Internet Engineering Task Force (IETF)
Request for Comments: 8368
Category: Informational
ISSN: 2070-1721

T. Eckert, Ed.
Huawei
M. Behringer

May 2018

Using an Autonomic Control Plane for Stable Connectivity of Network Operations, Administration, and Maintenance (OAM)

Abstract

Operations, Administration, and Maintenance (OAM), as per BCP 161, for data networks is 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 an autonomic control plane in order to provide stable and secure connectivity for those OAM processes. This connectivity is not subject to the aforementioned circular dependencies.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8368.

Table of Contents

1. Introduction

1.1. Self-Dependent OAM Connectivity

Operations, Administration, and Maintenance (OAM), as per BCP 161 [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 an autonomic control plane 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 Network Operations Center (NOC). 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.

Likewise, when performing change operations on a network, it 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 major overhauls of the infrastructure design. Examples include change of an IGP or area, change from Provider Aggregatable (PA) to Provider Independent (PI) addressing, or systematic topology changes (such as Layer 2 to Layer 3 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 1990s 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_G7712] and called "Data Communications Networks" (DCNs). These were (and still are)

physically separate IP or IP/MPLS networks that provide access to OAM interfaces of all equipment that had to be managed, from Public Switched Telephone Network (PSTN) switches over optical equipment to nowadays Ethernet and IP/MPLS production network equipment.

Such DCNs provide stable connectivity not subject to the aforementioned problems because they are a 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 Internet of Things (IoT) networks.

1.3. Leveraging a Generalized Autonomic Control Plane

One of the goals of the IETF ANIMA (Autonomic Networking Integrated Model and Approach) Working Group is the specification of a secure and automatically built in-band management plane that provides stable connectivity similar to a DCN, but without having to build a separate DCN. It is clear that such an "in-band" approach can never fully achieve the same level of separation, but the goal is to get as close to it as possible.

This document discusses how such an in-band management plane can be used to support the DCN-like OAM use case, how to leverage its stable connectivity, and what the options are for deploying it incrementally in the short and long term.

The ANIMA Working Group’s evolving specification [ACP] calls this in-band management plane the "Autonomic Control Plane" (ACP). The discussions in this document are not dependent on the specification of that ACP, but only on a set of high-level constraints listed below, which were decided upon early during the work on the ACP. Except when being specific about details of the ACP, this document uses the term "Generalized ACP" (GACP) and is applicable to any designs that meet the high-level constraints -- for example, the variations of the ACP protocol choices.

2. GACP Requirements

The high-level constraints of a GACP assumed and discussed in this document are as follows:

VRF isolation: The GACP is a virtual network (Virtual Routing and Forwarding (VRF)) across network devices; its routing and forwarding are separate from other routing and forwarding in the network devices. Non-GACP routing/forwarding is called the "data plane".

IPv6-only addressing: The GACP provides only IPv6 reachability. It uses Unique Local Addresses (ULAs) [RFC4193] that are routed in a location-independent fashion, for example, through a subnet prefix for each network device. Therefore, automatic addressing in the GACP is simple and stable: it does not require allocation by address registries, addresses are identifiers, they do not change when devices move, and no engineering of the address space to the network topology is necessary.

NOC connectivity: NOC equipment (controlling OAM operations) either has access to the GACP directly or has an IP subnet connection to a GACP edge device.

Closed Group Security: GACP devices have cryptographic credentials to mutually authenticate each other as members of a GACP. Traffic across the GACP is authenticated with these credentials and then encrypted.

GACP connect (interface): The only traffic permitted in and out of the GACP that is not authenticated by GACP cryptographic credentials is through explicit configuration for the traffic from/to the aforementioned non-GACP NOC equipment with subnet connections to a GACP edge device (as a transition method).

The GACP must be built to be autonomic and its function must not be able to be disrupted by operator or automated configuration/ provisioning actions (i.e., Network Management System (NMS) or Software-Defined Networking (SDN)). Those actions are allowed to impact only the data plane. This document does not cover those aspects; instead, it focuses on the impact of the above constraints: IPv6 only, dual connectivity, and security.

