Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 22, 2017

Z. Kahn
LinkedIn
J. Brzozowski
Comcast
R. White
LinkedIn

February 18, 2017

Requirements for IPv6 Routers
draft-ali-ipv6rtr-reqs-02

Abstract

The Internet is not one network, but rather a collection of networks. The interconnected nature of these networks, and the nature of the interconneted systems that make up these networks, is often more fragile than it appears. Perhaps "robust but fragile" is an overstatement, but the actions of each vendor, implementor, and operator in such an interconneted environment can have a major impact on the stability of the overall Internet (as a system). The widespread adoption of IPv6 could, particularly, disrupt network operations, in a way that impacts the entire system.

This time of transition is an opportune time to take stock of lessons learned through the operation of large scale networks on IPv4, and consider how to apply these lessons to IPv6. This document provides an overview of the design and architectural decisions that attend IPv6 deployment, and a set of IPv6 requirements for routers, switches, and middleboxes deployed in IPv6 networks. The hope of the editors and contributors is to provide the neccessary background to guide equipment manufacturers, protocol implemenetors, and network operators in effective IPv6 deployment.

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 22, 2017.

Table of Contents

1. Introduction

This memo defines and discusses requirements for devices that perform forwarding for Internet Protocol version 6 (IPv6). This can include (but is not limited to) the devices described below.

- Devices which are primarily designed to forward traffic between multiple interfaces. These are normally referred to by the Internet community as routers or, in some cases, intermediate systems.

- Devices which are designed to modify packets rather than "just" forwarding them. These are often referred to by the Internet community as middleboxes. See [RFC7663] for a fuller definition of middleboxes.

Readers should recognize that while this memo applies to IPv6, routers and middleboxes IPv6 packets will often also process IPv4 packets, forward based on MPLS labels, and potentially process many other protocols. This memo will only discuss IPv4, MPLS, and other protocols as they impact the behavior of an IPv6 forwarding device; no attempt is made to specify requirements for protocols other than IPv6. The reader should, therefore, not count on this document as a "sole source of truth," but rather use this document as a guide.

For IPv4 router requirements, readers are referred to [RFC1812]. For simplicity, the term "devices" is used interchangeably with the phrase "routers and middleboxes" and the term "routers" throughout this document. These three terms represent stylistic differences, rather than substantive differences.

This document is broken into the following sections: a review of Internet architecture and principles, requirements relating to device management, requirements related to telemetry, requirements related to IPv6 forwarding and addressing, and future considerations. Following these sections, a short conclusion is provided for review.

1.1. Contributors

Shawn Zandi, Pete Lumbis, Fred Baker, and Lee Howard contributed significant text and ideas to this draft.

1.2. Acknowledgements

The editors and contributors would like to thank....

2. Review of the Internet Architecture

The Internet relies on a number of basic concepts and considerations. These concepts are not explicitly called out in any specification, nor do they necessarily impact protocol design or packet forwarding directly. This section provides an overview of these concepts and considerations to help the reader understand the larger context of this document.

2.1. Robustness Principle

Every point where multiple protocols interact, is an interaction surface that can threaten the robustness of the overall system. While it may seem the global Internet has achieved a level of stability that makes it immune to such considerations, the reality is every network is a complex system, and is therefore subject to massive non repeatable unanticipated failures. Postel’s Robustness Principle countered this problem with a simple statement, explicated in [RFC7922]: "Be conservative in what you do, and liberal in what you accept from others."

However, since this time, it has been noted that following this law allows errors in protocols to accumulate over time, with overall negative effects on the system as a whole. [RFC1918] describes several points in conjunction with this principle that bear updating based on further experience with large scale protocol and network deployments within the Internet community, including:

- Applications should deal with error states gracefully; an application should not degrade in a way that will cause the failure of adjacent systems when possible. For instance, when a routing protocol implementation fails, it should not do so in a way that will cause the spreading of or continued existence of false reachability information, nor should it fail in a way that overloads adjacent routers or interacting protocols and causing a cascading failure.

- It is best to assume the network is filled with poor implementations and malevolent actors, both of which will find every possible failure mode over time.

- It is best to assume every technology will be used to the limits of its technical capabilities, rather than assuming a particular protocol's scope of use will align (in any way) with the intent of the original designer(s). [RFC5218] defines a wildly sucessful protocol as one that "far exceeds its original goals, in terms of purpose (being used in scenarios far beyond the initial design), in terms of scale (being deployed on a scale much greater than originally envisaged), or both." Successful implementations attract more functionality, much like a few nodes in a scale free graph eventually become connectivity hubs.

- Protocols and implementations change over time. A corollary of the assumption that protocols will be used until they reach their technical limits is that protocols will change over time as they gain new functionality. [RFC5218] points out several problems with "wild success" in a protocol: undesirable side effects, performance problems, and becoming a high value attack target. Protocol and implementation design should take into account use cases that have not yet been thought of by building flexibility into protocols. Protocols should also remained focused on a narrow range of use cases; it is often wise to invent a new protocol than to extend a single protocol into a broad set of use cases.

- Protocols are sometimes replaced or updated to new versions in order to add new capabilities or features. Updating a protocol requires great care in providing for a transition mechanism between older and newer versions. draft-iab-protocol-transitions [I-D.iab-protocol-transitions] provides sound advice on protocol transition planning and mehanisms.

- Obscure, but legal, protocol features are often ignored or left unimplemented. Protocols must handle receiving unexpected information gracefully so they do not fail because of incomplete or partial implementations. Protocols should avoid specifying contradictory states, or features that will cause interoperability issues if multiple implementations choose to implement different feature sets.

- Monocultures are almost always bad. While multiple implementations can represent an interaction surface which increases complexity, particularly if a brad set of protocol capabilities and/or implementation features are used, using the same implementation at every point in a deployment results in a

monoculture. In a monoculture, a single event can trigger a defect in every router, causing a network failure. Monocultures must be carefully balanced against interaction surfaces; often this is best accomplished by using multiple implementations and minimal, widely implemented, and well understood protocol features.

A summary of the points above might be this: It is important to work within the bounds of what is actually implemented in any given protocol, and to leave corner cases for another day. It is often easy to assume "virtual oceans" are easier to boil than physical ones, or for an ocean to appear much smaller because it is being implemented in software. This is often deceptive. It is never helpful to boil the ocean whether in a design, an implementation, or a protocol.

2.2. Complexity

Complexity, as articulated by Mike O'Dell (see [RFC3439]), is "the primary mechanism which impede efficient scaling, and as a result is the primary driver of increases in both capital expenditures (CAPEX) and operational expenditures (OPEX)." At the same time, complexity cannot be "solved," but rather must be "managed." The simplest and most obvious solution to any problem is often easy to design, deploy, and manage. It's also often wrong and/or broken. As much as developers, designers, and operators might like to make things as simple as possible, hard problems require complex solutions. See Alderson and Doyle [COMPLEXHARD] for a discussion of the relationship between hard problems and complex solutions.

