and how to use the Autonomic Control Plane (ACP) to solve it without further OAM complexity:

The ability to perform OAM on a network device requires first the execution of OAM necessary to create network connectivity to that device in all intervening devices. This typically leads to sequential, 'expanding ring configuration' from a NOC (Network Operations Center). It also leads to tight dependencies between provisioning tools and security enrollment of devices. Any process that wants to enroll multiple devices along a newly deployed network topology needs to tightly interlock with the provisioning process that creates connectivity before the enrollment can move on to the next device.

When performing change operations on a network, it likewise is necessary to understand at any step of that process that there is no interruption of connectivity that could lead to removal of connectivity to remote devices. This includes especially change provisioning of routing, forwarding, security and addressing policies in the network that often occur through mergers and acquisitions, the introduction of IPv6 or other mayor re-hauls in the infrastructure design. Examples include change of an IGP or areas, PA (Provider Aggregatabe) to PI (Provider Independent) addressing, or systematic topology changes (such as L2 to L3 changes).

All these circular dependencies make OAM complex and potentially fragile. When automation is being used, for example through provisioning systems, this complexity extends into that automation software.

1.2. Data Communication Networks (DCNs)

In the late 1990'th and early 2000, IP networks became the method of choice to build separate OAM networks for the communications infrastructure within Network Providers. This concept was standardized in ITU-T G.7712/Y.1703 [ITUT] and called "Data Communications Networks" (DCN). These where (and still are) physically separate IP(/MPLS) networks that provide access to OAM interfaces of all equipment that had to be managed, from PSTN (Public Switched Telephone Network) switches over optical equipment to nowadays Ethernet and IP/MPLS production network equipment.

Such DCN provide stable connectivity not subject to aforementioned problems because they are separate network entirely, so change configuration of the production IP network is done via the DCN but

never affects the DCN configuration. Of course, this approach comes at a cost of buying and operating a separate network and this cost is not feasible for many providers, most notably smaller providers, most enterprises and typical IoT networks (Internet of Things).

1.3. Leveraging the ACP

One of the goals of the Autonomic Networks Autonomic Control Plane (ACP as defined in [I-D.ietf-anima-autonomic-control-plane]) is to provide similar stable connectivity as a DCN, but without having to build a separate DCN. It is clear that such 'in-band' approach can never achieve fully the same level of separation, but the goal is to get as close to it as possible.

This solution approach has several aspects. One aspect is designing the implementation of the ACP in network devices to make it actually perform without interruption by changes in what we will call in this document the "data-plane", a.k.a: the operator or controller configured services planes of the network equipment. This aspect is not currently covered in this document.

Another aspect is how to leverage the stable IPv6 connectivity provided by the ACP for OAM purposes. This is the current scope of this document.

2. Solutions

2.1. Stable Connectivity for Centralized OAM

The ANI is the "Autonomic Networking Infrastructure" consisting of secure zero touch Bootstrap (BRSKI - [I-D.ietf-anima-bootstrapping-keyinfra]), GeneRic Autonomic Signaling Protocol (GRASP - [I-D.ietf-anima-grasp]), and Autonomic Control Plane (ACP - [I-D.ietf-anima-autonomic-control-plane]). Refer to [I-D.ietf-anima-reference-model] for an overview of the ANI and how its components interact and [RFC7575] for concepts and terminology of ANI and autonomic networks.

This section describes stable connectivity for centralized OAM via ACP/ANI starting by what we expect to be the most easy to deploy short-term option. It then describes limitation and challenges of that approach and their solutions/workarounds to finish with the preferred target option of autonomic NOC devices in Section 2.1.6.

This order was chosen because it helps to explain how simple initial use of ACP can be, how difficult workarounds can become (and therefore what to avoid), and finally because one very promising

long-term solution alternative is exactly like the most easy short-term solution only virtualized and automated.

In the most common case, OAM will be performed by one or more applications running on a variety of centralized NOC systems that communicate with network devices. We describe differently advanced approaches to leverage the ACP for stable connectivity. There is a wide range of options, some of which are simple, some more complex.