3. Solutions

3.1. Stable Connectivity for Centralized OAM

The ANI is the Autonomic Networking Infrastructure consisting of secure zero-touch bootstrap (BRSKI [BRSKI]), the GeneRic Autonomic Signaling Protocol (GRASP [GRASP]), and Autonomic Control Plane (ACP [ACP]). Refer to the reference model [REF_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 the GACP, for example, via the ACP with or without a complete ANI, starting with the option that we expect to be the most easy to deploy in the short term. It then describes limitations and challenges of that approach and the corresponding solutions and workarounds; it finishes with the preferred target option of autonomic NOC devices in Section 3.1.6.

This order was chosen because it helps to explain how simple initial use of a GACP can be and how difficult workarounds can become (and therefore what to avoid). Also, one very promising long-term solution 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. This document describes approaches to leverage a GACP for stable connectivity, from simple to complex, depending on the capabilities and limitations of the equipment used.

Three stages can be considered:

- There are simple options described in Sections 3.1.1 through 3.1.3 that we consider to be good starting points to operationalize the use of a GACP for stable connectivity today. These options require only network and OAM/NOC device configuration.

- The are workarounds to connect a GACP to non-IPv6-capable NOC devices through the use of IPv4/IPv6 NAT (Network Address Translation) as described in Section 3.1.4. These workarounds are not recommended; however, if non-IPv6-capable NOC devices need to be used longer term, then the workarounds are the only way to connect them to a GACP.

- Options for the near to long term 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 3.1.5 through 3.1.8.

3.1.1. Simple Connectivity for Non-GACP-Capable NMS Hosts

In the most simple candidate deployment case, the GACP extends all the way into the NOC via one or more GACP edge devices. See also Section 6.1 of [ACP]. These devices "leak" the (otherwise encrypted) GACP natively to NMS hosts. They act as the default routers to those NMS hosts and provide them with IPv6 connectivity into the GACP. NMS hosts with this setup need to support IPv6 (e.g., see [RFC6434]) but require no other modifications to leverage the GACP.

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

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

Because the GACP provides "only" for IPv6 connectivity, and because addressing provided by the GACP does not include any topological addressing structure that a NOC often relies on to recognize where devices are on the network, it is likely highly desirable to set up the Domain Name System (DNS; see [RFC1034]) so that the GACP IPv6 addresses of autonomic devices are known via domain names that include the desired structure. For example, if DNS in the network were set up with names for network devices as devicename.noc.example.com, and if the well-known structure of the data-plane IPv4 address space were used by operators to infer the region where a device is located, then the GACP 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 a GACP.

3.1.2. Challenges and Limitations 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) and limitations (that is, operators cannot achieve desired goals with this setup). The following list summarizes these challenges and limitations, and the following sections describe additional mechanisms to overcome them.

Note that these challenges and limitations exist because GACP is primarily designed to support distributed Autonomic Service Agent (ASA), a piece of autonomic software, in the most lightweight fashion. GACP is not required to support the additional mechanisms needed for centralized NOC systems. 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 3.1.3 for candidate solutions.

2. (Challenge) NMS hosts need to support IPv6, and this often is still not possible in enterprise networks. See Section 3.1.4 for some workarounds.

3. (Limitation) Performance of the GACP may be limited versus normal "data-plane" connectivity. The setup of the GACP will often support only forwarding that is not hardware accelerated. Running a large amount of traffic through the GACP, especially for tasks where it is not necessary, will reduce its performance and effectiveness for those operations where it is necessary or highly desirable. See Section 3.1.5 for candidate solutions.

4. (Limitation) Security of the GACP is reduced by exposing the GACP natively (and unencrypted) in a subnet in the NOC where the NOC devices are attached to it. See Section 3.1.7 for candidate solutions.