The following sections contain observations which apply to the management of complexity in both protocol and network design.

2.2.1. Elegance

Elegance should be the goal of protocol and network design. Rather than seeking out simple solutions because they are simple, seek out solutions that will solve the problem in the simplest way possible (and no simpler). Often this will require:

- Ensuring the goal is actually the goal. Many times the goal is taken from the operational realm into the protocol design realm before enough thought has been applied to ensure the correct problem is being addressed.

- Seeing the problem from different angles, trying to break the problem up in multiple ways; and trying, abandoning, and rebuilding ideas and implementations until a better way is found.

- Sometimes the complexity of the solution will overwhelm the use case; sometimes it is better to leave the apparent problem unsolved, or allow the community time and space to find a simpler solution.

2.2.2. Tradeoffs

There are always tradeoffs. For any protocol, network, or operational design decision, there will always be a tradeoff between at least two competing goals. If some problem appears to have a single solution without tradeoffs, this doesn't mean the tradeoffs don't exist. Rather, it means the tradeoffs haven't been discovered yet. In the area of protocol and network design, these tradeoffs often take the form of common "choose two or three" situations, such as "quick, cheap, high quality." In network and protocol design, the tradeoffs are often:

- The amount of state carried in the system and the speed at which it changes, or simply the state. The amount of state required to operate a system as it scales tends to be nonlinear. Some instances of this are described in [RFC3439] section 2.2.1, the Amplification Principle.

- The number of interaction surfaces between the components that make up the complete system, and the depth of those interaction surfaces. Some examples of surfaces are described in [RFC3439]section 2.2.2, the Coupling Principle. Layering is essentially a form of abstraction; all abstractions are subject to the law of leaky abstractions, [LEAKYABS] which states: "all nontrivial absractions leak."

- The desired optimization, including efficient use of network resources, optimal support for business objectives, and optimal support for a specific set of applications.

These three make up a "triangle problem." For instance, to increase the optimization of traffic flow through a network generally requires adding more state to the control plane, leading to problems in complexity due to amplification. To reduce amplification, the control plane (or perhaps the various functions the control plane serves) can be broken up into subsystems, or modules. Breaking the control plane up into subsystems, however, introduces interaction surfaces between the components, which is another form of complexity. [RFC7980] provides a good overview of network complexity; in particular, section 3 of that document provides some examples of complexity tradeoffs.

2.3. Layered Structure

The Internet data plane is organized around broad top and bottom layers, and much thinner middle layer. This is illustrated in the figure below.

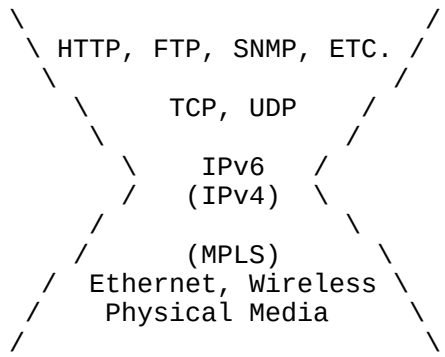


Figure 1

This layering emulates or mirrors many naturally occurring systems; it is a common strategy for managing complexity (see Meyer's presentation on complexity). [COMPLEXLAYER] The single protocol in the center, IPv6, serves to separate the complexity of the lower layers from the complexity of the upper layers. This center layer of the Internet ecosystem has traditionally been called the Network Layer, in reference to the Department of Defense (DoD) [DoD] and OSI models. [OSI] The Internet ecosystem includes two different protocols in this central location.

- IPv4, an older network protocol that, it is anticipated, will be replaced over time as the Internet ecosystem standardizes on IPv6

- IPv6, a newer network protocol that is being adopted

MPLS is often used as a "middle" subtransport layer, and at other times as "middle" sub data link layer; hence MPLS is difficult to classify within the strictly hierarchical model depicted here. These protocols are often treated as if they exist in strict hierarchical layers with a well defined and followed Application Programming Interface (API), data models, Remote Procedure Calls (RPCs), sockets, etc. The reality, however, is there are often solid reasons for violating these layers, creating interaction surfaces that are often deeper than intended or understood without some experience. Beyond this, such layering mechanisms act as information abstractions. It is well known that all such abstractions leak (see above on the law of leaky abstractions). Because of these intentional and

unintentional leakages of information, the interactions between protocols is often subtle.

2.4. Routers

A router connects to two or more logical interfaces and at least one physical interface. A router processes packets by:

- Receiving a packet through an interface
- Stripping the data link, physical header, or tunnel encapsulation off the packet
- Examining the packet for errors, and determining if this packet needs to be punted to another process on the router
- Looking up the destination in a local forwarding table
- Rewriting the data link and/or physical layer header
- Transmitting the packet out an interface

When consulting the forwarding table, the router searches for a match based on:

- The longest prefix containing the destination address (this is the most common matching element)
- A label, such as a flow label or MPLS label
- The source address or other header fields (not common)

The router then examines the information in the matching entry to determine the next hop, or rather the next logically connected device to forward the packet to. The next hop will either be another router, which will presumably carry the packet closer to the final destination, or it will be the destination host itself. The following figure provides a conceptual model of a router; not all routers actually have this set of tables and interactions, and some have many more moving parts. This model is simply used as a common reference to promote understanding.