Three stages can be considered:

o There are simple options described in sections Section 2.1.1 through Section 2.1.3 that we consider to be good starting points to operationalize the use of the ACP for stable connectivity today. These options require only network and OAN/NOC device configuration.

o The are workarounds to connect the ACP to non-IPv6 capable NOC devices through the use of IPv4/IPv6 NAT (Network Address Translation) as described in section Section 2.1.4. These workarounds are not recommended but if such non-IPv6 capable NOC devices need to be used longer term, then this is the only option to connect them to the ACP.

o Near to long term options can provide all the desired operational, zero touch and security benefits of an autonomic network, but a range of details for this still have to be worked out and development work on NOC/OAM equipment is necessary. These options are discussed in sections Section 2.1.5 through Section 2.1.8.

2.1.1. Simple Connectivity for Non-ACP capable NMS Hosts

In the most simple candidate deployment case, the ACP extends all the way into the NOC via one or more "ACP edge devices" as defined in section 6.1 of [I-D.ietf-anima-autonomic-control-plane]. These devices "leak" the (otherwise encrypted) ACP natively to NMS hosts. They acts as the default router to those NMS hosts and provide them with IPv6 connectivity into the ACP. NMS hosts with this setup need to support IPv6 (see e.g. [RFC6434]) but require no other modifications to leverage the ACP.

Note that even though the ACP only uses IPv6, it can of course support OAM for any type of network deployment as long as the network devices support the ACP: The Data Plane can be IPv4 only, dual-stack or IPv6 only. It is always spearate from the ACP, therefore there is no dependency between the ACP and the IP version(s) used in the Data Plane.

This setup is sufficient for troubleshooting such as SSH into network devices, NMS that performs SNMP read operations for status checking, software downloads into autonomic devices, provisioning of devices via NETCONF and so on. In conjunction with otherwise unmodified OAM via separate NMS hosts it can provide a good subset of the stable connectivity goals. The limitations of this approach are discussed in the next section.

Because the ACP provides 'only' for IPv6 connectivity, and because addressing provided by the ACP does not include any topological addressing structure that operations in a NOC often relies on to recognize where devices are on the network, it is likely highly desirable to set up DNS (Domain Name System - see [RFC1034]) so that the ACP IPv6 addresses of autonomic devices are known via domain names that include the desired structure. For example, if DNS in the network was set up with names for network devices as devicename.noc.example.com, and the well known structure of the Data Plane IPv4 addresses space was used by operators to infer the region where a device is located in, then the ACP address of that device could be set up as devicename_<region>.acp.noc.example.com, and devicename.acp.noc.example.com could be a CNAME to devicename_<region>.acp.noc.example.com. Note that many networks already use names for network equipment where topological information is included, even without an ACP.

2.1.2. Challenges and Limitation of Simple Connectivity

This simple connectivity of non-autonomic NMS hosts suffers from a range of challenges (that is, operators may not be able to do it this way) or limitations (that is, operator cannot achieve desired goals with this setup). The following list summarizes these challenges and limitations. The following sections describe additional mechanisms to overcome them.

Note that these challenges and limitations exist because ACP is primarily designed to support distributed ASA (Autonomic Service Agent, a piece of autonomic software) in the most lightweight fashion, but not mandatorily require support for additional mechanisms to best support centralized NOC operations. It is this document that describes additional (short term) workarounds and (long term) extensions.

1. (Limitation) NMS hosts cannot directly probe whether the desired so called 'data-plane' network connectivity works because they do not directly have access to it. This problem is similar to probing connectivity for other services (such as VPN services) that they do not have direct access to, so the NOC may already

employ appropriate mechanisms to deal with this issue (probing proxies). See Section 2.1.3 for candidate solutions.

2. (Challenge) NMS hosts need to support IPv6 which often is still not possible in enterprise networks. See Section 2.1.4 for some workarounds.

3. (Limitation) Performance of the ACP will be limited versus normal 'data-plane' connectivity. The setup of the ACP will often support only non-hardware accelerated forwarding. Running a large amount of traffic through the ACP, especially for tasks where it is not necessary will reduce its performance/ effectiveness for those operations where it is necessary or highly desirable. See Section 2.1.5 for candidate solutions.

4. (Limitation) Security of the ACP is reduced by exposing the ACP natively (and unencrypted) into a subnet in the NOC where the NOC devices are attached to it. See Section 2.1.7 for candidate solutions.