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

3.1.3. Simultaneous GACP and Data-Plane Connectivity

Simultaneous connectivity to both the GACP 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 GACP; IPv4 provides access via the data plane. If, as explained above in the simple case, an autonomic device supports native attachment to the GACP, and the existing NOC setup is IPv4 only, then it could be sufficient to attach the GACP 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 [IEEE.802.1Q]) or physical interface connecting to a data-plane subnet, and another connecting into a GACP connect subnet. See Section 8.1 of [ACP] for more details. That document also specifies how a NOC device can receive autoconfigured addressing and routes towards the ACP connect subnet if it supports default address selection as specified in [RFC6724] and default router preferences as specified in [RFC4191].

Configuring a second interface on a NOC host may be impossible or seen as undesired complexity. In that case, the GACP edge device needs to provide support for a "combined ACP and data-plane interface" as described in Section 8.1 of [ACP]. 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 Rule 5.5 of [RFC6724] regarding source address selection, including caching of next-hop information.

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

3.1.4. IPv4-Only NMS Hosts

One architectural expectation for the GACP as described in Section 1.3 is that all devices that want to use the GACP (including NMS hosts) support IPv6. Note that this expectation does not imply any requirements for the data plane, especially it does not imply that IPv6 must be supported in it. The data plane could be IPv4 only, IPv6 only, dual stack, or it may not need to have any IP host stack on the network devices.

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 very far from being ubiquitously supported in NOC backend solutions (hardware or software) or 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 GACP, 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 a GACP.

To connect IPv4-only management-plane devices/applications with a GACP, some form of IP/ICMP translation of packets between IPv4 and IPv6 is necessary. The basic mechanisms for this are in [RFC7915], which describes the Stateless IP/ICMP Translation Algorithm (SIIT). There are multiple solutions using this mechanism. To understand the possible solutions, we consider the requirements:

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

2. GACP 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 GACP. When a connection from a GACP device is made to an NMS host, the IPv4 source address of the connection (as seen by the NMS host) must be unique per GACP device and must be the same address as in (1) to maintain addressing simplicity similar to 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 GACP addresses into IPv4 space and 1:1 mapping of IPv4 NMS host space into IPv6 (for use in the GACP). This means that SIIT-based solutions are sufficient and preferred.

Note that GACP devices may use multiple IPv6 addresses in the GACP. For example, Section 6.10 of [ACP] defines multiple useful addressing sub-schemes supporting this option. All those addresses may then need to be reachable through IPv6/IPv4 address translation.

The need to allocate for every GACP 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 private IPv4 address space, it is important that the GACP IPv6 addresses can be mapped efficiently into IPv4 address space without too much waste.

Currently, the most flexible mapping scheme to achieve this is [RFC7757] because it allows configured IPv4 <-> IPv6 prefix mapping. Assume the GACP uses the ACP Zone Addressing Sub-Scheme and there are 3 registrars. In the ACP Zone Addressing Sub-Scheme, for each registrar, there is a constant /112 prefix for which an Explicit Address Mapping (EAM), as defined in RFC 7757, to a /16 prefix can be configured (e.g., in the private IPv4 address space described in [RFC1918]). Within the registrar’s /112 prefix, Device-Numbers for devices are sequentially assigned: with the V bit (Virtualization bit) effectively two numbers are assigned per GACP 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 prefix longer than a /112 can be configured into the EAM in order to use less IPv4 space.

When using the ACP Vlong Addressing Sub-Scheme, it is unlikely that one wants or needs to translate the full /8 or /16 of addressing space per GACP 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. However, this does imply that only 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 address prefix assigned to a GACP connect interface.

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

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

Note that prior solutions such as NAT64 ([RFC6146]) may equally be useable to translate between GACP IPv6 address space and NMS hosts' IPv4 address space. As a workaround, this can also be done on non-GACP Edge Devices connected to a GACP 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) GACP device address to an IPv4 address. Such an approach should use automation/scripting where these address translation entries are created dynamically whenever a GACP device is enrolled or first connected to the GACP network.

