ANIMA WG S. Jiang, Ed.
Internet-Draft Z. Du
Intended status: Informational Huawei Technologies Co., Ltd
Expires: June 18, 2018 B. Carpenter
Univ. of Auckland
Q. Sun
China Telecom
December 15, 2017

Autonomic IPv6 Edge Prefix Management in Large-scale Networks
draft-ietf-anima-prefix-management-07

Abstract

This document defines two autonomic technical objectives for IPv6 prefix management at the edge of large-scale ISP networks, with an extension to support IPv4 prefixes. An important purpose of the document is to use it for validation of the design of various components of the autonomic networking infrastructure.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The original purpose of this document was to validate the design of the Autonomic Networking Infrastructure (ANI) for a realistic use case. It shows how the ANI can be applied to IP prefix delegation and it outlines approaches to build a system to do this. A fully standardized solution would require more details, so this document is informational in nature.

This document defines two autonomic technical objectives for IPv6 prefix management in large-scale networks, with an extension to support IPv4 prefixes. The background to Autonomic Networking (AN) is described in [RFC7575] and [RFC7576]. The GeneRic Autonomic Signaling Protocol (GRASP) is specified by [I-D.ietf-anima-grasp] and can make use of the proposed technical objectives to provide a solution for autonomic prefix management. An important purpose of the present document is to use it for validation of the design of GRASP and other components of the autonomic networking infrastructure described in [I-D.ietf-anima-reference-model].

This document is not a complete functional specification of an autonomic prefix management system and it does not describe all detailed aspects of the GRASP objective parameters and Autonomic Service Agent (ASA) procedures necessary to build a complete system. Instead, it describes the architectural framework utilizing the components of the ANI, outlines the different deployment options and aspects, and defines GRASP objectives for use in building the system. It also provides some basic parameter examples.

This document is not intended to solve all cases of IPv6 prefix management. In fact, it assumes that the network's main infrastructure elements already have addresses and prefixes. The document is dedicated to how to make IPv6 prefix management at the edges of large-scale networks as autonomic as possible. It is specifically written for service provider (ISP) networks. Although there are similarities between ISPs and large enterprise networks, the requirements for the two use cases differ. In any case, the scope of the solution is expected to be limited, like any autonomic network, to a single management domain.

However, the solution is designed in a general way. Its use for a broader scope than edge prefixes, including some or all infrastructure prefixes, is left for future discussion.

A complete solution has many aspects that are not discussed here. Once prefixes have been assigned to routers, they need to be communicated to the routing system as they are brought into use. Similarly, when prefixes are released, they need to be removed from the routing system. Different operators may have different policies about prefix lifetimes, and they may prefer to have centralized or distributed pools of spare prefixes. In an autonomic network, these are properties decided by the design of the relevant ASAs. The GRASP objectives are simply building blocks.

A particular risk of distributed prefix allocation in large networks is that over time, it might lead to fragmentation of the address space and an undesirable increase in the interior routing protocol tables. The extent of this risk depends on the algorithms and policies used by the ASAs. Mitigating this risk might even become an autonomic function in itself.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses terminology defined in [RFC7575].

3. Problem Statement

The autonomic networking use case considered here is autonomic IPv6 prefix management at the edge of large-scale ISP networks.

Although DHCPv6 Prefix Delegation [RFC3633] supports automated delegation of IPv6 prefixes from one router to another, prefix management still largely depends on human planning. In other words, there is no basic information or policy to support autonomic decisions on the prefix length that each router should request or be delegated, according to its role in the network. Roles could be defined separately for individual devices or could be generic (edge router, interior router, etc.). Furthermore, IPv6 prefix management by humans tends to be rigid and static after initial planning.

The problem to be solved by autonomic networking is how to dynamically manage IPv6 address space in large-scale networks, so that IPv6 addresses can be used efficiently. Here, we limit the problem to assignment of prefixes at the edge of the network, close to access routers that support individual fixed-line subscribers, mobile customers, and corporate customers. We assume that the core infrastructure of the network has already been established with appropriately assigned prefixes. The AN approach discussed in this document is based on the assumption that there is a generic discovery and negotiation protocol that enables direct negotiation between intelligent IP routers. GRASP [I-D.ietf-anima-grasp] is intended to be such a protocol.