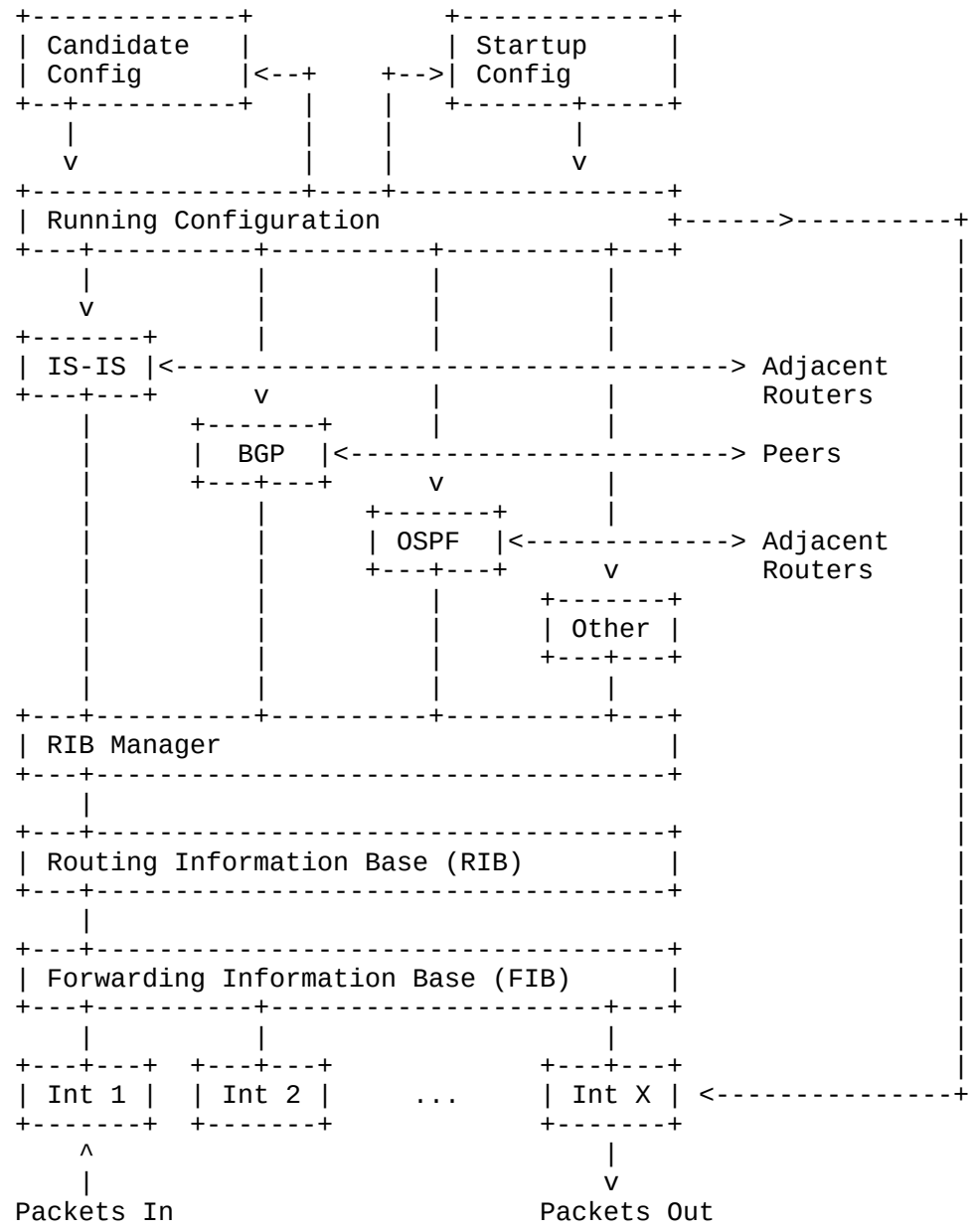

Candidate Config
Startup Config
Running Configuration
IS-IS
Adjacent Routers
BGP
Peers
OSPF
Adjacent Routers
Other
RIB Manager
Routing Information Base (RIB)
Forwarding Information Base (FIB)
Int 1
Int 2
...
Int X
Packets In
Packets Out

Figure 2

3. Requirements Related to Device Management and Security

Network engineering began in the era of Command Line Interfaces (CLIs), and has generally stayed with these CLIs even as the Graphical User Interface (GUI) has become the standard way of interacting with almost every other computing device. Direct human interaction with routers and middleboxes in large scale and complex environments, however, tends to result in an unacceptably low Mean Time Between Mistakes (MTBM), directly impacting the overall availability of the network. In reaction to this, operators have increased their reliance on automation, specifically targetting machine to machine intefaces, such as Remote Procesdure Calls (RPCs) and Application Programming Interface (API) solutions, to manage and configure routers and middleboxes. This section considers the various components of device management.

3.1. Programmable Device Access

Configuration primarily relates to the startup, candidate, and running configurations in the router model shown above. In order to deploy networks at scale, operators rely on automated management of router configuration. This effort has traditionally focused on Simple Network Management Protocol (SNMP) Management Information Base (MIBs). In the future, operators expect to move towards open source/ open standards YANG models.

Vendors and implementors should implement machine readable interfaces with overlays to support human interaction, rather than human readable interfaces with overlays to support machine to machine interaction. Emphasis should be placed on machine to machine interaction for day to day operations, rather than on human readable interfaces, which are largely used in the process of troubleshooting. Within the realm of machine to machine interfaces, emphasis should be placed on marshaling information in YANG models.

To support automated router configuration, IPv6 routers and routers SHOULD support YANG and SNMP configuration, including (but not limited to):

- Openconfig models [OPENCONF] related to the protocols configured on the device, interface state, and device state
- [RFC7223]: A YANG Data Model for Interface Management
- [RFC7224]: IANA Interface Type YANG Module
- [RFC7277]: A YANG Data Model for IP Management

- [RFC7317]: A YANG Data Model for System Management
- Simple Network Management Protocol (SNMP) MIBs as appropriate

3.2. Human Readable Device Access

To operate a network at scale, operators rely on the ability to access routers and middleboxes to troubleshoot and gather state manually through a number of different interfaces. These interfaces should provide current device configuration, current device state (such as interface state, packets drops, etc.), and current control plane contents (such as the RIB in the figure above). In other words, manual interfaces should provide information about the router (the whole device stack).

To support manual state gathering and troubleshooting, IPv6 routers and middleboxes SHOULD support:

- TELNET ([RFC0854]): TELNET SHOULD be disabled by default, but should be available for operational purposes as required or as configured by the operator
- SSH ([RFC4253]): SSH SHOULD be the default access for IPv6 capable routers

3.3. Zero Touch Provisioning

To operate a network at scale, operators rely on protocols and mechanisms that reduce provisioning time to a minimum. The preferred state is zero touch provisioning; plug a new router in and it just works without any manual configuration. The closer an operator can come to this ideal, the more MTBM and Operational Expenses (OPEX) can be reduced -- an important goals in the real world. IPv6 routers should support several standards, including, but not limited to:

- [I-D.ietf-dhc-rfc3315bis]: Dynamic Configuration Protocol for IPv6
- SLAAC ([RFC7217] and [RFC7527]): SLAAC SHOULD be enabled by default on all router interfaces

SLAAC SHOULD be able to be disabled by operators who prefer to use some other mechanism for address management and assignment.

3.4. Authentication, Authorization, and Accounting

(Need some text here)

3.5. Device Protection against Denial of Service Attacks

Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are unfortunately common in the Internet globally; these types of attacks cost network operators a great deal in opportunity and operational costs in prevention and responses. To provide for effective counters to DoS and DDoS attacks directly on routers:

- Manufacturers and system integrators should test and clearly report the packet/traffic load handling capabilities of devices with and without various encryption methods enabled
- Routers should be able to police traffic destined to the control plane based on the rate of traffic received, including the ability ot police individual flows, targeted services, etc., at individual rates as described in [RFC6192]
- Ideally, devices should be able to statefully filter traffic destined to the control plane

There are other useful techniques for dealing with DDoS attacks at the network level, including: transferring sessions to a new address and abandoning the address under attack, using BGP communities to spread the attack over multiple ingress ports and "consume" it, and requiring mutual authentication before allocating larger resource pools to a connection. These techniques are not "device level," and hence are not considered further here.

