Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 2, 2017

R. Kumar
A. Lohiya
Juniper Networks
M. Blanchet
Viagenie
January 29, 2017

Centralized Address Space Management(CASM) Problems and Use cases
draft-kumar-casm-problem-and-use-cases-00

Abstract

The organisations use IP Address Space Management (IPAM) tools to manage their IP address space, often with proprietary database and interfaces. This document describes evolution of IPAM into a standardized interfaces for centralized management of IP addresses.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 2, 2017.

Table of Contents

1. Introduction

The address space management is an intergral part of any network management solution. The network may be based on legacy design or a more modern private and public cloud, the network may be big or small but every network operator need to manage the addressing needs of network elements. Typically, network operators write proprietary scripts or use cheat sheets to manage the addressing requirements. In recent trends, open source communities have developed tools to manage available IP addreess space.

The open source or proprietray tools and scripts are collectively known as Internet Protocol Address Management (IPAM) system. The organizations use IPAM system to manage their IP address space, often with proprietary database and interfaces. One of the biggest challenges with IPAM systems, is lack of standardized interface for allocation, storing and retrieving information.

This document describes a diverse set of use cases for a IPAM system and the probelms identfied with current IPAM approach. The problems identifed here should become the basis for a new vision defined as Centralized Address Space Management (CSAM).

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Terminology

CASM: Centralized Address Space Management

IPAM: IP Address Management

4. Address Space Management Use cases

The address space management is an intergral part of any network management solution. Every device in the network be it a physical or virtual, needs an IP address for communication with other devices in the network. There is an absolute requirement that a network operator must find a way to assign address to these devices.

The address management could be as simple as having one address pool from where addresses are allocated or may a much more complex scheme based on various requirements and nature of the network. This section is going to identfiy few top uses cases of address management.

4.1. DHCP server pool

One of the most common method to assign an IP address to a device or function is DHCP. A device may request one or more IP addresses. The DHCP server on network handles all the DHCP requests and assign IP addresses. These addresses are allocated from a pre-defined address pool.

A DHCP server might need multiple address pools if it manages DHCP request on multiple network segments. An address management system may be used to initialize these address pools on DHCP servers or could also be configured statically. But the static assigment is prone to misconfiguration and if the DHCP server is ever replaced, the new server must be configured with the same old pool.

4.2. Static address configuration

Some devices or functions do not rely on DHCP protocols to obtain an IP address. This could be due to lack of DHCP client functionality or lack of DHCP server avaialable in the network segement for whatever reason. In such situations, an IP address may be configured statically but static IP address assignment is prone to errors as

mentioned earlier. The better way is to use an address management system for configuring devices without DHCP support.

4.3. Public IP address pool

The public IPv4 addresses are very precious resources and should be used very carefully. A given organization may have a small number of these addresses, so it must find a way to allocate and free these resources effectively. The manual configuration mechanism may not be the best way to manage this resource.

4.4. Multicast IP address pool

The multicast addresses are used for distributing broadcast contents. The multicast content distributor must be assigned an address and the content consumer must somehow figure out that address. This is usually configured manually or throgh proprietary mechanisms.

4.5. SDN controllers

In order to build private or public clouds, address management of virtual machines, virtual functions and overlay networks is a very important task. In addition, the network operator also need to manage addressing of undelay network elements. The SDN controllers and underlay management systems must coordinate addressing schemes to ensure smooth operation. There is need for one address management system that would meet the requirements of such a network deployment.

In order to create overlay networks and virtual workloads, the SDN controller also manage MAC addresses to assign to virtual network interfaces. But this is typically not handled by IPAM systems.

5. Legacy address space management (IPAM) systems

As mentioned earlier, address management is a central component of every network management system. Organizations small or large deploy different ways of achieving this; some write their own scripts or use cheat sheets, and others use open source tools.

These systems may not be suitable for all kind of uses cases due to lack of functionality and moreover the interfaces to these systems are closed which makes migration from one system to other difficult.

Although, the functionality of IPAM systems vary from vendor to vendor but in general as a whole, following drawbacks exists:

- Lack of common set of standard interfaces across IPAM system vendors

- Address usually allocated with very little or no context
- Lacks ability to annotate requests with user-defined attributes as private or public address
- Lacks capability to manage both unicast and multicast addresses
- MAC address and network segment (VLAN) does not given enough information about user or usages
- Lack of built-in multi-tetancy into interfaces
- Lack of information about address requester such as virtual or physical device
- Lack of integration with name services such as DNS
- Lack of integration with DHCP server to get address records
- Lack of integration with address translation services such as NAT44 and NAT64

The purpose is not to show a laudry list of deficiencies in the available IPAM system but to show a need to develop a new system that can meet the address allocation requirements of modern network architectures that gives consumers a portable way to use these systems.

6. Acknowledgements

This document started from a slide deck authored by Rakesh Kumar and Anil Lohiya.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

TBD

9. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

Authors' Addresses

Rakesh Kumar
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: rkkumar@juniper.net

Anil Lohiya
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: alohiya@juniper.net

Marc Blanchet
Viagenie
246 Aberdeen
Quebec, QC G1R 2E1
Canada

Email: marc.blanchet@viagenie.ca
URI: http://viagenie.ca

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 3, 2017

R. Kumar
A. Lohiya
Juniper Networks
M. Blanchet
Viagenie
January 30, 2017

Centralized Address Space Management(CASM) Requirements and Framework
draft-kumar-casm-requirements-and-framework-00

Abstract

The organizations use IP Address Space Management (IPAM) tools to manage their IP address space, often with proprietary database and interfaces. This document describes evolution of IPAM into a standardized interfaces for centralized management of IP addresses.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 3, 2017.

Table of Contents

1. Introduction

The address space management is an intergral part of any network management solution. The network architectures are rapidally changing with the migration toward private and public clouds. At the same time, application architectures are also evolving with a shift toward micro-services and multi-tiered approach.

There is a pressing need to define a new address management system which can meet these diverse set of requirements. Such a system must be built with well defined interfaces so users can easily migrate from one vendor to another without rewriting their network management systems.

This document identifies a broad set of requirements and defins a architectural framework that should become the basis to develop a new address management system. We are calling this new system as Centralized Address Space Management (CSAM) system.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Terminology

CASM: Centralized Address Space Management

IPAM: IP Address Management

4. Requirements from CASM system

In order to build CASM, there is a clear need to define a broad set of requirements that must be the basis for defining the architecture framework for CASM. The requirements should be able to meet the various use-cases identified in the draft.

This sections identifies the major set of requirements for defining CASM system.

4.1. General operational requirements

Some requirements are not specific to any particular functiolaity of CASM but applicable to all aspects of CASM system.

Multi-tenancy: All interfaces exposed by CASM system must be multi-tenant capable. This is highly desirable for cloud based network management solutions. It may also be applicable for a service provider with different managed services use-case scenario.

Autentication and Authorization: All interfaces exposed by CASM system must support an authentication scheme. It also highly desriable to support operational restrictions on certain resources based on identity for security reasons.

Audit Logging: All CASM actvities must be logged for auditing or debugging purposes. The system must provide an interface to access these records.

Error notification: All interfaces exposed by CASM system must support error handling and user-defined error notification mechanism suh as alert or email. There may also be need to take corrective action for autonomus operation.