3.1. Intended User and Administrator Experience

The intended experience is, for the administrators of a large-scale network, that the management of IPv6 address space at the edge of the network can be run with minimum effort, as devices at the edge are added and removed and as customers of all kinds join and leave the network. In the ideal scenario, the administrators only have to specify a single IPv6 prefix for the whole network and the initial prefix length for each device role. As far as users are concerned, IPv6 prefix assignment would occur exactly as it does in any other network.

The actual prefix usage needs to be logged for potential offline management operations including audit and security incident tracing.

3.2. Analysis of Parameters and Information Involved

For specific purposes of address management, a few parameters are involved on each edge device (some of them can be pre-configured before they are connected). They include:

- o Identity, authentication and authorization of this device. This is expected to use the autonomic networking secure bootstrap process [I-D.ietf-anima-bootstrapping-keyinfra], following which the device could safely take part in autonomic operations.

- o Role of this device. Some example roles are discussed in Section 6.1.

- o An IPv6 prefix length for this device.

- o An IPv6 prefix that is assigned to this device and its downstream devices.

A few parameters are involved in the network as a whole. They are:

- o Identity of a trust anchor, which is a certification authority (CA) maintained by the network administrators, used during the secure bootstrap process.

- o Total IPv6 address space available for edge devices. It is a pool of one or several IPv6 prefixes.

- o The initial prefix length for each device role.

3.2.1. Parameters each device can define for itself

This section identifies those of the above parameters that do not need external information in order for the devices concerned to set them to a reasonable default value after bootstrap or after a network disruption. There are few of these:

- o Default role of this device.

- o Default IPv6 prefix length for this device.

- o Cryptographic identity of this device, as needed for secure bootstrapping [I-D.ietf-anima-bootstrapping-keyinfra].

The device may be shipped from the manufacturer with pre-configured role and default prefix length, which could be modified by an autonomic mechanism. Its cryptographic identity will be installed by its manufacturer.

3.2.2. Information needed from network operations

This section identifies those parameters that might need operational input in order for the devices concerned to set them to a non-default value.

- o Non-default value for the IPv6 prefix length for this device. This needs to be decided based on the role of this device.

- o The initial prefix length for each device role.

- o Whether to allow the device to request more address space.

- o The policy when to request more address space, for example, if the address usage reaches a certain limit or percentage.

3.2.3. Comparison with current solutions

This section briefly compares the above use case with current solutions. Currently, the address management is still largely dependent on human planning. It is rigid and static after initial planning. Address requests will fail if the configured address space is used up.

Some autonomic and dynamic address management functions may be achievable by extending the existing protocols, for example, extending DHCPv6-PD (DHCPv6 Prefix Delegation, [RFC3633]) to request IPv6 prefixes according to the device role. However, defining uniform device roles may not be a practical task. Some functions are not suitable to be achieved by any existing protocols.

Using a generic autonomic discovery and negotiation protocol instead of specific solutions has the advantage that additional parameters can be included in the autonomic solution without creating new mechanisms. This is the principal argument for a generic approach.

3.3. Interaction with other devices

3.3.1. Information needed from other devices

This section identifies those of the above parameters that need external information from neighbor devices (including the upstream devices). In many cases, two-way dialogue with neighbor devices is needed to set or optimize them.

- o Identity of a trust anchor.
- o The device will need to discover a device, from which it can acquire IPv6 address space.
- o The initial prefix length for each device role, particularly for its own downstream devices.
- o The default value of the IPv6 prefix length may be overridden by a non-default value.
- o The device will need to request and acquire one or more IPv6 prefixes that can be assigned to this device and its downstream devices.
- o The device may respond to prefix delegation requests from its downstream devices.
- o The device may require to be assigned more IPv6 address space, if it used up its assigned IPv6 address space.

3.3.2. Monitoring, diagnostics and reporting

This section discusses what role devices should play in monitoring, fault diagnosis, and reporting.

- o The actual address assignments need to be logged for potential offline management operations.
- o In general, the usage situation of address space should be reported to the network administrators, in an abstract way, for example, statistics or visualized report.
- o A forecast of address exhaustion should be reported.