4. Requirements Related to Telemetry

Telemetry relates to information devices push to systems used to monitor and track the state of the network. This applies to individual devices as well as the network as a system. Two major challenges face operators in the area of telemetry:

- Information that is laid out primarily for human, rather than machine, consumption. While human consumption of telemetry is important in some situations, this information should be supplied in a form that focuses on machine readability with an overlay or interpretor that allows human consumption.
- Software systems that require information to be queried (or polled or even pushed) on a per-item basis. This form of organization can produce a lot of information, and a lot of individual packets, very quickly, overwhelming monitoring systems and consuming a large amount of available network resources. Instead, telemetry should be focused on bulk collection.

There are three broad categories of telemetry: device state and traceability, topology state and traceability, and flow traceability. These three roughly correspond to the management plane, the control plane, and the forwarding plane of the network. Each of the sections below considers one of these three telemetry types.

4.1. Device State and Traceablity

Ideally, the entire network could be monitored using a single modeling language to ease implementation of telemetry systems and increase the pace at which new software can be deployed in production environments. In real deployments, it is often impossible to reach this ideal; however, reducing the languages and methods used, while focusing on machine readibility, can greatly ease the deployment and management of a large scale network. Specifically, IPv6 routers SHOULD support:

- [RFC6241]: NETCONF/RESTCONF transporting telemetry formatted according to YANG (see above)
- [RFC5424]: Syslog
- gRPC based telemetry interfaces [GRPC]
- SNMP as appropriate

Syslog and SNMP access for telemetry should be considered "legacy," and should not be the focus of new telemetry access development efforts.

4.2. Topology State and Traceability

IPv6 routers are part of a system of devices that, combined, make up the entire network. Viewing the network as a system is often crucial for operational purposes. For instance, being able to understand changes in the topology and utlization of a network can lead to insights about traffic flow and network growth that lead to a greater understanding of how the network is operating, where problems are developing, and how to improve the network’s performance. To support systemic monitoring of the network topology, IPv6 devices SHOULD support at least:

- [RFC5424]: North-Bound Distribution of Link-State and Traffic Engineering (TE) Information using BGP
- [I-D.ietf-i2rs-yang-l2-network-topology]: An I2RS model for layer 2 topologies

- o [I-D.ietf-i2rs-yang-l3-topology]: An I2RS model for layer 3 topologies

- o [I-D.ietf-i2rs-yang-network-topo]: A Data Model for Network Topologies

4.3. Flow Traceability

(To be added)

5. Requirements Related to IPv6 Forwarding and Addressing

There are a number of capabilities that a device SHOULD have to be deployed into an IPv6 network, and several forwarding plane considerations operators and vendors need to bear in mind. The sections below explain these considerations.

5.1. The IPv6 Address is not a Host Identifier

The IPv6 address is commonly treated as a host identifier; it is not. Rather, it is an interface identifier that describes the topological point where a particular host connects to the Internet. Specifically:

- o The IPv6 address will change when a device changes where it connects to the network.

- o A single host can have multiple addresses. For instance, a host may have one address per interface, or multiple addresses assigned through different mechanisms, or through multiple connection points.

- o A single IPv6 address may represent many hosts, as in the case of a group of hosts reachable through multicast address, or a set of services reachable through an anycast address.

Because the host address may change at any time, it is generally harmful to embed IPv6 addresses inside upper layer headers to identify a particular host.

5.2. Router Handling of IPv6 Addresses

Internet Routing Registries may allocate a network operator a wide range of prefix lengths (see [RFC6177] for further information). Within this allocation, network operators will often suballocate address space along nibble boundaries (/48, /52, /56, /60, and /64) for ease of configuration and management. Several common practices are:

- Each multiaccess interface is allocated a /64

- Point-to-point links are allocated a /64, but SHOULD be addressed with a longer prefix length to prevent certain kinds of denial of service attacks ([RFC3627] originally mandated 64 bit prefix lengths on point-to-point links; [RFC6164] explains possible security issues with assigning a 64 bit prefix length to a point-to-point, and recommends a /127 instead)

- Although aggregation may only performed to the nibble boundaries noted above, variances are possible

- Loopback addresses are assigned a /128

Given these common practices, routers designed to run IPv6 SHOULD support the following addressing conventions:

- The default prefix length on any interface other than a loopback SHOULD be a /64

- Configuring a prefix length longer than a /64 on any multi-access interface should require additional configuration steps to prevent manual configuration errors

- Routers SHOULD NOT assume IPv6 prefix lengths only on nibble boundaries

- Routers SHOULD support any prefix length shorter or greater than /64

- Loopback interfaces SHOULD default to a /128 prefix length unless some additional configuration is undertaken to override this default setting

5.3. Maximum Transmission Unit and Jumbo Frames

The long history of the Maximum Transmission Unit (MTU) in networks is not a happy one. Specific problems with MTU sizing include:

- Many different default sizes on different media types, from very small (576 bytes on X.25) to very large (17914 bytes on 16Mbps Token Ring)

- Many different ways to calcualte the MTU on any given link; for instance a 9000 byte MTU can be calculated as 8184 bytes on one operating system, 8972 on another, and 9000 on a third

- The increasing use of tunnel encapsulations in the network; for instance MPLS over GRE over IP over...
- The wide variety of default MTUs across many different end hosts and operating systems
- The general ineffectiveness of path MTU discovery to operate correctly in the face of packet filters and rate limiters (see the section on ICMP filtering below)
- Lower speed links at the network edge which require a lot of time to serialize a packet with a large MTU
- Increased jitter caused by the disparity between large and small packet size across a lower bandwidth links

The final point requires some further elucidation. The time required to serialize various packets at various speeds are:

- 64 byte packet onto a 10Mb/s link: .5ms
- 1500 byte packet onto a 10Mb/s link: 1.2ms
- 9000 byte packet onto a 10Mb/s link: 7.2ms
- 64 byte packet onto a 100Mb/s link: .05ms
- 1500 byte packet onto a 100Mb/s link: .12ms
- 9000 byte packet onto a 100Mb/s link: .72ms

A 64 byte packet trapped behind a single 1500 byte packet on a 10Mb/s link suffers 1.2ms of serialization delay. Each additional 1500 byte packet added to the queue in front of the 64 byte packet adds and addtional 1.2ms of delay. In contrast, a 64 byte packet trapped behind a single 9000 byte packet on a 10Mb/s link suffers 7.7ms of serialization delay. Each additional 9000 byte packet added to the queue adds an additional 7.2ms of serialization delay. The practical result is that larger MTU sizes on lower speed links can add a significant amount of delay and jitter into a flow. On the other hand, increasing the MTU on higher speed links appears to add megligable additional delay and jitter.