4.2. Interafce modeling requirements

The interface to external user must be meta-dat driven as much as possible to meet wider set of use-cases, e.g., instead of requesting an explicit IPv4 address, user should specify an adsress request based on its requirements.

The following requirements should be considered for pool management interface defintion. The attributes should be realted to the requestor which could a physical device, virtual machine, container or other entities present in a network.

Fucntional attributes such as switch, router, firewall, server, end-point

Form-factoral attributes such as physical, virtual

Opertional attributes such as management plane, control plane, data plane

Network segment identifier, such as VLAN, VxLAN or other user-defined value

Network segment type such as point-to-point, multi-point, loopback

Addressing scope attributes such as private, public, vpn, unicast, multicast

Extensible user-defined attributes

4.3. Functional requirements

The CASM should support following functionality for it to be adopted for wide varierty of use cases.

4.3.1. Address pools

A CASM system should allow ability to manage different kind of address pools. The following pools should be considered for implementation; this is not mandatory or exhaustive by any means but given here as most commonly used in networks. The CASM system should allow user-defined pools with any address objects.

Unicast address pool:

- Private IPv4 addresses
- Public IPv4 addresses
- IPv6 addresses
- MAC Addresses

Mulicast address pool:

IPv4 address

IPv6 address

4.3.2. Pool management

There should be a rich set of functionality as defined in this section for operation of a given pool.

Address management:

Address allocation either as single or block

Address reservation

Allocation logic such as mapping schemes or algorithm per pool

General management:

Pool initializing, resizing, threshold markings for resource monitoring

Pool attributes such as used to automatcally create DNS record

Pool priority for searching across different pools

Pool fragmenetation rules, such as how pool can be sub-divided

Pool lease rules for alloation requests

4.3.3. Integration with other address services

In order to build a complete address management system, it is important that CASM should be able to integrate with other address services. This will provide a complete solution to network operators without requiring any manual or proprietary workflows.

DHCP server:

Interface to initialize address pools on DHCP server

Notification interface whenever an address lease is modified

Interface to access address lease records from DHCP server

Ability to store lease records and play back to DHCP server on reboot

DNS server:

Interface to create DNS records on DNS server based on DHCP server events

NAT device:

Interface to initialize NAT pools

Interface to access NAT records from NAT device

Ability to store NAT records and play back to NAT device on reboot

5. Archiectural framework

Based on the requirements specifed in this document, we propose the following high-level architecture for building CASM.

There are three broad categories for CASM interface defintion:

Pool management interface: Interface to external user or applications such as SDN controller to manage addresses

Log interface: Interface to access log and records such as DHCP, DNS, NAT

Integration interface: Interface to address services such as DHCP, DNS, NAT

The propsed CASM framework is shown in Figure 1.

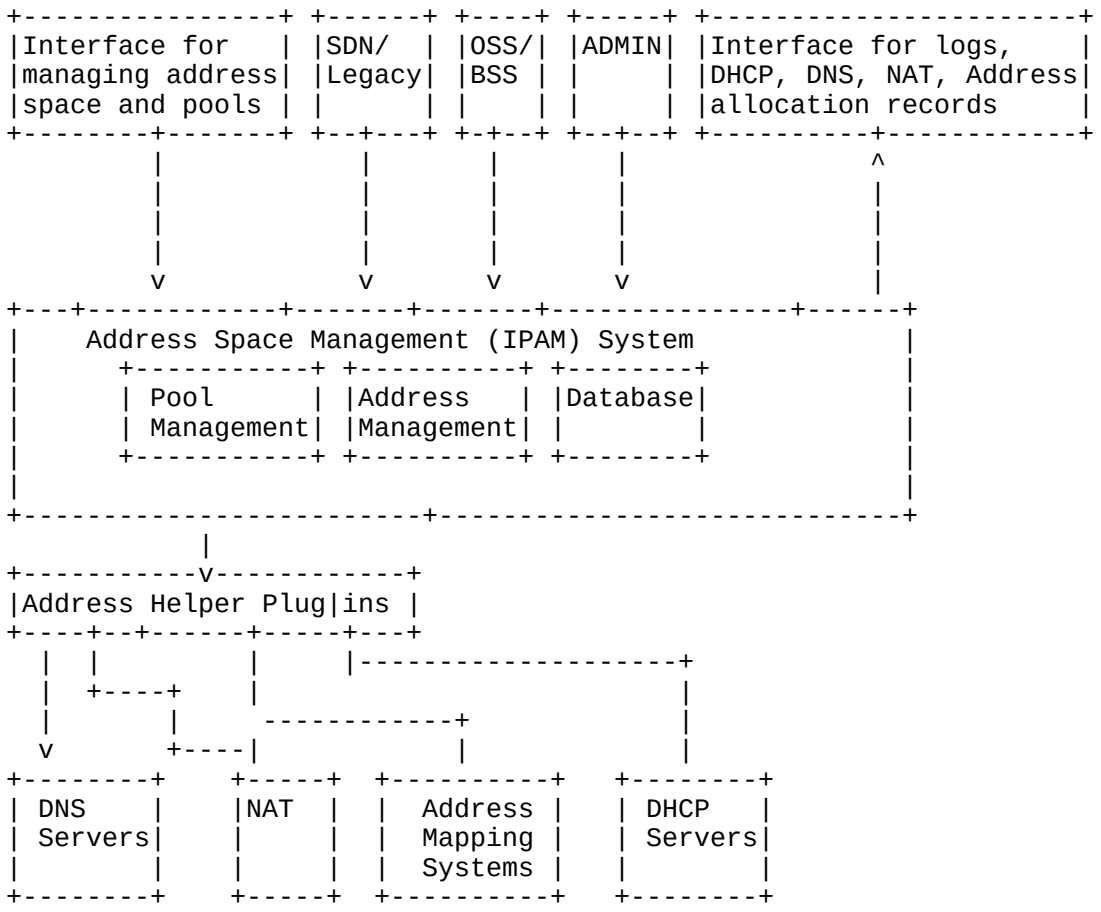

Figure 1: CASM Architecture

6. Acknowledgements

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

TBD

9. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

Authors' Addresses

Rakesh Kumar
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: rkkumar@juniper.net

Anil Lohiya
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: alohiya@juniper.net

Marc Blanchet
Viagenie
246 Aberdeen
Quebec, QC G1R 2E1
Canada

Email: marc.blanchet@viagenie.ca
URI: http://viagenie.ca

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 7, 2017

C. Li
C. Xie
China Telecom
J. Bi
Tsinghua University
W. Xu
Huawei Technologies
March 6, 2017

Interface to the Address Pool Management
draft-li-casm-address-pool-management-arch-00

Abstract

This document describes an mechanism for a standard, programmatic interface for address pool management. With the remaining IPv4 address becoming more and more scattered, it is complicated to manually configure the address pools on lots of Broadband Network Gateways(BNGs) for operators. By introducing SDN/NFV in BNG, the address pools can be allocated in a centralized way. It will not only simplify the address management for operators, but also improve the utilization efficiency of the address pool.