The NAT methods described here are not specific to a GACP. Instead, they are similar to what would be necessary when some parts of a network only support IPv6, but the NOC equipment does not support IPv6. Whether it is more appropriate to wait until the NOC equipment supports IPv6 or to use NAT beforehand depends in large part on how long the former will take and how easy the latter will be when using products that support the NAT options described to operationalize the above recommendations.

3.1.5. Path Selection Policies

As mentioned above, a GACP is not expected to have high performance because its primary goal is connectivity and security. 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 GACP to achieve the desired security.

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

To solve the expected performance limitations of the GACP, we do expect to have the above-described dual connectivity via both GACP and data plane between NOC application devices and devices with GACP. The GACP 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 and will not be present during, e.g., early stages of device bootstrap, failures, provisioning mistakes, or network configuration changes.

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

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

Ideally, a NOC system would learn and keep track of all addresses of a device (GACP 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, GACP only, default to data plane, fallback to GACP, etc.). 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 GACP uses, whether it uses the DNS or names local to the NOC system. Below, 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 GACP reachable address of the device for troubleshooting and probing/discovery that is desired to always only use the GACP.

- One name (name-both.noc.example.com) with data-plane and GACP addresses.

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

Using different names that map to different addresses (or subsets of addresses) can be difficult to set up and maintain, especially because data-plane addresses may change due to reconfiguration or relocation of devices. The name-based approach alone cannot strongly support policies for existing applications and long-lived flows to automatically switch between the ACP and data plane in the face of data-plane failure and recovery. A solution would be host transport stacks on GACP nodes that support the following requirements:

1. Only the GACP addresses of the responder must be required by the initiator for the initial setup of a connection/flow across the GACP.

2. Responder and Initiator must be able to exchange their data-plane addresses through the GACP, and then -- if needed by policy -- build an additional flow across the data plane.

3. For unmodified application, the following policies should be configurable on at least a per-application basis for its TCP connections with GACP peers:

   Fallback (to GACP): An additional data-plane flow is built and used exclusively to send data whenever the data plane is operational. When the additional flow cannot be built during connection setup or when it fails later, traffic is sent across the GACP flow. This could be a default policy for most OAM applications using the GACP.

Suspend/Fail: Like the Fallback policy, except that traffic will not use the GACP flow; instead, it will be suspended until a data-plane flow is operational or until a policy-configurable timeout indicates a connection failure to the application. This policy would be appropriate for large-volume background or scavenger-class OAM applications such as firmware downloads or telemetry/diagnostic uploads -- applications that would otherwise easily overrun performance-limited GACP implementations.

GACP (only): No additional data-plane flow is built, traffic is only sent via the GACP flow. This can just be a TCP connection. This policy would be most appropriate for OAM operations known to change the data plane in a way that could impact connectivity through it (at least temporarily).

4. In the presence of responders or initiators not supporting these host stack functions, the Fallback and GACP policies must result in a TCP connection across the GACP. For Suspend/Fail, presence of TCP-only peers should result in failure during connection setup.

5. In case of Fallback and Suspend/Fail, a failed data-plane connection should automatically be rebuilt when the data plane recovers, including when the data-plane address of one side or both sides may have changed -- for example, because of reconfiguration or device repositioning.

6. Additional data-plane flows created by these host transport stack functions must be end-to-end authenticated by these host transport stack functions with the GACP domain credentials and encrypted. This maintains the expectation that connections from GACP addresses to GACP addresses are authenticated and encrypted. This may be skipped if the application already provides for end-to-end encryption.

7. For enhanced applications, the host stack may support application control to select the policy on a per-connection basis, or even more explicit control for building of the flows and which flow should pass traffic.

Protocols like Multipath TCP (MPTCP; see [RFC6824]) and the Stream Control Transmission Protocol (SCTP; see [RFC4960]) can already support part of these requirements. MPTCP, for example, supports signaling of addresses in a TCP backward-compatible fashion, establishing additional flows (called subflows in MPTCP), and having primary and fallback subflows via MP_PRIO signaling. The details of how MPTCP, SCTP, and/or other approaches (potentially with extensions

and/or (shim) layers on top of them) can best provide a complete solution for the above requirements need further work and are outside the scope of this document.