The result is that it costs less in terms of overall systemic performance to use higher MTUs on higher speed links than on lower speed links. Based on this, increasing the MTU across any particular link may not increase overall end-to-end performance, but can greatly enhance the performance of local applications (such as a local BGP

peering session, or a large/long standing elephant flow used to transfer data across a local fabric), while also providing room for tunnel encapsulations to be added with less impact on lower MTU end systems.

The general rule of thumb is to assume the largest size MTU should be used on higher speed transit only links in order to support a wide array of available link sizes, default MTUs, and tunnel encapsulations. Routers designed for a network or data center core SHOULD support at least 9000 byte MTUs on all interfaces. MTU detection mechanisms, such as IS-IS hello padding, described in [RFC7922], SHOULD be enabled to ensure correct point-to-point MTU configuration. Devices SHOULD also support:

- [RFC1191]: Path MTU Discovery
- [RFC1981]: Path MTU Discovery for IP version 6
- [RFC4821]: Packetization Layer Path MTU Discovery

5.4. ICMP Considerations

Internet Control Message Protocool (ICMP) is described in [RFC0792] and [RFC4443]. ICMP is often used to perform a traceroute through a network (normally by using a TTL expired ICMP message), for Path MTU discovery, and, in IPv6, for autoconfiguration and neighbor discovery. ICMP is often blocked by middleboxes of various kinds and/or ICMP filters configured on the ingress edge of a provider network, most often to prevent the discovery of reachable hosts and network topology. Routers implementing IPv6:

- SHOULD NOT filter ICMP unreachables by default, as this has negative impacts on many aspects of IPv6 operation, particularly path MTU
- SHOULD filter ICMP echo and echo response by default, to prevent the discovery of reachable hosts and topology.
- SHOULD rate limit the generation of ICMP messages relative to the ability of the device to generate packets and to block the use of ICMP packets being used as part of a distributed denial of service attack
- SHOULD implement the filtering suggestions in [I-D.gont-opsec-icmp-ingress-filtering]

There are implications for path MTU discovery and other useful mechanisms in filtering and rate limiting ICMP. The tradeoff here is

between allowing unlimited ICMP, which would allow path MTU detection to work, or limiting ICMP in a way that prevents negative side effects for individual devices, and hence the operational capabilities of the network as a whole. Operators rightly limit ICMP to reduce the attack surface against their network, as well as the opportinity for "perfect storm" events that inadvertently reduce the capability of routers and middleboxes. Hence ICMP can be treated as "quasi-reliable" in many situations; existence of an ICMP message can prove, for instance, that a particular host is unreachable. The non-existence of an ICMP message, however, does not prove a particular host exists or does not.

5.5. Machine Access to the Forwarding Table

In order to support treating the "network as a whole" as a single programmable system, it is important for each router have the ability to directly program forwarding information. This programmatic interface allows controllers, which are programmed to support specific business logic and applications, to modify and filter traffic flows without interfering with the distributed control plane. While there are several programmatic interfaces available, this document suggests that the I2RS interface to the RIB be supported in all IPv6 routers. Specifically, these drafts should be supported to enable network programmability:

- [I-D.ietf-i2rs-fb-rib-data-model]: Filter-Based RIB Data Model
- [I-D.ietf-i2rs-fb-rib-info-model]: Filter-Based RIB Information Model
- [I-D.ietf-i2rs-rib-data-model]: A YANG Data Model for Routing Information Base (RIB)
- [RFC7922]: I2RS Traceability

5.6. Processing IPv6 Extension Headers

(To be added)

5.7. IPv6 Only Operation

While the transition to IPv6 only networks may take years (or perhaps decades), a number of operators are moving to deploy IPv6 on internal networks supporting transport and data center fabric applications more quickly. Routers and middleboxes that support IPv6 SHOULD support IPv6 only operation, including:

- o Link Local addressing SHOULD be configurable and useable as the primary address on all interfaces on a device.

- o IPv4 and/or MPLS SHOULD NOT be required for proper device operation. For instance, an IPv4 address should not be required to determine the router ID for any protocol. See [RFC6540] section 2.

- o Any control plane protocol implementations SHOULD support the recommendations in [RFC7404] for operation using link local addresses only.

6. Future Considerations

(To be added)

6.1. Segment Routing

(To be added)

7. Security Considerations

This document addresses several ways in which devices designed to support IPv6 forwarding. Some of the recommendations here are designed to increase device security; for instance, see the section on device access. Others may intersect with security, but are not specifically targeted at security, such as running IPv6 link local only on links. These are not discussed further here, as they improve the security stance of the network. Other areas discussed in this draft are more nuanced. This section gathers the intersection between operational concerns and security concerns into one place.

ICMP security is already considered in the section on ICMP; it will not be considered further here. Link local only addressing wil increase security by removing transit only links within the network as a reachable destination.

7.1. Robustness and Security

Robustness, particularly in the area of error handling, largely improves security if designed and implemented correctly. Many attacks take advantage of mistakes in implementations and variations in protocols. In particular, any feature that is unevenly implemented among a number of implementations often offers an attack surface. Hence, reducing protocol complexity helps reduce the breadth of attack surfaces.

Another point to consider at the intersection of robustness and security is the issue of monocultures. Monocultures are in and of themselves a potential attack surface, in that finding a single failure mode can be exploited to take an entire network (or operator) down. On the other hand, reducing the number of implementations for any particular protocol will decrease the set of "random" features deployed in the network. These two goals will often be opposed to one another. Network designers and operators need to consider these two sides of this tradeoff, and make an intelligent decision about how much diversity to implement versus how to control the attack surface represented by deploying a wide array of implementations.

7.2. Programmable Device Access and Security

Programmable interfaces, including programmable configuration, telemetry, and machine interface to the routing table, introduce a large attack surface; operators should be careful to ensure this attack surface is properly secured. Specifically:

- Prevent external access to any administrative access points used for device programmability
- Use AAA systems to ensure only valid devices and/or users access devices
- Rate limit the change rate and protect management interfaces from DoS and DDoS attacks

Such interfaces should be treated no differently than SSH, SFTP, and other interfaces available to manage routers and middleboxes.

7.3. Zero Touch Provisioning and Security

Zero touch provisioning opens a new attack surface; insider attackers can simply install a new device, and assume it will be autoconfigured into the network. A "simple" solution would be to install door locks, but this will likely not be enough; defenses need to be layered to be effective. It is recommended that devices installed in the network need to contain a hardware or software identification system that allows the operator to identify devices that are installed in the network.