4. Autonomic Edge Prefix Management Solution

This section introduces the building blocks for an autonomic edge prefix management solution. As noted in Section 1, this is not a complete description of a solution, which will depend on the detailed design of the relevant Autonomic Service Agents. It uses the generic discovery and negotiation protocol defined by [I-D.ietf-anima-grasp]. The relevant GRASP objectives are defined in Section 5.

The procedures described below are carried out by an Autonomic Service Agent (ASA) in each device that participates in the solution. We will refer to this as the PrefixManager ASA.

4.1. Behaviors on prefix requesting device

If the device containing a PrefixManager ASA has used up its address pool, it can request more space according to its requirements. It should decide the length of the requested prefix and request it by the mechanism described in Section 6. Note that although the device's role may define certain default allocation lengths, those defaults might be changed dynamically, and the device might request more, or less, address space due to some local operational heuristic.

A PrefixManager ASA that needs additional address space should firstly discover peers that may be able to provide extra address space. The ASA should send out a GRASP Discovery message that contains a PrefixManager Objective option (see Section 5.1) in order to discover peers also supporting that option. Then it should choose one such peer, most likely the first to respond.

If the GRASP discovery Response message carries a divert option pointing to an off-link PrefixManager ASA, the requesting ASA may initiate negotiation with that ASA diverted device to find out whether it can provide the requested length prefix.

In any case, the requesting ASA will act as a GRASP negotiation initiator by sending a GRASP Request message with a PrefixManager Objective option. The ASA indicates in this option the length of the requested prefix. This starts a GRASP negotiation process.

During the subsequent negotiation, the ASA will decide at each step whether to accept the offered prefix. That decision, and the decision to end negotiation, is an implementation choice.

The ASA could alternatively initiate rapid mode GRASP discovery with an embedded negotiation request, if it is implemented.

4.2. Behaviors on prefix providing device

At least one device on the network must be configured with the initial pool of available prefixes mentioned in Section 3.2. Apart from that requirement, any device may act as a prefix providing device.

A device that receives a Discovery message with a PrefixManager Objective option should respond with a GRASP Response message if it contains a PrefixManager ASA. Further details of the discovery process are described in [I-D.ietf-anima-grasp]. When this ASA receives a subsequent Request message, it should conduct a GRASP negotiation sequence, using Negotiate, Confirm-waiting, and Negotiation-ending messages as appropriate. The Negotiate messages carry a PrefixManager Objective option, which will indicate the prefix and its length offered to the requesting ASA. As described in [I-D.ietf-anima-grasp], negotiation will continue until either end stops it with a Negotiation-ending message. If the negotiation succeeds, the prefix providing ASA will remove the negotiated prefix from its pool, and the requesting ASA will add it. If the negotiation fails, the party sending the Negotiation-ending message may include an error code string.

During the negotiation, the ASA will decide at each step how large a prefix to offer. That decision, and the decision to end negotiation, is an implementation choice.

The ASA could alternatively negotiate in response to rapid mode GRASP discovery, if it is implemented.

This specification is independent of whether the PrefixManager ASAs are all embedded in routers, but that would be a rather natural scenario. In a hierarchical network topology, a given router typically provide prefixes for routers below it in the hierarchy, and it is also likely to contain the first PrefixManager ASA discovered by those downstream routers. However, the GRASP discovery model, including its Redirect feature, means that this is not an exclusive scenario, and a downstream PrefixManager ASA could negotiate a new prefix with a device other than its upstream router.

A resource shortage may cause the gateway router to request more resource in turn from its own upstream device. This would be another independent GRASP discovery and negotiation process. During the processing time, the gateway router should send a Confirm-waiting Message to the initial requesting router, to extend its timeout. When the new resource becomes available, the gateway router responds with a GRASP Negotiate message with a prefix length matching the request.

The algorithm to choose which prefixes to assign on the prefix providing devices is an implementation choice.

4.3. Behavior after Successful Negotiation

Upon receiving a GRASP Negotiation-ending message that indicates that an acceptable prefix length is available, the requesting device may use the negotiated prefix without further messages.