These four problems can be tackled independently of each other by solution improvements. Combining some of these solutions improvements together can lead towards a candiate long term solution.

2.1.3. Simultaneous ACP and Data Plane Connectivity

Simultaneous connectivity to both ACP and data-plane can be achieved in a variety of ways. If the data-plane is IPv4-only, then any method for dual-stack attachment of the NOC device/application will suffice: IPv6 connectivity from the NOC provides access via the ACP, IPv4 will provide access via the data-plane. If as explained above in the simple case, an autonomic device supports native attachment to the ACP, and the existing NOC setup is IPv4 only, then it could be sufficient to attach the ACP device(s) as the IPv6 default router to the NOC subnet and keep the existing IPv4 default router setup unchanged.

If the data-plane of the network is also supporting IPv6, then the most compatible setup for NOC devices is to have two IPv6 interfaces. One virtual ((e.g. via IEEE 802.1Q [IEEE802.1Q]) or physical interface connecting to a data-plane subnet, and another into an ACP connect subnet as specified in the ACP connection section of [I-D.ietf-anima-autonomic-control-plane]. That document also specifies how the NOC devices can receive autoconfigured addressing and routes towards the ACP connect subnet if it supports [RFC6724] and [RFC4191].

Configuring a second interface on a NOC host may be impossible or be seen as undesired complexity. In that case the ACP edge device needs to provide support for a "Combined ACP and Data Plane interface" as also described in the ACP connect section of [I-D.ietf-anima-autonomic-control-plane]. This setup may not work with autoconfiguration and all NOC host network stacks due to limitations in those network stacks. They need to be able to perform RFC6724 source address selection rule 5.5 including caching of next-hop information. See the ACP document text for more details.

For security reasons, it is not considered appropriate in the ACP document to connect a non-ACP router to an ACP connect interface. The reason is that the ACP is a secured network domain and all NOC devices connecting via ACP connect interfaces are also part of that secure domain - the main difference is that the physical link between the ACP edge device and the NOC devices is not authenticated/ encrypted and therefore needs to be physically secured. If the secure ACP was extendable via untrusted routers then it would be a lot more verify the secure domain assertion. Therefore the ACP edge devices are not supposed to redistribute routes from non-ACP routers into the ACP.

2.1.4. IPv4-only NMS Hosts

ACP does not support IPv4: Single stack IPv6 management of the network via ACP and (as needed) data plane. Independent of whether the data plane is dual-stack, has IPv4 as a service or is single stack IPv6. Dual plane management, IPv6 for ACP, IPv4 for the data plane is likewise an architecturally simple option.

The implication of this architectural decision is the potential need for short-term workarounds when the operational practices in a network do not yet meet these target expectations. This section explains when and why these workarounds may be operationally necessary and describes them. However, the long term goal is to upgrade all NMS hosts to native IPv6, so the workarounds described in this section should not be considered permanent.

Most network equipment today supports IPv6 but it is by far not ubiquitously supported in NOC backend solutions (HW/SW), especially not in the product space for enterprises. Even when it is supported, there are often additional limitations or issues using it in a dual stack setup or the operator mandates for simplicity single stack for all operations. For these reasons an IPv4 only management plane is still required and common practice in many enterprises. Without the desire to leverage the ACP, this required and common practice is not a problem for those enterprises even when they run dual stack in the

network. We discuss these workarounds here because it is a short term deployment challenge specific to the operations of the ACP.

To connect IPv4 only management plane devices/applications with the ACP, some form of IP/ICMP translation of packets IPv4<->IPv6 is necessary. The basic mechanisms for this are defined in SIIT ([RFC7915]). There are multiple solutions using this mechanisms. To understand the possible solutions, we consider the requirements:

1. NMS hosts need to be able to initiate connections to any ACP device for management purposes. Examples include provisioning via Netconf/(SSH), SNMP poll operations or just diagnostics via SSH connections from operators. Every ACP device/function that needs to be reachable from NMS hosts needs to have a separate IPv4 address.

2. ACP devices need to be able to initiate connections to NMS hosts for example to initiate NTP or radius/diameter connections, send syslog or SNMP trap or initiate Netconf Call Home connections after bootstrap. Every NMS host needs to have a separate IPv6 address reachable from the ACP. When connections from ACP devices are made to NMS hosts, the IPv4 source address of these connections as seen by the NMS Host must also be unique per ACP device and the same address as in (1) to maintain the same addressing simplicity as in a native IPv4 deployment. For example in syslog, the source-IP address of a logging device is used to identify it, and if the device shows problems, an operator might want to SSH into the device to diagnose it.