8. Conclusion

The deployment of IPv6 throughout the Internet marks a point in time where it is good to review the overall Internet architecture, and assess the impact on operations of these changes. This document provides an overview of a lot of these changes and lessons learned,

as well as providing pointers to many of the relevant documents to understand each topic more deeply.

9. References

9.1. 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>.

9.2. Informative References

[COMPLEXHARD]
Alderson, D. and J. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures", 2010, <http://ieeexplore.ieee.org/abstract/document/5477188/?reload=true>.

[COMPLEXLAYER]
Meyer, D., "Macro Trends, Architecture, and the Hidden Nature of Complexity", 2010, <http://www.slideshare.net/dmm613/macro-trends-complexityandsdn-32951199>.

[DoD] Wikipedia, "The Internet Protocol Suite", 2016, <https://en.wikipedia.org/wiki/Internet_protocol_suite>.

[GRPC] gRPC, "gRPC", 2016, <http://www.grpc.io>.

[I-D.gont-opsec-icmp-ingress-filtering]
Gont, F., Hunter, R., Massar, J., and S. LIU, "Defeating Attacks which employ Forged ICMP/ICMPv6 Error Messages", draft-gont-opsec-icmp-ingress-filtering-02 (work in progress), March 2016.

[I-D.iab-protocol-transitions]
Thaler, D., "Out With the Old and In With the New: Planning for Protocol Transitions", draft-iab-protocol-transitions-05 (work in progress), January 2017.

[I-D.ietf-dhc-rfc3315bis]
Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) bis", draft-ietf-dhc-rfc3315bis-06 (work in progress), October 2016.

[I-D.ietf-i2rs-fb-rib-data-model]
Hares, S., Kini, S., Dunbar, L., Krishnan, R., Bogdanovic, D., and R. White, "Filter-Based RIB Data Model", draft-ietf-i2rs-fb-rib-data-model-00 (work in progress), June 2016.

[I-D.ietf-i2rs-fb-rib-info-model]
Kini, S., Hares, S., Dunbar, L., Ghanwani, A., Krishnan, R., Bogdanovic, D., and R. White, "Filter-Based RIB Information Model", draft-ietf-i2rs-fb-rib-info-model-00 (work in progress), June 2016.

[I-D.ietf-i2rs-rib-data-model]
Wang, L., Ananthakrishnan, H., Chen, M., amit.dass@ericsson.com, a., Kini, S., and N. Bahadur, "A YANG Data Model for Routing Information Base (RIB)", draft-ietf-i2rs-rib-data-model-07 (work in progress), January 2017.

[I-D.ietf-i2rs-yang-l2-network-topology]
Dong, J. and X. Wei, "A YANG Data Model for Layer-2 Network Topologies", draft-ietf-i2rs-yang-l2-network-topology-03 (work in progress), July 2016.

[I-D.ietf-i2rs-yang-l3-topology]
Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", draft-ietf-i2rs-yang-l3-topology-08 (work in progress), January 2017.

[I-D.ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-11 (work in progress), February 2017.

[I-D.ietf-netconf-restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-18 (work in progress), October 2016.

[LEAKYABS] Spolsky, J., "The Law of Leaky Abstractions", 2002, <https://www.joelonsoftware.com/2002/11/11/the-law-of-leaky-abstractions/>.

[OPENCONF] OpenConfig, "Openconfig release YANG models", 2016, <https://github.com/openconfig/public/tree/master/release>.

[OSI] Wikipedia, "OSI Model", 2016, <https://en.wikipedia.org/wiki/OSI_model>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <http://www.rfc-editor.org/info/rfc792>.

[RFC0854] Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, DOI 10.17487/RFC0854, May 1983, <http://www.rfc-editor.org/info/rfc854>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <http://www.rfc-editor.org/info/rfc1191>.

[RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <http://www.rfc-editor.org/info/rfc1812>.

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

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, DOI 10.17487/RFC1981, August 1996, <http://www.rfc-editor.org/info/rfc1981>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <http://www.rfc-editor.org/info/rfc2629>.

[RFC3439] Bush, R. and D. Meyer, "Some Internet Architectural Guidelines and Philosophy", RFC 3439, DOI 10.17487/RFC3439, December 2002, <http://www.rfc-editor.org/info/rfc3439>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <http://www.rfc-editor.org/info/rfc3552>.

[RFC3627] Savola, P., "Use of /127 Prefix Length Between Routers Considered Harmful", RFC 3627, DOI 10.17487/RFC3627, September 2003, <http://www.rfc-editor.org/info/rfc3627>.

[RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <http://www.rfc-editor.org/info/rfc4253>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, DOI 10.17487/RFC4443, March 2006, <http://www.rfc-editor.org/info/rfc4443>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <http://www.rfc-editor.org/info/rfc4821>.

[RFC5218] Thaler, D. and B. Aboba, "What Makes For a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <http://www.rfc-editor.org/info/rfc5218>.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <http://www.rfc-editor.org/info/rfc5424>.

[RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", RFC 6164, DOI 10.17487/RFC6164, April 2011, <http://www.rfc-editor.org/info/rfc6164>.

[RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", BCP 157, RFC 6177, DOI 10.17487/RFC6177, March 2011, <http://www.rfc-editor.org/info/rfc6177>.

[RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <http://www.rfc-editor.org/info/rfc6192>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6540] George, W., Donley, C., Liljenstolpe, C., and L. Howard, "IPv6 Support Required for All IP-Capable Nodes", BCP 177, RFC 6540, DOI 10.17487/RFC6540, April 2012, <http://www.rfc-editor.org/info/rfc6540>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <http://www.rfc-editor.org/info/rfc7217>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <http://www.rfc-editor.org/info/rfc7223>.

[RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", RFC 7224, DOI 10.17487/RFC7224, May 2014, <http://www.rfc-editor.org/info/rfc7224>.

[RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", RFC 7277, DOI 10.17487/RFC7277, June 2014, <http://www.rfc-editor.org/info/rfc7277>.

[RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", RFC 7317, DOI 10.17487/RFC7317, August 2014, <http://www.rfc-editor.org/info/rfc7317>.

[RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <http://www.rfc-editor.org/info/rfc7404>.

[RFC7527] Asati, R., Singh, H., Beebee, W., Pignataro, C., Dart, E., and W. George, "Enhanced Duplicate Address Detection", RFC 7527, DOI 10.17487/RFC7527, April 2015, <http://www.rfc-editor.org/info/rfc7527>.

[RFC7663] Trammell, B., Ed. and M. Kuehlewind, Ed., "Report from the IAB Workshop on Stack Evolution in a Middlebox Internet (SEMI)", RFC 7663, DOI 10.17487/RFC7663, October 2015, <http://www.rfc-editor.org/info/rfc7663>.

[RFC7922] Clarke, J., Salgueiro, G., and C. Pignataro, "Interface to the Routing System (I2RS) Traceability: Framework and Information Model", RFC 7922, DOI 10.17487/RFC7922, June 2016, <http://www.rfc-editor.org/info/rfc7922>.