3.1.6. Autonomic NOC Device/Applications

Setting up connectivity between the NOC and autonomic devices when the NOC device itself is non-autonomic is a security issue, as mentioned at the beginning of this document. It also results in a range of connectivity considerations (discussed in Section 3.1.5), some of which may be quite undesirable or complex to operationalize.

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

According to [ACP], 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 a (physical) NMS host system by combining a virtual GACP-enabled router with non-GACP-enabled Virtual Machines (VMs) for NOC applications via a hypervisor. This would leverage the configuration options described in the previous sections but just virtualize them.

3.1.7. Encryption of Data-Plane Connections

When combining GACP and data-plane connectivity for availability and performance reasons, this too has an impact on security: When using the GACP, most traffic will be encryption protected, especially when considering the above-described use of application devices with GACP. 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 GACP-capable NMS hosts, because in that case the communicating GACP-capable NMS host and the GACP network device have credentials they can mutually

trust (same GACP domain). As a result, data-plane connectivity that does support this can simply leverage TLS [RFC5246] or DTLS [RFC6347] with those GACP credentials for mutual authentication -- and this does not incur new key management.

If this automatic security benefit is seen as most important, but a "full" GACP stack into the NMS host is unfeasible, then it would still be possible to design a stripped-down version of GACP functionality for such NOC hosts that only provides enrollment of the NOC host with the GACP cryptographic credentials and does not directly participate in the GACP encryption method. Instead, the host would just leverage TLS/DTLS using its GACP credentials via the data plane with GACP network devices as well as indirectly via the GACP connect interface with the above-mentioned GACP connect interface into the GACP.

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

3.1.8. 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 a GACP is undesirable in the long term. 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 GACP as a lightweight application for NMS hosts so GACP extends all the way into the actual NMS hosts.

3. Leverage and (as necessary) enhance host transport stacks with automatic GACP with multipath connectivity and data plane as outlined in Section 3.1.5.

4. Consider how to best map NMS host desires to underlying transport mechanisms: The three points above do not cover all options. Depending on the OAM, one may still want only GACP, want only data plane, automatically prefer one over the other, and/or want to use the GACP with low performance or high performance (for emergency OAM such as countering DDoS). As of today, it is not clear what the simplest set of tools is to explicitly enable the

choice of desired behavior of each OAM. The use of the above-mentioned DNS and multipath mechanisms is a start, but this will require additional work. This is likely a specific case of the more generic scope of TAPS.

3.2. Stable Connectivity for Distributed Network/OAM

Today, many distributed protocols implement their own unique security mechanisms.

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

Furthermore, a GRASP instance ([GRASP]) can run on top of a GACP as a security and transport substrate and provide common local and remote neighbor discovery and peer negotiation mechanisms; this would allow unifying and reusing future protocol designs.

4. Architectural Considerations

4.1. No IPv4 for GACP

The GACP is intended 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 GACP when the OAM agents on a GACP network device do not support IPv6. Therefore, the question was raised whether the GACP should optionally also support IPv4.

The decision not to include IPv4 for GACP in the use cases in this document was made for the following reasons:

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

In other types of networks as well, we think that efforts to support autonomic networking are better spent in ensuring that one address family will be supported so all use cases will work with it in the long term, instead of duplicating effort with IPv4. Also, auto-addressing for the GACP with IPv4 would be more complex than in IPv6 due to the IPv4 addressing space.

5. Security Considerations

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

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

GACP addresses are ULAs. Using these addresses also for NOC devices, as proposed in this document, is not only necessary for the simple routing functionality explained above, but it is also more secure than global IPv6 addresses. ULAs are not routed in the global Internet and will therefore be subject to more filtering even in places where specific ULAs are being used. Packets are therefore less likely to leak and less likely to be successfully injected into the isolated GACP 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 GACPs will never connect all together, and that only a few GACPs may ever need to connect together, e.g., when mergers and acquisitions occur.