Because of these requirements, the necessary and sufficient set of solutions are those that provide 1:1 mapping of IPv6 ACP addresses into IPv4 space and 1:1 mapping of IPv4 NMS host space into IPv6 (for use in the ACP). This means that stateless SIIT based solutions are sufficient and preferred.

Note that ACP devices may use multiple IPv6 addresses in the ACP based on which Sub-Scheme they use. For example in the Zone Sub-Scheme, an ACP device could use two addresses, one with the last address bit (V-bit) 0 and one with 1. Both addresses may need to be reachable thought the IPv6/IPv4 address translation.

The need to allocate for every ACP device one or multiple IPv4 addresses should not be a problem if - as we assume - the NMS hosts can use private IPv4 address space ([RFC1918]). Nevertheless even with RFC1918 address space it is important that the ACP IPv6 addresses can efficiently be mapped into IPv4 address space without too much waste.

The currently most flexible mapping scheme to achieve this is [RFC7757] because it allows configured IPv4 <-> IPv6 prefix mapping. Assume the ACP uses the Zone Addressing Sub-Scheme and there are 3 registrars. In the Zone Addressing Sub-Scheme, there is for each registrar a constant /112 prefix for which in RFC7757 an EAM (Explicit Address Mapping) into a /16 (eg: RFC1918) prefix into IPv4 can be configured. Within the registrars /112 prefix, Device-Numbers for devices are sequentially assigned: with V-bit effectively two numbers are assigned per ACP device. This also means that if IPv4 address space is even more constrained, and it is known that a registrar will never need the full /15 extent of Device-Numbers, then a longer than /112 prefix can be configured into the EAM to use less IPv4 space.

When using the Vlong Addressing Sub-Scheme, it is unlikely that one wants or need to translate the full /8 or /16 bits of addressing space per ACP device into IPv4. In this case, the EAM rules of dropping trailing bits can be used to map only N bits of the V-bits into IPv4. This does imply though that only V-addresses that differ in those high-order N V-bits can be distinguished on the IPv4 side.

Likewise, the IPv4 address space used for NMS hosts can easily be mapped into an ACP prefix assigned to an ACP connect interface.

A full specification of a solution to perform SIIT in conjunction with ACP connect following the considerations below is outside the scope of this document.

To be in compliance with security expectations, SIIT has to to happen on the ACP edge device itself so that ACP security considerations can be taken into account. Eg: that IPv4 only NMS hosts can be dealt with exactly like IPv6 hosts connected to an ACP connect interface.

Note that prior solutions such as NAT64 ([RFC6146]) may equally be useable to translate between ACP IPv6 address space and NMS Hosts IPv4 address space, and that as workarounds this can also be done on non ACP Edge Devices connected to an ACP connect interface. The details vary depending on implementation because the options to configure address mappings vary widely. Outside of EAM, there are no standardized solutions that allow for mapping of prefixes, so it will most likely be necessary to explicitly map every individual (/128) ACP device address to an IPv4 address. Such an approach should use automation/scripting where these address translation entries are created dynamically whenever an ACP device is enrolled or first connected to the ACP network.

Overall, the use of NAT is especially subject to the ROI (Return On Investment) considerations, but the methods described here may not be

too different from the same problems encountered totally independent of AN/ACP when some parts of the network are to introduce IPv6 but NMS hosts are not (yet) upgradeable.

2.1.5. Path Selection Policies

As mentioned above, the ACP is not expected to have high performance because its primary goal is connectivity and security, and for existing network device platforms this often means that it is a lot more effort to implement that additional connectivity with hardware acceleration than without - especially because of the desire to support full encryption across the ACP to achieve the desired security.

Some of these issues may go away in the future with further adoption of the ACP and network device designs that better tender to the needs of a separate OAM plane, but it is wise to plan for even long-term designs of the solution that does NOT depend on high-performance of the ACP. This is opposite to the expectation that future NMS hosts will have IPv6, so that any considerations for IPv4/NAT in this solution are temporary.