[RFC7980] Behringer, M., Retana, A., White, R., and G. Huston, "A Framework for Defining Network Complexity", RFC 7980, DOI 10.17487/RFC7980, October 2016, <http://www.rfc-editor.org/info/rfc7980>.

Authors' Addresses

Zaid Ali Kahn, Editor
LinkedIn
xxx
xxx, CA xxx
USA

Email: zaid@linkedin.com

John Brzozowski, Editor
Comcast
xxx
xxx, xxx xxx
USA

Email: John_Brzozowski@comcast.com

Russ White, Editor
LinkedIn
Oak Island, NC 28465
USA

Email: russ@riw.us

intarea
Internet-Draft
Intended status: Informational
Expires: January 29, 2018

P. Pfister, Ed.
Cisco
D. Schinazi
T. Pauly
Apple
E. Vyncke
Cisco
B. Bruneau
Ecole Polytechnique
July 28, 2017

Discovering Provisioning Domain Names and Data draft-bruneau-intarea-provisioning-domains-02

Abstract

An increasing number of hosts and networks are connected to the Internet through multiple interfaces, some of which may provide multiple ways to access the internet by the mean of multiple IPv6 prefix configurations.

This document describes a way for hosts to retrieve additional information about their network access characteristics. The set of configuration items required to access the Internet is called a Provisioning Domain (PvD) and is identified by a Fully Qualified Domain Name (FQDN). This identifier, retrieved using a new Router Advertisement (RA) option, is associated with the set of information included within the RA and may later be used to retrieve additional information associated with the PvD by the mean of an HTTP request.

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 January 29, 2018.

Table of Contents

1. Introduction

It has become very common in modern networks that hosts have internet or more specific network access through different networking interfaces, tunnels, or next-hop routers. The concept of Provisioning Domain (PvD) was defined in [RFC7556] as a set of network configuration information which can be used by hosts in order to access the network.

This specification provides a way to identify explicit PvDs with Fully Qualified Domain Names called PvD IDs, which are included in a new Router Advertisement [RFC4861] option. This new option, when present, is used to associate the correlated set of configuration information with the identified PvD. It is worth noting that multiple PvDs with different PvD IDs could be provisioned on any host interface, as well as noting that the same PvD ID could be used on different interfaces in order to inform the host that both PvDs, on different interfaces, ultimately provide identical services.

This document also introduces a way for hosts to retrieve additional information related to a specific PvD by the mean of an HTTP-over-TLS query using an URI derived from the PvD ID. The retrieved JSON object contains additional network information that would typically be considered unfit, or too large, to be directly included in the Router Advertisements. This information can be used by the networking stack, the applications, or even be partially displayed to the users (e.g., by displaying a localized network service name).

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 [RFC2119].

In addition, this document uses the following terminology:

PvD: A Provisioning Domain, a set of network configuration information; for more information, see [RFC7556].

PvD ID: A Fully Qualified Domain Name (FQDN) used to identify a PvD.

Explicit PvD: A PvD uniquely identified with a PvD ID. for more information, see [RFC7556].

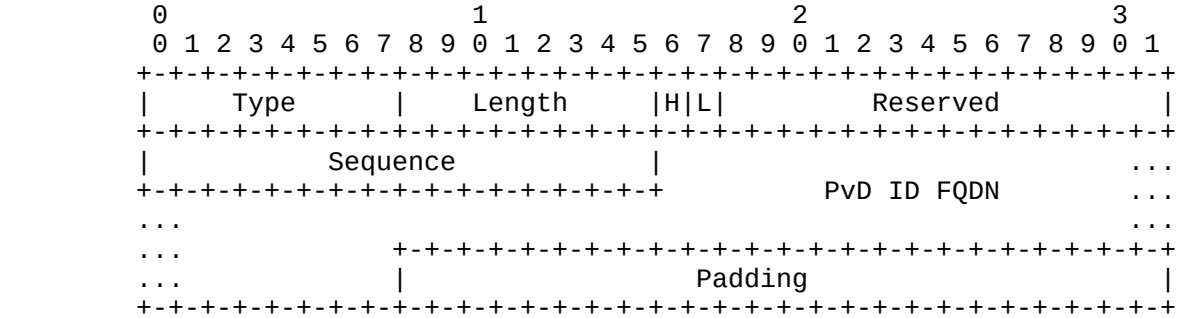
Implicit PvD: A PvD associated with a set of configuration information that, in the absence of a PvD ID, is associated with the advertising router.

3. Provisioning Domain Identification using Router Advertisements

Each provisioning domain is identified by a PvD ID. The PvD ID is a Fully Qualified Domain Name (FQDN) which MUST belong to the network operator in order to avoid ambiguity. The same PvD ID MAY be used in several access networks when the set of configuration information is identical (e.g. in all home networks subscribed to the same service).

3.1. PvD ID Option for Router Advertisements

This document introduces a new Router Advertisement (RA) option called the PvD ID Router Advertisement Option, used to convey the FQDN identifying a given PvD.



PvD ID Router Advertisements Option format

Type : (8 bits) To be defined by IANA.

Length : (8 bits) The length of the option (including the Type and Length fields) in units of 8 octets.

H-flag : (1 bit) Whether some PvD Additional Information is made available through HTTP over TLS, as described in Section 4.

L-flag : (1 bit) Whether the router is also providing IPv4 access using DHCPv4 (see Section 3.3.2).

Reserved : (14 bits) Reserved for later use. It MUST be set to zero by the sender and ignored by the receiver.

Sequence : (16 bits) Sequence number for the PvD Additional Information, as described in Section 4.

PvD ID FQDN : An ASCII string representation of the FQDN used as PvD ID. The string ends at the first byte set to zero, or the end of the option, whichever comes first.

Padding : Zero or more padding octets such as to set the option length (Type and Length fields included) to eight times the value of the Length field. It MUST be set to zero by the sender and ignored by the receiver.

Routers MUST NOT include more than one PvD ID Router Advertisement Option in each RA. In case multiple PvD ID options are found in a given RA, hosts MUST ignore all but the first PvD ID option.

Note: The existence and/or size of the sequence number is subject to discussion. The validity of a PvD Additional Information object is included in the object itself, but this only allows for 'pull based' updates, whereas the RA options usually provide 'push based' updates.

3.2. Router Behavior

A router MAY insert at most one PvD ID Option in its RAs. The included PvD ID is associated with all the other options included in the same RA (e.g., Prefix Information [RFC4861], Recursive DNS Server [RFC6106], Routing Information [RFC4191] options).

In order to provide multiple independent PvDs, a router MUST send multiple RAs using different source link-local addresses (LLA) (as proposed in [I-D.bowbakova-rtgwg-enterprise-pa-multihoming]), each of which MAY include a PvD ID option. In such cases, routers MAY originate the different RAs using the same datalink layer address.