Note that the GACP constraints demand that only packets from connected subnet prefixes are permitted from GACP connect interfaces, limiting the scope of non-cryptographically secured transport to a subnet within a NOC that instead has to rely on physical security (i.e., only connect trusted NOC devices to it).

To help diagnose packets that unexpectedly leaked, for example, from another GACP (that was meant to be deployed separately), it can be useful to voluntarily list your own ULA GACP prefixes on some sites on the Internet and hope that other users of GACPs do the same so that you can look up unknown ULA prefix packets seen in your network. Note that this does not constitute registration. <https://www.sixxs.net/tools/grh/ula/> was a site to list ULA

prefixes, but it has not been open for new listings since mid-2017. The authors are not aware of other active Internet sites to list ULA use.

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

According to Section 4.4 of [RFC4193], PTR records for ULA addresses should not be installed into the global DNS (no guaranteed ownership). Hence, there is also the need to rely on voluntary lists (as mentioned above) to make the use of an ULA prefix globally known.

Nevertheless, some legacy OAM applications running across the GACP may rely on reverse DNS lookup for authentication of requests (e.g., TFTP for download of network firmware, configuration, or software). Therefore, operators may need to use a private DNS setup for the GACP ULAs. This is the same setup that would be necessary for using RFC 1918 addresses in DNS. For example, see the last paragraph of Section 5 of [RFC1918]. In Section 4 of [RFC6950], these setups are discussed in more detail.

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 cryptographic credential mechanisms of the GACP.

If DNS and especially reverse DNS are set up, then they should be set up in an automated fashion when the GACP address for devices are assigned. In the case of the ACP, DNS resource record creation can be 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 ULAs 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 publicly for "security" by concealment reasons, then GACP DNS entries are probably one of the least problematic use cases for split DNS: The GACP DNS names are only needed for the NMS hosts intending to use the GACP -- but not network wide across the enterprise.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. Normative References

[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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc4193>.

[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, <https://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, <https://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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc7915>.

7.2. Informative References

[ACP] Eckert, T., Behringer, M., and S. Bjarnason, "An Autonomic Control Plane (ACP)", Work in Progress, draft-ietf-anima-autonomic-control-plane-13, December 2017.

[BRSKI] Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", Work in Progress, draft-ietf-anima-bootstrapping-keyinfra-15, April 2018.

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

[IEEE.802.1Q] IEEE, "IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks", IEEE 802.1Q-2014, DOI 10.1109/ieeestd.2014.6991462, December 2014, <http://ieeexplore.ieee.org/servlet/opac?punumber=6991460>.

[ITUT_G7712] ITU, "Architecture and specification of data communication network", ITU-T Recommendation G.7712/Y.1703, November 2001, <https://www.itu.int/rec/T-REC-G.7712/en>.

[REF_MODEL] Behringer, M., Carpenter, B., Eckert, T., Ciavaglia, L., and J. Nobre, "A Reference Model for Autonomic Networking", Work in Progress, draft-ietf-anima-reference-model-06, February 2018.

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

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <https://www.rfc-editor.org/info/rfc4960>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, DOI 10.17487/RFC6434, December 2011, <https://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, <https://www.rfc-editor.org/info/rfc6518>.

[RFC6950] Peterson, J., Kolkman, O., Tschofenig, H., and B. Aboba, "Architectural Considerations on Application Features in the DNS", RFC 6950, DOI 10.17487/RFC6950, October 2013, <https://www.rfc-editor.org/info/rfc6950>.

Acknowledgements

This work originated from an Autonomic Networking project at Cisco Systems, which started in early 2010, with 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, and Ravi Kumar Vadapalli. The authors 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 reviews.

Authors' Addresses

Toerless Eckert (editor)
Huawei USA
2330 Central Expy
Santa Clara 95050
United States of America

Email: tte+ietf@cs.fau.de, toerless.eckert@huawei.com

Michael H. Behringer

Email: michael.h.behringer@gmail.com