To solve the expected performance limitations of the ACP, we do expect to have the above describe dual-connectivity via both ACP and data-plane between NOC application devices and AN devices with ACP. The ACP connectivity is expected to always be there (as soon as a device is enrolled), but the data-plane connectivity is only present under normal operations but will not be present during e.g. early stages of device bootstrap, failures, provisioning mistakes or during network configuration changes.

The desired policy is therefore as follows: In the absence of further security considerations (see below), traffic between NMS hosts and AN devices should prefer data-plane connectivity and resort only to using the ACP when necessary, unless it is an operation known to be so much tied to the cases where the ACP is necessary that it makes no sense to try using the data plane. An example here is of course the SSH connection from the NOC into a network device to troubleshoot network connectivity. This could easily always rely on the ACP. Likewise, if an NMS host is known to transmit large amounts of data, and it uses the ACP, then its performance need to be controlled so that it will not overload the ACP performance. Typical examples of this are software downloads.

There is a wide range of methods to build up these policies. We describe a few:

Ideally, a NOC system would learn and keep track of all addresses of a device (ACP and the various data plane addresses). Every action of the NOC system would indicate via a "path-policy" what type of connection it needs (e.g. only data-plane, ACP-only, default to data-plane, fallback to ACP,...). A connection policy manager would then build connection to the target using the right address(es). Shorter term, a common practice is to identify different paths to a device via different names (e.g. loopback vs. interface addresses). This approach can be expanded to ACP uses, whether it uses NOC system local names or DNS. We describe example schemes using DNS:

DNS can be used to set up names for the same network devices but with different addresses assigned: One name (name.noc.example.com) with only the data-plane address(es) (IPv4 and/or IPv6) to be used for probing connectivity or performing routine software downloads that may stall/fail when there are connectivity issues. One name (name-acp.noc.example.com) with only the ACP reachable address of the device for troubleshooting and probing/discovery that is desired to always only use the ACP. One name with data plane and ACP addresses (name-both.noc.example.com).

Traffic policing and/or shaping of at the ACP edge in the NOC can be used to throttle applications such as software download into the ACP.

MPTCP (Multipath TCP -see [RFC6824]) is a very attractive candidate to automate the use of both data-plane and ACP and minimize or fully avoid the need for the above mentioned logical names to pre-set the desired connectivity (data-plane-only, ACP only, both). For example, a set-up for non MPTCP aware applications would be as follows:

DNS naming is set up to provide the ACP IPv6 address of network devices. Unbeknownst to the application, MPTCP is used. MPTCP mutually discovers between the NOC and network device the data-plane address and caries all traffic across it when that MPTCP subflow across the data-plane can be built.

In the Autonomic network devices where data-plane and ACP are in separate VRFs, it is clear that this type of MPTCP subflow creation across different VRFs is new/added functionality. Likewise, the policies of preferring a particular address (NOC-device) or VRF (AN device) for the traffic is potentially also a policy not provided as a standard.

2.1.6. Autonomic NOC Device/Applications

Setting up connectivity between the NOC and autonomic devices when the NOC device itself is non-autonomic is as mentioned in the beginning a security issue. It also results as shown in the previous

paragraphs in a range of connectivity considerations, some of which may be quite undesirable or complex to operationalize.

Making NMS hosts autonomic and having them participate in the ACP is therefore not only a highly desirable solution to the security issues, but can also provide a likely easier operationalization of the ACP because it minimizes NOC-special edge considerations - the ACP is simply built all the way automatically, even inside the NOC and only authorized and authenticate NOC devices/applications will have access to it.

Supporting the ACP all the way into an application device requires implementing the following aspects in it: AN bootstrap/enrollment mechanisms, the secure channel for the ACP and at least the host side of IPv6 routing setup for the ACP. Minimally this could all be implemented as an application and be made available to the host OS via e.g. a tap driver to make the ACP show up as another IPv6 enabled interface.

Having said this: If the structure of NMS hosts is transformed through virtualization anyhow, then it may be considered equally secure and appropriate to construct (physical) NMS host system by combining a virtual AN/ACP enabled router with non-AN/ACP enabled NOC-application VMs via a hypervisor, leveraging the configuration options described in the previous sections but just virtualizing them.