There are use cases where the ANI/GRASP based prefix management approach can work together with DHCPv6-PD [RFC3633] as a complement. For example, the ANI/GRASP based method can be used intra-domain, while the DHCPv6-PD method works inter-domain (i.e., across an administrative boundary). Also, ANI/GRASP can be used inside the domain, and DHCP/DHCPv6-PD be used on the edge of the domain to client (non-ANI devices). Another similar use case would be ANI/GRASP inside the domain, with RADIUS [RFC2865] providing prefixes to client devices.

4.4. Prefix logging

Within the autonomic prefix management, all the prefix assignment is done by devices without human intervention. It may be required to record all the prefix assignment history, for example to detect or trace lost prefixes after outages, or to meet legal requirements. However, the logging and reporting process is out of scope for this document.

5. Autonomic Prefix Management Objectives

This section defines the GRASP technical objective options that are used to support autonomic prefix management.

5.1. Edge Prefix Objective Option

The PrefixManager Objective option is a GRASP objective option conforming to [I-D.ietf-anima-grasp]. Its name is "PrefixManager" (see Section 8) and it carries the following data items as its value: the prefix length, and the actual prefix bits. Since GRASP is based on CBOR (Concise Binary Object Representation [RFC7049]), the format of the PrefixManager Objective option is described as follows in CBOR data definition language (CDDL) [I-D.ietf-cbor-cddl]:

```
objective = ["PrefixManager", objective-flags, loop-count,
             [length, ?prefix]]

loop-count = 0..255         ; as in the GRASP specification
objective-flags /=          ; as in the GRASP specification
length = 0..128             ; requested or offered prefix length
prefix = bytes .size 16     ; offered prefix in binary format
```

The use of the 'dry run' mode of GRASP is NOT RECOMMENDED for this objective, because it would require both ASAs to store state about the corresponding negotiation, to no real benefit - the requesting ASA cannot base any decisions on the result of a successful dry run negotiation.

5.2. IPv4 extension

This section presents an extended version of the PrefixManager Objective that supports IPv4 by adding an extra flag:

```
objective = ["PrefixManager", objective-flags, loop-count, prefval]

loop-count = 0..255         ; as in the GRASP specification
objective-flags /=          ; as in the GRASP specification

prefval /= pref6val
pref6val = [version6, length, ?prefix]
version6 = 6
length = 0..128             ; requested or offered prefix length
prefix = bytes .size 16     ; offered prefix in binary format

prefval /= pref4val
pref4val = [version4, length4, ?prefix4]
version4 = 4
length4 = 0..32             ; requested or offered prefix length
prefix4 = bytes .size 4     ; offered prefix in binary format
```

Prefix and address management for IPv4 is considerably more difficult than for IPv6, due to the prevalence of NAT, ambiguous addresses [RFC1918], and address sharing [RFC6346]. These complexities might require further extending the objective with additional fields which are not defined by this document.

6. Prefix Management Parameters

An implementation of a prefix manager MUST include default settings of all necessary parameters. However, within a single administrative domain, the network operator MAY change default parameters for all devices with a certain role. Thus it would be possible to apply an intended policy for every device in a simple way, without traditional configuration files. As noted in Section 4.1, individual autonomic devices may also change their own behavior dynamically.

For example, the network operator could change the default prefix length for each type of role. A prefix management parameters objective, which contains mapping information of device roles and their default prefix lengths, MAY be flooded in the network, through the Autonomic Control Plane (ACP) [I-D.ietf-anima-autonomic-control-plane]. The objective is defined in CDDL as follows:

```
objective = ["PrefixManager.Params", objective-flags, any]

loop-count = 0..255         ; as in the GRASP specification
objective-flags /=          ; as in the GRASP specification
```

The ’any’ object would be the relevant parameter definitions (such as the example below) transmitted as a CBOR object in an appropriate format.

This could be flooded to all nodes, and any PrefixManager ASA that did not receive it for some reason could obtain a copy using GRASP unicast synchronization. Upon receiving the prefix management parameters, every device can decide its default prefix length by matching its own role.