Table of Contents

1. Introduction

The Broadband Network Gateway(BNG), which manages a routable IP address on behalf of each subscriber, should be configured with the IP address pools allocated to subscribers. However, currently operators are facing with the address shortage problem, the remaining IPv4 address pools are usually quite scattered, no more than /24 per address pool in many cases. Therefore, it is complicated to manually configure the address pools on lots of Broadband Network Gateway(BNG) for operators. For large scale MAN, the number of BNGs can be up to over one hundred. Manual configuration on all the BNGs statically will not only greatly increase the workload, but also decrease the utilization efficiency of the address pools when the number of subscribers changes in the future.

Another use case which needs to configure the address pools is IPv6 migration. For IPv6 transition mechanisms, e.g. DS-Lite, lw4over6, etc., they all need to be configured with address pools as translated routeable addresses. When high availability features, e.g. active-active/active-standby failover mechanism, etc., are enabled for these IPv6 transition mechanisms, different address pools need to be

configured on each transition instance. This will further increase the number of address pools need to be configured. Besides, the occupation of the address pools may vary during different transition periods, (e.g. at the early stage of IPv6 transition, IPv4 traffic will normally occupy a great portion of the total traffic, while in the later stage of IPv6 transition, IPv4 traffic will decrease and the amount of IPv4 address pools will decrease accordingly.

There are other devices which may need to configure address pools as well. For example, the Firewall need to configure the address pool for acl/NAT process. The VPN also needs to configure the address pools for end-users.

When SDN/NFV is introduced in the network, these devices (e.g. BNG, CGN, firewall, VPN, etc.) will run as VNFs in virtualized environment. A common centralized address management server can interact with different VNFs and allocate address pools automatically.

In this document, we propose a mechanism to manage the address pools centrally. In this way, operators do not need to configure the address pools one by one manually and it also helps to use the address pools more efficiently.

2. Terminology

The following terms are used in this document:

APMS A management system which has a centralized databse manage the overall address pools and allocate address pools to the device in the devices.

DA A device agent in device, which contact with APM server to manipulate address pool.

3. Architectural Overview

In this architecture, the Address Pool Management (APM) server is a centralized address pool management server for operators to configure the overall address pools. It maintains the address pool database including the overall address pools (OAP) and the address pool status(APS). Operators can configure its remaining address pools in the OAP. They can also reserve some address pool for special-purpose usage. The address pools status is to reflects the current usage of the address pools for different devices. APM also has the interface to configure the address pools to different devices dynamically.

In each device, there is an device agent (DA) to contact with APM server. It initiates the address pools allocation requests, passes the address pools to local instances, report the status of local address pool usage and update the address pools requests, etc. For some devices, e.g. v6transition, VPN, etc., additional routing modules needs to update the routing table accordingly.

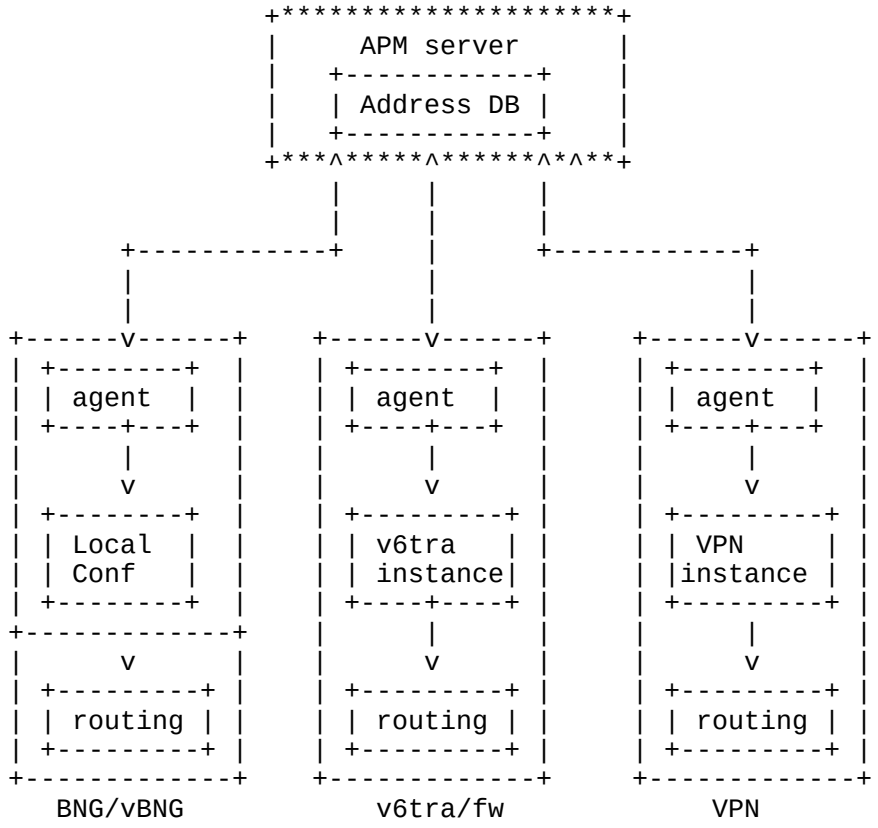

Figure 1: Interface to Address Pool Management (APM)

The overall procedure is as follows:

o Operators will configure remaining address pools centrally in the Address Pool Management System (APMS). There are multiple address pools which can be configured centrally. The APMS server will then divide the address pools into addressing unit (AU) which will be allocated to the agent in devices by default.

- The agent will initiate Address Pool request to the APMS. It can carry its desired size of address pool the request, or just use a default value. The address pool size in the request is only used as a hint. The actual size of the address pool is totally determined by APMS. It will also carry the DA's identification and the type of address pool.

- APMS looks up the remaining address pool in its local database. It will then allocate a set of address pools to the DA. Each address pool has a related lifetime.

- DA receives the AddressPool reply and use them for their purpose.

- If the lifetime of the address pool is going to expire, the DA should issue an AddressPoolRenew request to extend the lifetime,including the IPv4, IPv6, Ports, etc.

- The AddressPoolReport module keeps monitoring and reports the current usage of all current address pools for each transition mechanism. if it is running out of address pools, it can renew the AddressPoolRequest for a newly allocated one. It can also release and recycle an existing address pool if the that address pool has not been used for a specific and configurable time.

- When the connection of APMS is lost or the APMS needs the status information of certain applications, the APMS may pre-actively query the DA for the status information.

4. Initial Address Pool Configuration

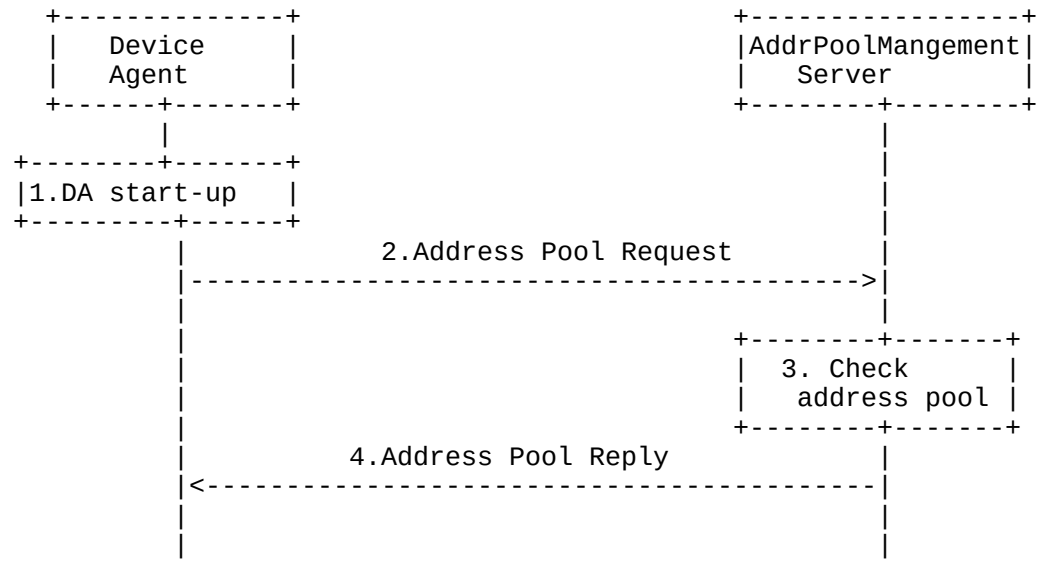

Figure 2: Initial Address Pool Configuration

Figure 2 illustrates the initial address pool configuration procedure:

1. The DA checks whether there is already address pool configured in the local site when it starts up. if no, it means the initial start-up or the address pool has been released. if yes, the address pool could be used directly.