2.1.7. Encryption of data-plane connections

When combining ACP and data-plane connectivity for availability and performance reasons, this too has an impact on security: When using the ACP, the traffic will be mostly encryption protected, especially when considering the above described use of AN application devices. If instead the data-plane is used, then this is not the case anymore unless it is done by the application.

The simplest solution for this problem exists when using AN capable NMS hosts, because in that case the communicating AN capable NMS host and the AN network device have certificates through the AN enrollment process that they can mutually trust (same AN domain). In result, data-plane connectivity that does support this can simply leverage TLS/DTLS ([RFC5246]/[RFC6347]) with mutual AN-domain certificate authentication - and does not incur new key management.

If this automatic security benefit is seen as most important, but a "full" ACP stack into the NMS host is unfeasible, then it would still be possible to design a stripped down version of AN functionality for such NOC hosts that only provides enrollment of the NOC host into the

AN domain to the extent that the host receives an AN domain certificate, but without directly participating in the ACP afterwards. Instead, the host would just leverage TLS/DTLS using its AN certificate via the data-plane with AN network devices as well as indirectly via the ACP with the above mentioned in-NOC network edge connectivity into the ACP.

When using the ACP itself, TLS/DTLS for the transport layer between NMS hosts and network device is somewhat of a double price to pay (ACP also encrypts) and could potentially be optimized away, but given the assumed lower performance of the ACP, it seems that this is an unnecessary optimization.

2.1.8. Long Term Direction of the Solution

If we consider what potentially could be the most lightweight and autonomic long term solution based on the technologies described above, we see the following direction:

1. NMS hosts should at least support IPv6. IPv4/IPv6 NAT in the network to enable use of ACP is long term undesirable. Having IPv4 only applications automatically leverage IPv6 connectivity via host-stack translation may be an option but this has not been investigated yet.

2. Build the ACP as a lightweight application for NMS hosts so ACP extends all the way into the actual NMS hosts.

3. Leverage and as necessary enhance MPTCP with automatic dual-connectivity: If an MPTCP unaware application is using ACP connectivity, the policies used should add subflow(s) via the data-plane and prefer them.

4. Consider how to best map NMS host desires to underlying transport mechanisms: With the above mentioned 3 points, not all options are covered. Depending on the OAM, one may still want only ACP, only data-plane, or automatically prefer one over the other and/or use the ACP with low performance or high-performance (for emergency OAM such as countering DDoS). It is as of today not clear what the simplest set of tools is to enable explicitly the choice of desired behavior of each OAM. The use of the above mentioned DNS and MPTCP mechanisms is a start, but this will require additional thoughts. This is likely a specific case of the more generic scope of TAPS.

2.2. Stable Connectivity for Distributed Network/OAM

The ANI (ACP, Bootstrap, GRASP) can provide via the GRASP protocol common direct-neighbor discovery and capability negotiation (GRASP via ACP and/or data-plane) and stable and secure connectivity for functions running distributed in network devices (GRASP via ACP). It can therefore eliminate the need to re-implement similar functions in each distributed function in the network. Today, every distributed protocol does this with functional elements usually called "Hello" mechanisms and with often protocol specific security mechanisms.

KARP (Keying and Authentication for Routing Protocols, see [RFC6518]) has tried to start provide common directions and therefore reduce the re-invention of at least some of the security aspects, but it only covers routing-protocols and it is unclear how well it applicable to a potentially wider range of network distributed agents such as those performing distributed OAM. The ACP can help in these cases.

3. Architectural Considerations

3.1. No IPv4 for ACP

The ACP is targeted to be IPv6 only, and the prior explanations in this document show that this can lead to some complexity when having to connect IPv4 only NOC solutions, and that it will be impossible to leverage the ACP when the OAM agents on an ACP network device do not support IPv6. Therefore, the question was raised whether the ACP should optionally also support IPv4.

The decision not to include IPv4 for ACP as something that is considered in the use cases in this document is because of the following reasons:

In SP networks that have started to support IPv6, often the next planned step is to consider moving out IPv4 from a native transport as just a service on the edge. There is no benefit/need for multiple parallel transport families within the network, and standardizing on one reduces OPEX and improves reliability. This evolution in the data plane makes it highly unlikely that investing development cycles into IPv4 support for ACP will have a longer term benefit or enough critical short-term use-cases. Support for IPv4-only for ACP is purely a strategic choice to focus on the known important long term goals.