6.1. Example of Prefix Management Parameters

The parameters comprise mapping information of device roles and their default prefix lengths in an autonomic domain. For example, suppose an IPRAN (IP Radio Access Network) operator wants to configure the prefix length of Radio Network Controller Site Gateway (RSG) as 34, the prefix length of Aggregation Site Gateway (ASG) as 44, and the prefix length of Cell Site Gateway (CSG) as 56. This could be described in the value of the PrefixManager.Params objective as:

```
[
   [["role", "RSG"],["prefix_length", 34]],
   [["role", "ASG"],["prefix_length", 44]],
   [["role", "CSG"],["prefix_length", 56]]
]
```

This example is expressed in JSON notation [RFC7159], which is easy to represent in CBOR.

An alternative would be to express the parameters in YANG [RFC7950] using the YANG-to-CBOR mapping [I-D.ietf-core-yang-cbor].

For clarity, the background of the example is introduced below, which can also be regarded as a use case of the mechanism proposed in this document.

An IPRAN network is used for mobile backhaul, including radio stations, RNC (in 3G) or the packet core (in LTE), and the IP network between them as shown in Figure 1. The eNB (Evolved Node B), RNC (Radio Network Controller), SGW (Service Gateway), and MME (Mobility Management Entity) are mobile network entities defined in 3GPP. The CSG, ASG, and RSG are entities defined in the IPRAN solution.

The IPRAN topology shown in Figure 1 includes Ring1 which is the circle following ASG1->RSG1->RSG2->ASG2->ASG1, Ring2 following CSG1->ASG1->ASG2->CSG2->CSG1, and Ring3 following CSG3->ASG1->ASG2->CSG3. In a real deployment of IPRAN, there may be more stations, rings, and routers in the topology, and normally the network is highly dependent on human design and configuration, which is neither flexible nor cost-effective.

```
+------+   +------+
| eNB1 |---| CSG1 |\
+------+   +------+  \   +-------+       +------+          +-------+
              |       \ |  ASG1 |-------| RSG1 |-----------|SGW/MME|
              |  Ring2  +-------+       +------+ \        /+-------+
+------+   +------+     /     |              |      \    /
| eNB2 |---| CSG2 | \  /      |      Ring1   |        \/
+------+   +------+   \  Ring3|              |        /\
                     / \      |              |      /   \
+------+   +------+ /    \ +-------+      +------+/       \+-------+
| eNB3 |---| CSG3 |--------|  ASG2 |------| RSG2 |---------|  RNC  |
+------+   +------+        +-------+      +------+         +-------+
```

Figure 1: IPRAN Topology Example

If ANI/GRASP is supported in the IPRAN network, the network nodes should be able to negotiate with each other, and make some autonomic decisions according to their own status and the information collected from the network. The Prefix Management Parameters should be part of the information they communicate.

The routers should know the role of their neighbors, the default prefix length for each type of role, etc. An ASG should be able to request prefixes from an RSG, and an CSG should be able to request prefixes from an ASG. In each request, the ASG/CSG should indicate the required prefix length, or its role, which implies what length it needs by default.

7. Security Considerations

Relevant security issues are discussed in [I-D.ietf-anima-grasp]. The preferred security model is that devices are trusted following the secure bootstrap procedure [I-D.ietf-anima-bootstrapping-keyinfra] and that a secure Autonomic Control Plane (ACP) [I-D.ietf-anima-autonomic-control-plane] is in place.

It is RECOMMENDED that DHCPv6-PD, if used, should be operated using DHCPv6 authentication or Secure DHCPv6.

8. IANA Considerations

This document defines two new GRASP Objective Option names, "PrefixManager" and "PrefixManager.Params". The IANA is requested to add these to the GRASP Objective Names Table registry defined by [I-D.ietf-anima-grasp] (if approved).

9. Acknowledgements

Valuable comments were received from William Atwood, Fred Baker, Michael Behringer, Ben Campbell, Laurent Ciavaglia, Toerless Eckert, Joel Halpern, Russ Housley, Geoff Huston, Warren Kumari, Dan Romascanu, and Chongfeng Xie.