2. The DA will initiate Address Pool request to the APMS. It can carry its desired size of address pool in the request, or just use a default value. The address pool size in the DA's request is only used as a hint. The actual size of the address pool is totally determined by APMS. It will also carry the DA's identification, the type of transition mechanism and the indication of port allocation support.

3. The APMS determines the address pool allocated for the DA based on the parameters received.

4. The APMS sends the Address Pool Reply to the DA. It will also distribute the routing entry of the address pool automatically. In particular, if the newly received address pool can be aggregated to an existing one, the routing should be aggregated accordingly.

5. Address Pool Status Report

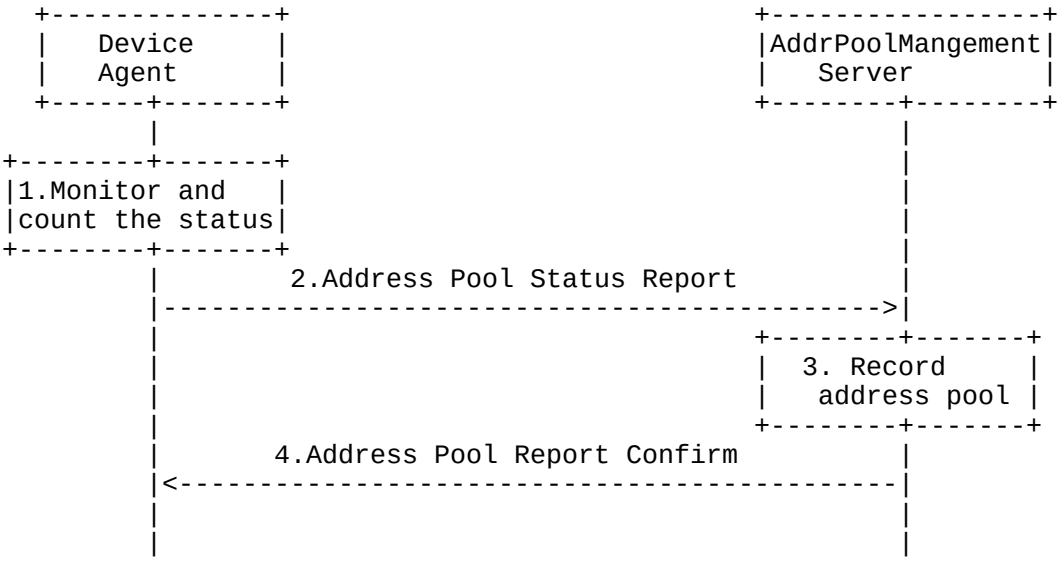

Figure 3: Address Pool Status Report

Figure 3 illustrates the active address pool status report procedure:

1. The DA will monitor and count the usage status of the local address pool. The DA counts the address usage status in one month, one week and one day, which includes the local address, address usage ratio (peak and average values), and the port usage ratio (peak and average values).

2. The DA reports the address pool usage status to the APMS. for example, it will report the address usage status in one day, which contains the IP address, NAT44, address list: 30.14.44.0/28, peak address value 14, average address usage ratio 90%, TCP port usage ratio 20%, UDP port usage ratio 30% and etc.

3. The APMS records the status and compares with the existing address information to determine whether additional address pool is needed.

4. The APMS will confirm the address pool status report request to the DA. It will keep sending the address pool status report request to the APMS if no confirm message is received.

6. Address Pool Status Query

When the status of APMS is lost or the AMS needs the status information of the DAs, the APMS may actively query the TD for the status information, as shown in step 1 of Figure 4. The following steps 2,3,4,5 are the same as the Address Pool Status Report procedure.

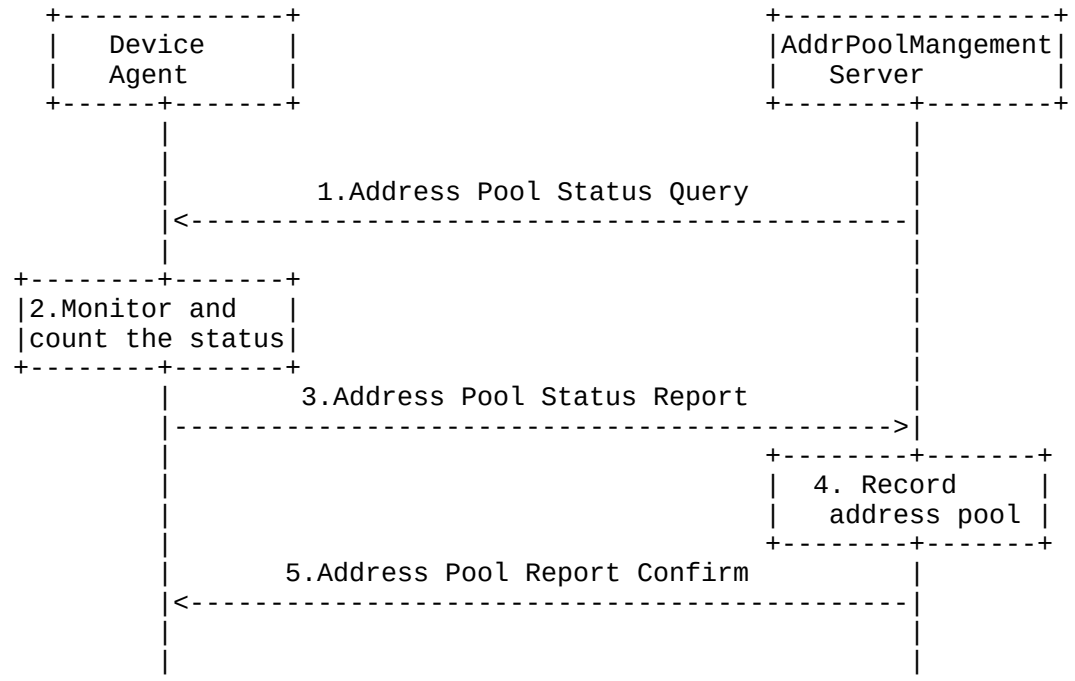

Figure 4: Address Pool Status Query

7. Address Exhaustion

When the DA uses up the addresses allocated, it will renew the address pool request to the APMS for an additional address pool. The procedure is the same as the initial address pool request.