In other type of networks as well, we think that efforts to support autonomic networking is better spent in ensuring that one address family will be support so all use cases will long-term work with it, instead of duplicating effort into IPv4. Especially because auto-

addressing for the ACP with IPv4 would be more complex than in IPv6 due to the IPv4 addressing space.

4. Security Considerations

In this section, we discuss only security considerations not covered in the appropriate sub-sections of the solutions described.

Even though ACPs are meant to be isolated, explicit operator misconfiguration to connect to insecure OAM equipment and/or bugs in ACP devices may cause leakage into places where it is not expected. Mergers/Acquisitions and other complex network reconfigurations affecting the NOC are typical examples.

ACP prefix addresses are ULA addresses. Using these addresses also for NOC devices as proposed in this document is not only necessary for above explained simple routing functionality but it is also more secure than global IPv6 addresses. ULA addresses are not routed in the global Internet and will therefore be subject to more filtering even in places where specific ULA addresses are being used. Packets are therefore less likely to leak to be successfully injected into the isolated ACP environment.

The random nature of a ULA prefix provides strong protection against address collision even though there is no central assignment authority. This is helped by the expectation, that ACPs are never expected to connect all together, but only few ACPs may ever need to connect together, e.g. when mergers and aquisitions occur.

The ACP specification demands that only packets from configured ACP prefixes are permitted from ACP connect interfaces. It also requires that RPL root ACP devices need to be able to diagnose unknown ACP destination addresses.

To help diagnose packets that unexpectedly leaked for example from another ACP (that was meant to be deployed separately), it can be useful to voluntarily list your own the ULA ACP prefixes in one of the sites on the Internet, for example https://www.sixxs.net/tools/grh/ula/. Note that this does not constitute registration and if you want to ensure that your leaked ACP packets can be recognized to come from you, you may need to list your prefixes in multiple of those sites.

Note that there is a provision in [RFC4193] for non-locally assigned address space (L bit = 0), but there is no existing standardization for this, so these ULA prefixes must not be used.

According to RFC4193 section 4.4, PTR records for ULA addresses should not be installed into the global DNS (no guaranteed ownership). Hence also the need to rely on voluntary lists (and in prior paragraph) to make the use of an ULA prefix globally known.

Nevertheless, some legacy OAM applications running across the ACP may rely on reverse DNS lookup for authentication of requests (eg: TFTP for download of network firmware/config/software). Operators may therefore use split horizon DNS to provide global PTR records for their own ULA prefixes only into their own domain to continue relying on this method. Given the security of the ACP, this may even increase the security of such legacy methods.

Any current and future protocols must rely on secure end-to-end communications (TLS/DTLS) and identification and authentication via the certificates assigned to both ends. This is enabled by the certificate mechanisms of the ACP.

If DNS and especially reverse DNS are set up, then it should be set up in an automated fashion, linked to the autonomic registrar backend so that the DNS and reverse DNS records are actually derived from the subject name elements of the ACP device certificates in the same way as the autonomic devices themselves will derive their ULA addresses from their certificates to ensure correct and consistent DNS entries.

If an operator feels that reverse DNS records are beneficial to its own operations but that they should not be made available publically for "security" by concealment reasons, then the case of ACP DNS entries is probably one of the least problematic use cases for split-DNS: The ACP DNS names are only needed for the NMS hosts intending to use the ACP - but not network wide across the enterprise.

5. IANA Considerations

This document requests no action by IANA.

6. Acknowledgements

This work originated from an Autonomic Networking project at cisco Systems, which started in early 2010 including customers involved in the design and early testing. Many people contributed to the aspects described in this document, including in alphabetical order: BL Balaji, Steinthor Bjarnason, Yves Herthoghs, Sebastian Meissner, Ravi Kumar Vadapalli. The author would also like to thank Michael Richardson, James Woodyatt and Brian Carpenter for their review and comments. Special thanks to Sheng Jiang and Mohamed Boucadair for their thorough review.

7. Change log [RFC Editor: Please remove]

05: Integrated fixes from Brian Carpenters review. Details on semantic/structural changes:

* Folded most comments back into draft-ietf-anima-autonomic-control-plane-09 because this stable connectivity draft was suggesting things that are better written out and standardized in the ACP document.