10. Change log [RFC Editor: Please remove]

draft-jiang-anima-prefix-management-00: original version, 2014-10-25.

draft-jiang-anima-prefix-management-01: add intent example and coauthor Zongpeng Du, 2015-05-04.

draft-jiang-anima-prefix-management-02: update references and the format of the prefix management intent, 2015-10-14.

draft-ietf-anima-prefix-management-00: WG adoption, clarify scope and purpose, update text to match latest GRASP spec, 2016-01-11.

draft-ietf-anima-prefix-management-01: minor update, 2016-07-08.

draft-ietf-anima-prefix-management-02: replaced intent discussion by parameter setting, 2017-01-10.

draft-ietf-anima-prefix-management-03: corrected object format, improved parameter setting example, 2017-03-10.

draft-ietf-anima-prefix-management-04: add more explanations about the solution, add IPv4 options, removed PD flag, 2017-06-23.

draft-ietf-anima-prefix-management-05: selected one IPv4 option, updated references, 2017-08-14.

draft-ietf-anima-prefix-management-06: handled IETF Last Call comments, 2017-10-18.

draft-ietf-anima-prefix-management-07: handled IESG comments, 2017-12-18.

11. References

11.1. Normative References

[I-D.ietf-anima-autonomic-control-plane]
Behringer, M., Eckert, T., and S. Bjarnason, "An Autonomic Control Plane (ACP)", draft-ietf-anima-autonomic-control-plane-12 (work in progress), October 2017.

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-09 (work in progress), October 2017.

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

[I-D.ietf-cbor-cddl]
Birkholz, H., Vigano, C., and C. Bormann, "Concise data definition language (CDDL): a notational convention to express CBOR data structures", draft-ietf-cbor-cddl-00 (work in progress), July 2017.

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

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <https://www.rfc-editor.org/info/rfc3633>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

11.2. Informative References

[I-D.ietf-anima-reference-model]
Behringer, M., Carpenter, B., Eckert, T., Ciavaglia, L., Pierre, P., Liu, B., Nobre, J., and J. Strassner, "A Reference Model for Autonomic Networking", draft-ietf-anima-reference-model-05 (work in progress), October 2017.

[I-D.ietf-core-yang-cbor]
Veillette, M., Pelov, A., Somaraju, A., Turner, R., and A. Minaburo, "CBOR Encoding of Data Modeled with YANG", draft-ietf-core-yang-cbor-05 (work in progress), August 2017.

[I-D.liu-dhc-dhcp-yang-model]
Liu, B., Lou, K., and C. Chen, "Yang Data Model for DHCP Protocol", draft-liu-dhc-dhcp-yang-model-06 (work in progress), March 2017.

[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>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/info/rfc2865>.

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, DOI 10.17487/RFC3046, January 2001, <https://www.rfc-editor.org/info/rfc3046>.

[RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, DOI 10.17487/RFC6221, May 2011, <https://www.rfc-editor.org/info/rfc6221>.

[RFC6346] Bush, R., Ed., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, DOI 10.17487/RFC6346, August 2011, <https://www.rfc-editor.org/info/rfc6346>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[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>.

[RFC7576] Jiang, S., Carpenter, B., and M. Behringer, "General Gap Analysis for Autonomic Networking", RFC 7576, DOI 10.17487/RFC7576, June 2015, <https://www.rfc-editor.org/info/rfc7576>.

Appendix A. Deployment Overview

This Appendix includes logical deployment models, and explanations of the target deployment models. The purpose is to help in understanding the mechanism of the document.

This Appendix includes two sub-sections: A.1 for the two most common DHCP deployment models, and A.2 for the proposed PD deployment model. It should be noted that these are just examples, and there are many more deployment models.

A.1. Address & Prefix management with DHCP

Edge DHCP server deployment requires every edge router connecting to CPE to be a DHCP server assigning IPv4/IPv6 addresses to CPE - and optionally IPv6 prefixes via DHCPv6-PD for IPv6 capable CPE that are router and have LANs behind them.

```
                                              edge
        dynamic, "netconf/YANG"             interfaces
         <---------------> +-------------+
+------+    <- telemetry   | edge router/|-+  -----  +-----+
|config|  .... Domain ...  | DHCP server | |  ...    | CPE |+  LANs
|server|                   +-------------+ |  -----  +-----+| (---| )
+------+                    +--------------+  DHCP/   +-----+
                                            DHCPv6 / PD
```

Figure 2: DHCP Deployment Model without a Central DHCP Server

This requires various coordination functions via some backend system depicted as "config server": The address prefixes on the edge interfaces should be slightly larger than required for the number of CPEs connected so that the overall address space is best used.