8. Address Pool Release

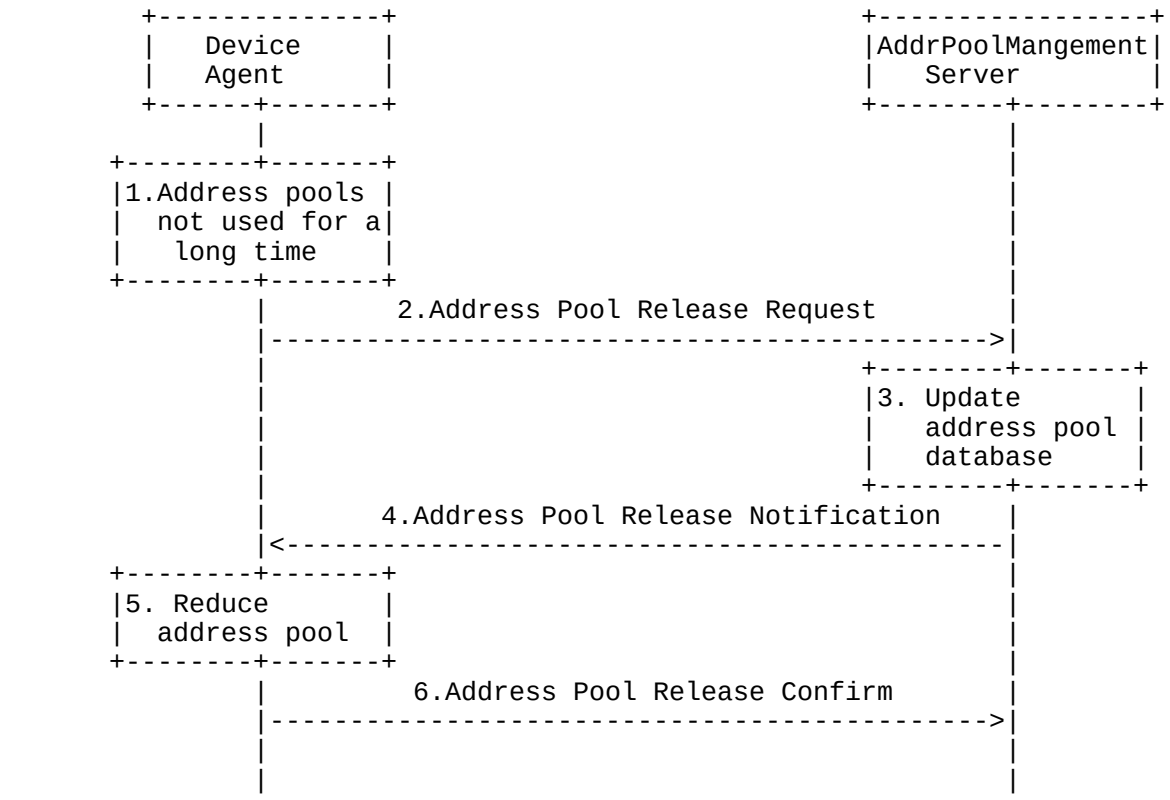

Figure 5: Address Pool Release

Figure 5 illustrates the address pool release procedure:

1. The counting module in the DA checks that there are addresses not used for a long time;

2. The DA sends the address pool release request to the APMS to ask the release of those addresses;

3. The APMS updates the local address pool information to add the new addressed released.

4. The APMS notifies the TD that the addresses have been release successfully;

5. The DA will update the local address pool. if no Address Pool Release Notification is received, the DA will repeat step 2;

6. The DA confirms with the APMS that the addres pool has been released successfully.

9. Compatibility of different forms of devices

As described in section 3, each device has its address pools, the Address Pool Management (APM) server act as a centralized address pool management server for operators to configure the overall address pools of each devices. In this form of device, the user plane and the control plane are integrated in the box. There are another form of device, the control plane is separated from the box and one or more devices share centralized control plane. In this device form, the control plane will manage multiple user plane devices. A number of devices that are subordinate to a control plane will jointly share the address pools. The control plane device, together with the dependent multiple user plane devices, forms a "big" device. This bigger device contacts with the APM server to manipulate IP address pool. For example, the device acts as a server side when running the NETCONF protocol between the device and the APM. It determines whether the usage status of the IP address pool resource in device is satisfies the condition. For example, the address pool resource of device is not enough or excessive. It sends address pools resource request to the APM server, and receives response with address pools resource for this device allocated from APM server. Then it passes the address pools resource to local instances. In addition, it report the status of local address pool resource usage and update the address pools requests, etc.

10. Control Protocol consideration

The I2APM architecture consists of two major distinct entities: APM Server and network equipment with an APM Agent. In order to provide address pool manipulations between these two entities, the I2APM architecture calls for well-defined protocols for interfacing between them. For compatibility with legacy network equipment, the architecture reuse legacy protocol such as radius. While the IETF may also choose to define one or more specific approaches to manipulate address pool, such as NETCONF or RESTCONF with address pool YANG data model.In modern network management system, the NETCONF or RESTCONF is used widely, the device implements as the NETCONF or RESTCONF server, and the network management system implements as the NETCONF or RESTCONF client, that achieving more automated network management.

11. Security Considerations

12. Acknowledgements

N/A.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

13.2. Informative References

[RFC6674] Brockners, F., Gundavelli, S., Speicher, S., and D. Ward, "Gateway-Initiated Dual-Stack Lite Deployment", RFC 6674, DOI 10.17487/RFC6674, July 2012, <http://www.rfc-editor.org/info/rfc6674>.

[RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <http://www.rfc-editor.org/info/rfc6888>.

Authors' Addresses

Chen Li
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: lichen.bri@chinatelecom.cn

Chongfeng Xie
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: xiechf.bri@chinatelecom.cn

Jun Bi
Tsinghua University
3-212, FIT Building, Tsinghua University, Haidian District
Beijing 100084
P.R. China

Email: junbi@tsinghua.edu.cn

Weiping Xu
Huawei Technologies
Bantian, Longgang District
shenzhen 518129
P.R. China

Email: xuweiping@huawei.com

Internet Working Group
Internet Draft
Intended status: Informational
Expires: September 2017

C. Xie
Q. Sun
China Telecom
W. Xu
W. Liu
Huawei
I. Farrer
N. Kowalewski
Deutsche Telekom AG
Y. Cheng
China Unicom
March 12, 2017

Problem statement for centralized address management
draft-xie-ps-centralized-address-management-02

Status of this Memo

Copyright Notice

Abstract

The increase in number, diversity and complexity of devices and services in modern networks bring new challenges for the management

of network resources, such as IP addresses, network prefixes, bandwidth, and services that utilize such resources. This draft contains a problem statement for IP address management and defines requirements with practical use cases provided by operators.

Table of Contents

1. Introduction

The increase in number, diversity and complexity of modern network devices and services bring new challenges for the management of network resources, such as IP addresses, bandwidth, and services that utilize such resources. However, current approaches for address management often result in sub-optimal allocation efficiency and significant complexity for using, sharing and sharing such resources.
Address resources are often managed across multiple, partly disconnected technical systems which have limited means of model based inter-operation. In the interest of reducing complexity, improve utilization of resources and reduce overall associated OPEX and CAPEX, operators are looking for an intelligent, agile and flexible integrated approach to control and manage IP address resources. Assignment of such resources should be possible across many services, and offer means of categorizing, selecting and decision making on the assignment and revocation of address resources.