* Section on simultaneous ACP and data plane connectivity section reduced/rewritten because of this.

* Re-emphasized security model of ACP - ACP-connect can not arbitrarily extend ACP routing domain.

* Re-wrote much of NMS section to be less suggestive and more descriptive, avoiding the term NAT and referring to relevant RFCs (SIIT etc.).

* Main additional text in IPv4 section is about explaining how we suggest to use EAM (Explicit Address Mapping) which actuall would well work with the Zone and Vlong Addressing Sub-Schemes of ACP.

* Moved, but not changed section of "why no IPv4 in ACP" before IANA considerations to make structure of document more logical.

* Refined security considerations: explained how optional ULA prefix listing on Internet is only for diagnostic purposes. Explained how this is useful because DNS must not be used. Explained how split horizon DNS can be used nevertheless.

04: Integrated fixes from Mohamed Boucadairs review (thorough textual review).

03: Integrated fixes from thorough Shepherd review (Sheng Jiang).

01: Refresh timeout. Stable document, change in author association.

01: Refresh timeout. Stable document, no changes.

00: Changed title/dates.

individual-02: Updated references.

individual-03: Modified ULA text to not suggest ULA-C as much better anymore, but still mention it.

individual-02: Added explanation why no IPv4 for ACP.

individual-01: Added security section discussing the role of address prefix selection and DNS for ACP. Title change to emphasize focus on OAM. Expanded abstract.

individual-00: Initial version.

8. References

[I-D.ietf-anima-autonomic-control-plane]
Behringer, M., Eckert, T., and S. Bjarnason, "An Autonomic Control Plane (ACP)", draft-ietf-anima-autonomic-control-plane-08 (work in progress), July 2017.

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-07 (work in progress), July 2017.

[I-D.ietf-anima-grasp]
Bormann, C., Carpenter, B., and B. Liu, "A Generic Autonomic Signaling Protocol (GRASP)", draft-ietf-anima-grasp-15 (work in progress), July 2017.

[I-D.ietf-anima-reference-model]
Behringer, M., Carpenter, B., Eckert, T., Ciavaglia, L., Pierre, P., Liu, B., Nobre, J., and J. Strassner, "A Reference Model for Autonomic Networking", draft-ietf-anima-reference-model-04 (work in progress), July 2017.

[IEEE802.1Q]
International Telecommunication Union, "802.1Q-2014 - IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks", 2014.

[ITUT] International Telecommunication Union, "Architecture and specification of data communication network", ITU-T Recommendation G.7712/Y.1703, Noevember 2001.

This is the earliest but superceeded version of the series. See REC-G.7712 Home Page [1] for current versions.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <http://www.rfc-editor.org/info/rfc1918>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <http://www.rfc-editor.org/info/rfc4191>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <http://www.rfc-editor.org/info/rfc4193>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <http://www.rfc-editor.org/info/rfc6146>.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <http://www.rfc-editor.org/info/rfc6291>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC6418] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, DOI 10.17487/RFC6418, November 2011, <http://www.rfc-editor.org/info/rfc6418>.

[RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, DOI 10.17487/RFC6434, December 2011, <http://www.rfc-editor.org/info/rfc6434>.

[RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, DOI 10.17487/RFC6518, February 2012, <http://www.rfc-editor.org/info/rfc6518>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <http://www.rfc-editor.org/info/rfc6724>.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <http://www.rfc-editor.org/info/rfc6824>.

[RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, DOI 10.17487/RFC7575, June 2015, <http://www.rfc-editor.org/info/rfc7575>.

[RFC7757] Anderson, T. and A. Leiva Popper, "Explicit Address Mappings for Stateless IP/ICMP Translation", RFC 7757, DOI 10.17487/RFC7757, February 2016, <http://www.rfc-editor.org/info/rfc7757>.

[RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <http://www.rfc-editor.org/info/rfc7915>.

Authors' Addresses

Toerless Eckert (editor)
Futurewei Technologies Inc.
2330 Central Expy
Santa Clara 95050
USA

Email: tte+ietf@cs.fau.de

Michael H. Behringer

Email: michael.h.behringer@gmail.com