The config server needs to provision edge interface address prefixes and DHCP parameters for every edge router. If too fine grained prefixes are used, this will result in large routing tables across the "Domain". If too coarse grained prefixes are used, address space is wasted. (This is less of a concern for IPv6, but if the model includes IPv4, it is a very serious concern.)

There is no standard describing algorithms for how configuration servers would best perform this ongoing dynamic provisioning to optimize routing table size and address space utilization.

There are currently no complete YANG models that a config server could use to perform these actions (including telemetry of assigned addresses from such distributed DHCP servers).

For example, a YANG model for controlling DHCP server operations is still in draft [I-D.liu-dhc-dhcp-yang-model].

Due to these and other problems of the above model, the more common DHCP deployment model is as follows:

```
+------+                                       edge
|config|    initial, "CLI"                   interfaces
|server| ----------------> +-------------+
+------+                   | edge router/|-+  -----  +-----+
   |     .... Domain ...   | DHCP relay  | |  ...    | CPE |+  LANs
+------+                   +-------------+ |  -----  +-----+| (---| )
|DHCP  |                    +--------------+   DHCP/  +-----+
|server|                                    DHCPv6 / PD
+------+
```

Figure 3: DHCP Deployment Model with a Central DHCP Server

Dynamic provisioning changes to edge routers are avoided by using a central DHCP server and reducing the edge router from DHCP server to DHCP relay. The "configuration" on the edge routers is static, the DHCP relay function inserts "edge interface" and/or subscriber identifying options into DHCP requests from CPE (e.g., [RFC3046], [RFC6221]), the DHCP server has complete policies for address assignments and prefixes useable on every edge-router/interface/ subscriber-group. When the DHCP relay sees the DHCP reply, it inserts static routes for the assigned address/address-prefix into the routing table of the edge router which are then to be distributed by the IGP (or BGP) inside the domain to make the CPE and LANs reachable across the Domain.

There is no comprehensive standardization of these solutions. [RFC3633] section 14, for example, simply refers to "a [non-defined] protocol or other out-of-band communication to add routing information for delegated prefixes into the provider edge router".

A.2. Prefix management with ANI/GRASP

With the proposed use of ANI and Prefix-management ASAs using GRASP, the deployment model is intended to look as follows:

```
|<............ ANI Domain / ACP............>| (...) ........->

                                   Roles
                                     |
                                     v   "Edge routers"
GRASP parameter               +----------+
 Network wide                 |  PM-ASA  | downstream
parameters/policies           |  (DHCP-  | interfaces
     |                        |functions)| ------
     v  "central device"      +----------+
+------+                            ^             +--------+
|PM-ASA|      <............GRASP ....      ....   |  CPE   |-+ (LANs)
+------+             .              v             |(PM-ASA)| |  ---|
     .           +........+   +----------+        +--------+ |
+...........+    . PM-ASA .   |  PM-ASA  | ------  +---------+
.DHCP server.    +........+   |  (DHCP-  | SLAAC/
+...........+  "intermediate  |functions)| DHCP/DHCP-PD
                  router"     +----------+
```

Figure 4: Proposed Deployment Model using ANI/GRASP

The network runs an ANI domain with ACP [I-D.ietf-anima-autonomic-control-plane] between some central device (e.g., router or ANI enabled management device) and the edge routers. ANI/ACP provides a secure, zero-touch communication channel between the devices and enables the use of GRASP[I-D.ietf-anima-grasp] not only for p2p communication, but also for distribution/flooding.

The central devices and edge routers run software in the form of "Autonomic Service Agents" (ASA) to support this document’s autonomic IPv6 edge prefix management (PM). The ASAs for prefix management are called PM-ASAs below, and together comprise the Autonomic Prefix Management Function.

Edge routers can have different roles based on the type and number of CPE attaching to them. Each edge router could be an RSG, ASG, or CSG in mobile aggregation networks (see Section 6.1). Mechanisms outside the scope of this document make routers aware of their roles.

Some considerations about the proposed deployment model are listed as follows.