?

Among the resources aforementioned, the relevance of address management gained traction by operators as it is a fundamental precursor for the provision of Internet connectivity and services. This draft describes problems and requirements of address management with solid and practical use cases provided by operators.

IPAM (IP address management), is a means of planning, tracking, and managing the Internet Protocol address space used in a network. This topic is increasingly important as aforementioned that networks are deployed with increasing in number, diversity and complexity of modern network devices and services, resulting in more and larger address pools, different subnetting techniques, and more complex 128-bit hexadecimal numbers for IPv6, which are significant less easily human-readable than IPv4 addresses. IPv6 networking, mobile computing, multi-homing and virtualization of compute and network functions require a much more dynamic approach to IP address management. [WIKI]

In some scenarios, the address management system is integrated with the operator’s network. For example, the address system integrated in CMTS (Cable Modem Termination Systems), which is used to allocate

specific IP addresses and options to CMs (Cable Modems).
The second example is the address system integrated in Network Function Virtualization Infrastructure (NFVI), which is used to assign specified IP address(es) to VMs (Virtual Machines).
The third example is the address system in SDN networks, the SDN controller could learn IP address of two inter-communication hosts, and then compute and configure an optimized forwarding path between them.

In the examples above, the address allocation policy, e.g., specific IP address assigned to a specific VM, usually originates from a management system, e.g, OSS, OpenStack, SDN controller, DHCP server instance. Many such systems are configured rather statically, via CLI or per configuration file.

This approach poses the following problems for operators:

- o Low allocation efficiency due to pre-allocation
- o Manual configuration of address policy, with risk for consistency in applying policy
- o Complexity in making real-time changes to assignment
- o Lack of an open, programmable interface between systems which requires IP addresses and the Management Systems handling the respective IP address resources

Address pool management is a sub-issue of address management.
Currently, operators are facing the following issues:

1) The need to control and share addresses among devices
   a) Supply of IPv4 addresses is short of has even ended; the remaining IPv4 address pools do usually no longer consist of large blocks of consecutive addresses, but of a randomly scattered sets of many small blocks or even of independent individual addresses
   b) It is complicated to configure all the address pools statically in Broadband Network Gateways (BNGs).
   c) Sometimes, the address pools need to transition from one BNG to another .
2) The need to control and share addresses among entities or functions
   a) For IPv6 transition technologies, e.g. DS-Lite, lw4over6, etc., the entities need to be configured with IPv4 and IPv6 address pools, as well as with mapping information between individual address resources.
   b) Different address pools may be needed to be configured on each transition instance for HA (High Availability) support.
   c) The level of utilization of address pools may vary during different transition periods.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology

IPAM: IP address management

4. Problems and Use Cases

The BNG, which manages one of more routable IP addresses on behalf of each subscriber, should be configured with the IP address pools allocated to subscribers. However, operators are increasingly challenged by the IPv4 address shortage and IPv4 address pools are scattered into many blocks as small as an IPv4/24 per in many cases. In the worst case configuration of such address pools on a large number of Broadband Network Gateway (BNG) has to be done manually by for operators and is labor intensive. For large scale MAN, there can a three digit number of BNGs to configure.

Usual approaches of manual configuration on BNGs with such data in a static way will not only create great workload, it also limits utilization efficiency of the address pools when the number of subscribers varies or shrinks at a given BNG instance.

With NFV technology maturing, it can be envisioned that the edge of the IP network will become a software-based virtualized vBNG entity itself, so the network element itself is dynamically created and changed. Such virtualized network elements are going to become more common and may be launched and withdrawn dynamically, based on actual traffic and user load, and an efficient dynamic assignments and re-use of address resources will be much more necessary than with a classical hardware-based entities.

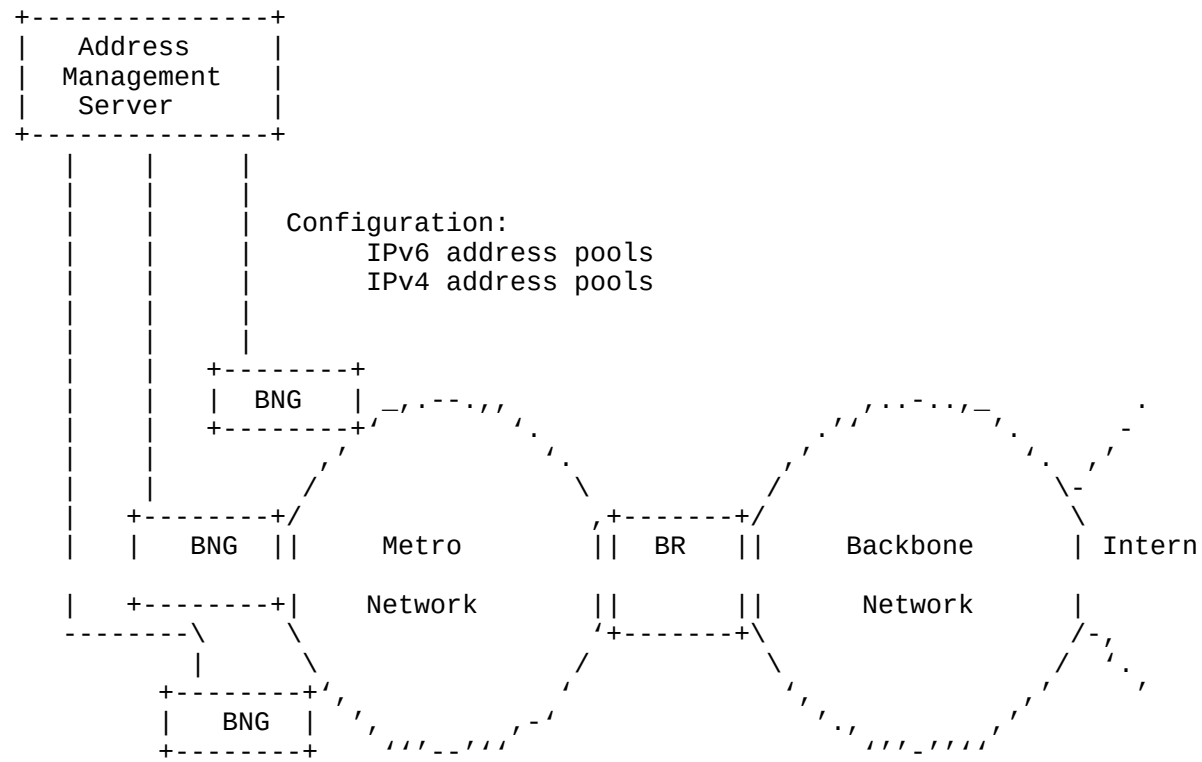

Figure 1 Address pools configuration on the BNGs

Figure 1 illustrates address pool configuration for BNGs. Each BNG requires configuration with several IPv4 and IPv6 address pools used

for allocation to subscribers. Those address pools are configured through an API from a centralized Address Management Server. Typical examples include IPv4 and IPv6 address pool configuration. The centralized management approach is very crucial for dynamically service creation that concerned Virtual BNGs

The second use case for address pool configuration is for IPv6 migration. IPv6 transition mechanisms (e.g. DS-Lite, lw4over6, etc.), need to be configured with address pools to be used as translated routable addresses. When high availability features, e.g. active-active/active-standby failover mechanism, are used, different address pools may need to be configured on each transition instance. This will further increase the number of address pools that need to be configured.

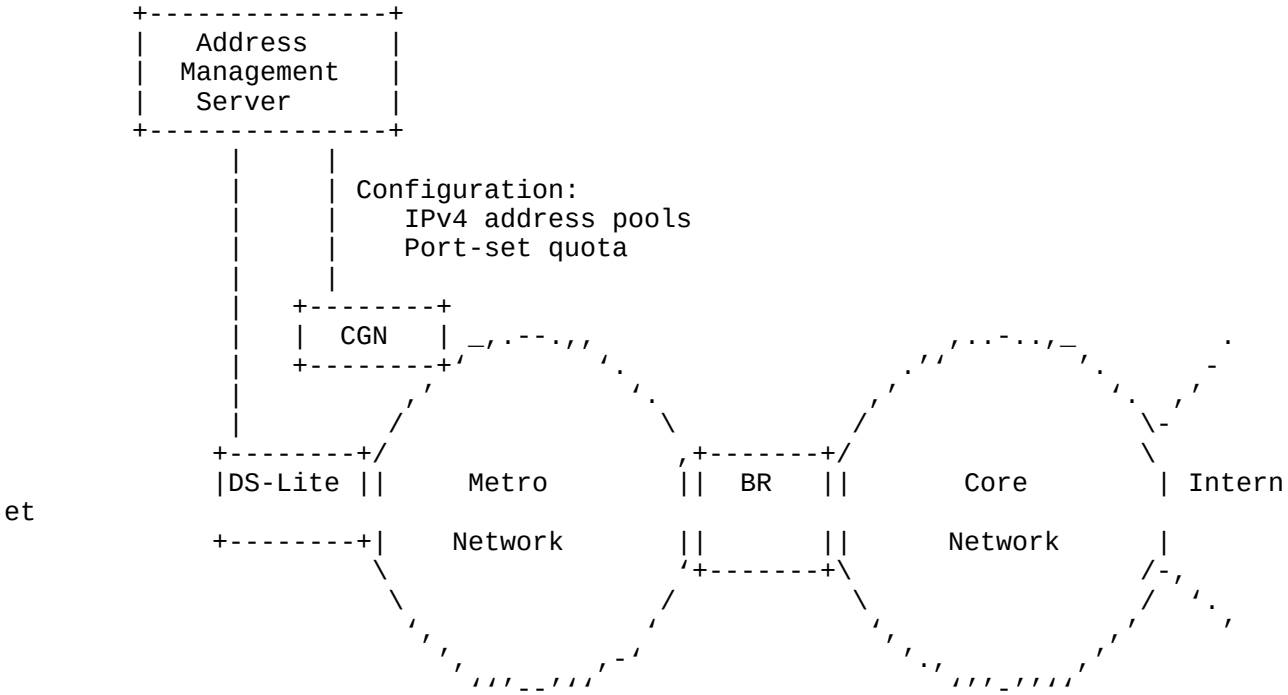

Figure 2 Configuring address pools on IPv6 transition devices

Figure 2 illustrates address configuration on the IPv6 transition devices. For example, the DS-Lite AFTR and the CGN devices need both

be configured with aligned information of the IPv4 address pool that is used. Those address pools are configured through an API from centralized Address Management Server.

The third use case for address pool configuration is IPAM. Nowadays in provider environments, address management is implemented at various levels, from centrally aggregated spreadsheets to application specific databases/software (IPAM). Many IPAM software packages implement RESTful APIs so that organizations that employ modern operational methods like DevOps can use and expand IPAM for their needs, while at the same time establishing a centralized database to administer their IP address resources. Often such systems need to be integrated with provisioning systems for domain name resolution functions.

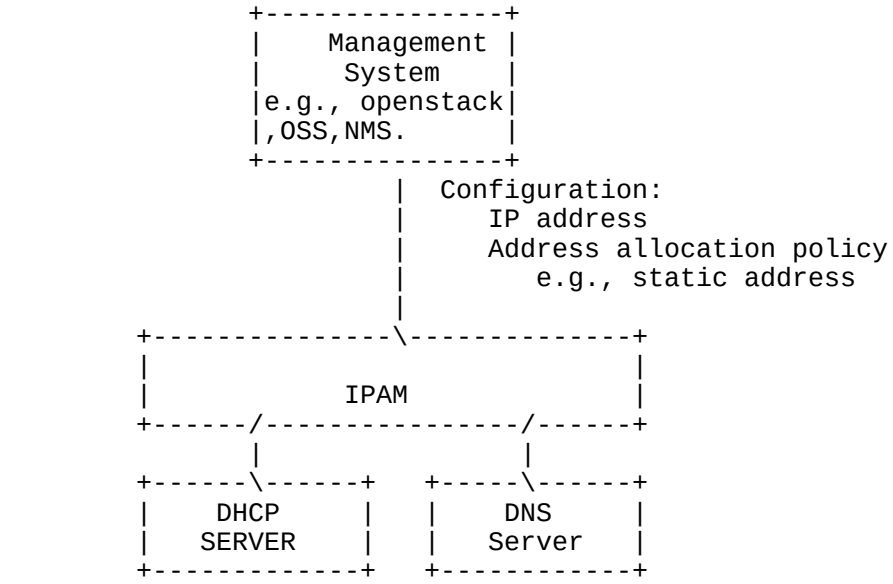

Figure 3 Address configuration API of IPAM

Figure 3 illustrates one possible approach of a general address configuration model where an network management system of OSS is triggering the IPAM tool to perform configuration actions on network elements. A management system, like an instance of OpenStack, of OSS, NMS, could configure address and address allocation policy through API. Typical policy example is specific static IP address allocate to specific host.

in Figure 3, in the CMTS case, operations support system(OSS) or control system defines the address allocation policy, deploys resources to the CMTS device through an open, programmable interface. Then the CM would get its individually customized IP address and DHCP options from the designated address management sub-system in the CMTS.

In the Network Function Virtualization Infrastructure(NFVI) case, the Management System (e.g., OpenStack) designs the address allocation policy, deploys it to the IPAM tool through an open, programmable interface. Then the VM could get customized IP address from IPAM tool.

In SDN network scenario, two host communicate pass through a SDN network.

The Management System(SDN controller) get the IP address of the two inter-communication hosts from address management system through an open, programmable interface, then the SDN controller could design an optimized forwarding path, and deploy it into forwarding plane.

Another common model is that the MNS/OSS and IPAM perform address management on different levels of granularity. The overall authoritative ownership of all address resources lies with the IPAM, and the resources available in there are subject to a formally regulated assignment process (e.g. ARIN, RIPE, etc.). From IPAM, blocks of addresses can be requested according to inherently defined IP Address assignment policy. Requests are made by or on behalf of IP address consuming entities, typically by provisioning intermediaries like MNS, OSS. These systems may further break down the resource according to application specific substructures (e.g. DNS, DHCPv4, DHCPv6, OpenStack, ...) and sub-delegate them as needed.