1. In a minimum Prefix Management solution, the central device uses the "PrefixManager.Params" GRASP Objective introduced in this document to disseminate network wide, per-role parameters to edge routers. The PM-ASA uses the parameters applying to its role to locally configure pre-existing addressing functions. Because PM-ASA does not manage the dynamic assignment of actual IPv6 address prefixes in this case, the following options can be considered:

1.a The edge router connects via downstream interfaces to (host) CPE that each requires an address. The PM-ASA sets up for each such interface a DHCP requesting router (according to [RFC3633]) to request an IPv6 prefix for the interface. The router’s address on the downstream interface can be another parameter from the GRASP Objective. The CPEs assign addresses in the prefix via RAs from the router or the PM-ASA manages a local DHCPv6 server to assign addresses to the CPEs. A central DHCP server acting as the DHCP delegating router (according to [RFC3633]) is required. Its address can be another parameter from the GRASP Objective.

1.b The edge router also connects via downstream interfaces to (customer managed) CPEs that are routers and act as DHCPv6 requesting routers. The need to support this could be derived from role and/or GRASP parameters and the PM-ASA sets up a DHCP relay function to pass on requests to the central DHCP server as in 1.a.

2. In a solution without a central DHCP server, the PM-ASA on the edge routers not only learn parameters from "PrefixManager.Params" but also utilize GRASP to request/negotiate actual IPv6 prefix delegation via the GRASP "PrefixManager" objective described in more detail below. In the most simple case, these prefixes are delegated via this GRASP objective from the PM-ASA in the central device. This

device must be provisioned initially with a large pool of prefixes. The delegated prefixes are then used by the PM-ASA on the edge routers to edge routers to configure prefixes on their downstream interfaces to assign addresses via RA/SLAAC to host CPEs. The PM-ASA may also start local DHCP servers (as in 1.a) to assign addresses via DHCP to CPE from the prefixes it received. This includes both host CPEs requesting IPv6 addresses as well as router CPEs that request IPv6 prefixes. The PM-ASA needs to manage the address pool(s) it has requested via GRASP and allocate sub-address pools to interfaces and the local DHCP servers it starts. It needs to monitor the address utilization and accordingly request more address prefixes if its existing prefixes are exhausted, or return address prefixes when they are unneeded.

This solution is quite similar to the initial described IPv6 DHCP deployment model without central DHCP server, and ANI/ACP/GRASP and the PM-ASA do provide the automation to make this approach work more easily than it is possible today.

3. The address pool(s) from which prefixes are allocated does not need to be taken all from one central location. Edge router PM-ASA that received a big (short) prefix from a central PM-ASA could offer smaller sub-prefixes to neighboring edge-router PM-ASA. GRASP could be used in such a way that the PM-ASA would find and select the objective from the closest neighboring PM-ASA, therefore allowing to maximize aggregation: A PM-ASA would only request further (smaller/ shorter) prefixes when it exhausts its own poll (from the central location) and can not get further large prefixes from that central location anymore. Because the overflow prefixes taken from a topological nearby PM-ASA, the number of longer prefixes that have to be injected into the routing tables is limited and the topological proximity increases the chances that aggregation of prefixes in the IGP can most likely limit the geography in which the longer prefixes need to be routed.

4. Instead of peer-to-peer optimization of prefix delegation, a hierarchy of PM-ASA can be built (indicated in the picture via a dotted intermediate router). This would require additional parameters to the "PrefixManager" objective to allow creating a hierarchy of PM-ASA across which the prefixes can be delegated. This is not detailed further below.

5. In cases where CPEs are also part of the ANI Domain (e.g., "Managed CPE"), then GRASP will extend into the actual customer sites and can equally run a PM-ASA. All the options described in points 1 to 4 above would then apply to the CPE as the edge router with the mayor changes being that a) a CPE router will most likley not need to run DHCPv6-PD itself, but only DHCP address assignment, b) The edge

routers to which the CPE connect would most likely become ideal places to run a hierarchical instance of PD-ASAs on as outlined in point 1.

Authors' Addresses

Sheng Jiang (editor)
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: jiangsheng@huawei.com

Zongpeng Du
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: duzongpeng@huawei.com

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Qiong Sun
China Telecom
No.118, Xizhimennei Street
Beijing 100035
P. R. China

Email: sunqiong@ctbri.com.cn