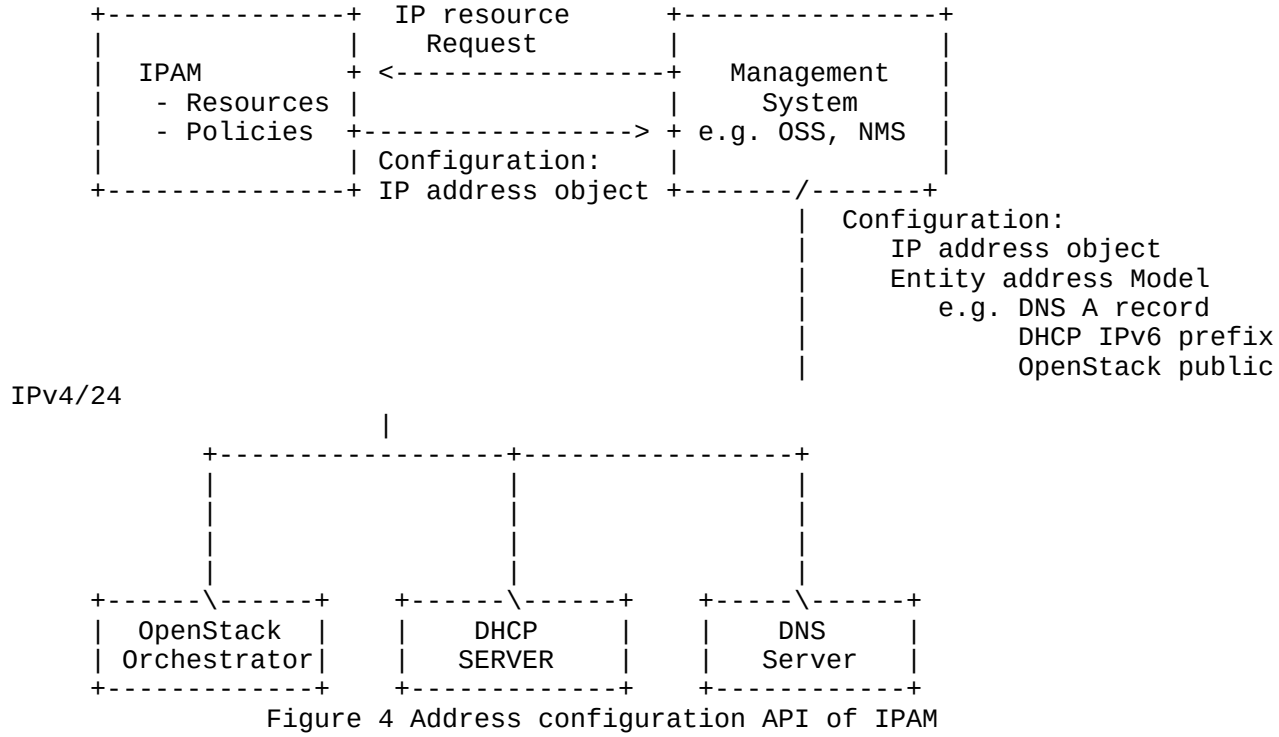

Figure 4 Address configuration API of IPAM

Figure 4 illustrates such a case where the address resources and management policy is represented in the IPAM tool, and the management system relies on an API to the IPAM system to offer the proper set of resources upon request based on an IPAM inherently defined and managed assignment policy. All consuming entities, such as the management system and the resource consuming target entities, like an instance of OpenStack, OSS, NMS, are configured with addresses as per an entity specific allocation model through API.

An examples in the CMTS case could be the deployment of a new access router instance which requires new addresses for the expected new users be available for them to connect. Such addresses need to be deployed in the respective DHCPv4 and DHCPv6 entities. To achieve that, the MNS would request resources from IPAM and assigns the specific /48 address pool to a specific DHCPv6 instance, as well as adding a specific set of IPv4 /24 in a DHCPv4 instance.

As example for a Network Function Virtualization Infrastructure (NFVI) case could be, that at the same time the NMS may need to query for a small set of internal IP resources for a newly to be launched set of additional machines to scale up the VOIP service for these new additional access users. NMS goes out to request these resources from IPAM, adds them to the resources that the OpenStack Orchestrator is aware of and triggers creation of the newly required VMs and virtual networks.

The SDN case, the NMNS would instruct the OpenStack Orchestrator to setup the entities and provide the pool of require IP address endpoints respective

5. Requirements

Based on the analysis above, some requirements for IP address management can be highlighted as following:

1) An integrated, centralized IP address management is desirable as it offers an aggregated view on all stages of the life cycle of IP address resources, from selection, allocation, assignment to reclaiming them into to free resources in an optimized and efficient way.
2) The approach needs to be much more dynamic and act on a much finer granularity than in the past, since address consumption in each device is changing over time, and resource usage can dynamically change over time based on actual user, service, traffic or session volume. A fast return of unused resources for reassignment is of high value.
3) IP address resource assignment policies have to be adaptable to a broad variety of usage scenarios and multiple types of network entities - physical and virtual. Examples are various types of network IP equipment, i.e., BNG, vBNG, CGN, FW, etc, which all need to be supported with resources - directly or indirectly - through the same IP address management server.
4) IP address management needs to be cable of handling IPv4 and IPv6 resources, including sub-netting, and prefixes in any valid configurable prefix length. All well defined and RFC covered address types should be administrable.
5) Overlapping pools of private addresses must be supported.

It should be pointed out that the IP address management server SHALL meet additional requirements of high reliability, availability, security and performance, according to best practices for mission critical infrastructure, but these aspects are considers out of scope of this document.

6. Related IETF work

TBD

7. Security Considerations

TBD.

8. IANA Considerations

No IANA action is needed for this document.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[WIKI] https://en.wikipedia.org/wiki/IP_address_management

10. Acknowledgments

The authors of this draft would like to thank the following persons for the provided valuable feedback and contributions: Benoit Claise, Marc Blancet, Yu Fu, John Strassner, Jun Bi, Diego Lopez, Zhiheng Liu, Laurent Ciavaglia, Fred Baker, Joel Jaeggli, Will Liu, Giuseppe Fioccola.

Authors' Addresses

Chongfeng Xie
China Telecom Beijing Research Institute
China Telecom Beijing Information Science&Technology Innovation Park
Beiqijia Town Changping District Beijing 102209 China
Email: xiechf.bri@chinatelecom.cn

Qiong Sun
China Telecom Beijing Research Institute
China Telecom Beijing Information Science&Technology Innovation Park
Beiqijia Town Changping District Beijing 102209 China
Email: sunqiong@ctbri.com.cn

Weiping Xu
Huawei Technologies Co., Ltd.
Bantian, Longgang district
Shenzhen 518129, China
Email: xuweiping@huawei.com

Will(Shucheng) Liu
Huawei Technologies

Bantian, Longgang District, Shenzhen 518129
P.R. China
Email: liushucheng@huawei.com

Ian Farrer
Deutsche Telekom AG
CTO-ATI, Landgrabenweg 151
Bonn, NRW 53227
Germany
Email: ian.farrer@telekom.de

Normen B. Kowalewski
Deutsche Telekom AG
CTO-ATI, Landgrabenweg 151
Bonn, NRW 53227
Germany
Email: normen.kowalewski@telekom.de

Ying Cheng
China Unicom
No.21 Financial Street, XiCheng District
Beijing 100033
China
Email: chengying10@chinaunicom.cn