If the router is actually a VRRP instance [RFC5798], then the procedure is identical except that the virtual datalink layer address is used as well as the virtual IPv6 addresses.

3.3. Host Behavior

RAs are used to configure IPv6 hosts. When a host receives an RA message including a PvD ID Option, it MUST associate all the configuration objects which are updated by the received RA (e.g., Prefix Information [RFC4861], Recursive DNS Server [RFC6106], Routing Information [RFC4191] options) with the PvD identified by the PvD ID Option, even if some objects are already ssociated with a different explicit or implicit PvD.

If the received RA does not include a PvD ID Option, the host MUST associate the configuration objects which are updated by the received RA with an implicit PvD, even if some objects were already associated

with a different explicit or implicit PvD. This implicit PvD is identified by the link-local address of the router sending the RA and the interface on which the RA was received.

This document does not update the way Router Advertisement options are processed. But in addition to the option processing defined in other documents, hosts implementing this specification MUST associate each created or updated object (e.g. address, default route, more specific route, DNS server list) with the PvD associated with the received RA.

Note: There is a discussion whether there can be multiple implicit PvDs on a single interface (i.e. whether the router link-local address should be used to identify the implicit PvDs).

While resolving names, executing the default address selection algorithm [RFC6724] or executing the default router selection algorithm ([RFC2461], [RFC4191] and [RFC8028]), hosts MAY consider only the configuration associated with an arbitrary set of PvDs.

For example, a host MAY associate a given process with a specific PvD, or a specific set of PvDs, while associating another process with another PvD. A PvD-aware application might also be able to select, on a per-connection basis, which PvDs should be used for a given connection. In particular, constrained devices such as small battery operated devices (e.g. IoT), or devices with limited CPU or memory resources may purposefully use a single PvD while ignoring some received RAs containing different PvD IDs.

The way an application expresses its desire to use a given PvD, or a set of PvDs, or the way this selection is enforced, is out of the scope of this document. Useful insights about these considerations can be found in [I-D.kline-mif-mpvd-api-reqs].

3.3.1. DHCPv6 configuration association

When a host retrieves configuration elements using DHCPv6, they MUST be associated with the explicit or implicit PvD of the RA received on the same interface, using the same link-local address, and with the O-flag set [RFC4861]. If no such PvD is found, or whenever multiple different PvDs are found, the host behavior is unspecified.

This process requires hosts to keep track of received RAs, associated PvD IDs, and routers link-local addresses.

3.3.2. DHCPv4 configuration association

When a host retrieves configuration elements from DHCPv4, they MUST be associated with the explicit PvD received on the same interface, whose PVD ID Options L-flag is set and, in the case of a non point-to-point link, using the same link-layer address. If no such PvD is found, or whenever multiple different PvDs are found, the configuration elements coming from DHCPv4 MUST be associated with an IPv4-only implicit PvD identified by the interface on which the DHCPv4 transaction happened.

3.3.3. Interconnection Sharing by the Host

The situation when a host becomes also a router by acting as a router or ND proxy on a different interface (such as WiFi) to share the connectivity of another interface (such as cellular), also known as "tethering" is TBD but it is expected that the one or several PvD associated to the shared interface will also be advertised to the clients.

4. Provisioning Domain Additional Information

Once a new PvD ID is discovered, it may be used to retrieve additional information about the characteristics of the provided connectivity. This set of information is called PvD Additional Information, and is encoded as a JSON object [RFC7159].

The purpose of this additional set of information is to securely provide additional information to hosts about the connectivity that is provided using a given interface and source address pair. It typically includes data that would be considered too large, or not critical enough, to be provided within an RA option. The information contained in this object MAY be used by the operating system, network libraries, applications, or users, in order to decide which set of PvDs should be used for which connection, as described in Section 3.3.

4.1. Retrieving the PvD Additional Information

When the H-flag of the PvD ID Option is set, hosts MAY attempt to retrieve the PvD Additional Information associated with a given PvD by performing an HTTP over TLS [RFC2818] GET query to https://<PvD-ID >/.well-known/pvd [RFC5785]. Inversely, hosts MUST NOT do so whenever the H-flag is not set.

Note: Should the PvD AI retrieval be a MAY or a SHOULD ? Could the object contain critical data, or should it only contain informational data ?

Note that the DNS name resolution of <PvD-ID> as well as the actual query MUST be performed using the PvD associated with the PvD ID. In other words, the name resolution, source address selection, as well as the next-hop router selection MUST be performed while using exclusively the set of configuration information attached with the PvD, as defined in Section 3.3. In some cases, it may therefore be necessary to wait for an address to be available for use (e.g., once the Duplicate Address Detection or DHCPv6 processes are complete) before initiating the HTTP over TLS query.

If the HTTP status of the answer is greater than or equal to 400 the host MUST abandon and consider that there is no additional PvD information. If the HTTP status of the answer is between 300 included and 399 included it MUST follow the redirection(s). If the HTTP status of the answer is between 200 included and 299 included the host MAY get a file containing a single JSON object. When a JSON object could not be retrieved, an error message SHOULD be logged and/or displayed in a rate-limited fashion.

After retrieval of the PvD Additional Information, hosts MUST watch the PvD ID Sequence field for change. In case a different value than the one in the RA Sequence field is observed, or whenever the validity time included in the PVD Additional Information JSON object is expired, hosts MUST either perform a new query and retrieve a new version of the object, or deprecate the object and stop using it.

Hosts retrieving a new PvD Additional Information object MUST check for the presence and validity of the mandatory fields Section 4.3. A retrieved object including an outdated expiration time or missing a mandatory element MUST be ignored. In order to avoid traffic spikes toward the server hosting the PvD Additional Information when an object expires, a host which last retrieved an object at a time A, including a validity time B, SHOULD renew the object at a uniformly random time in the interval [(B-A)/2,A].

The PvD Additional Information object includes a set of IPv6 prefixes which MUST be checked against all the Prefix Information Options advertised in the Router Advertisement. If any of the prefixes included in the Prefix Information Options is not included in at least one of the listed prefixes, the PvD associated with the tested prefix MUST be considered unsafe and MUST NOT be used. While this does not prevent a malicious network provider, it does complicate some attack scenarios, and may help detecting misconfiguration.

The server providing the JSON files SHOULD also check whether the client address is part of the prefixes listed into the additional information and SHOULD return a 403 response code if there is no

match. The server MAY also use the client address to select the right JSON object to be returned.

4.2. Providing the PvD Additional Information

Whenever the H-flag is set in the PvD RA Option, a valid PvD Additional Information object MUST be made available to all hosts receiving the RA. In particular, when a captive portal is present, hosts MUST still be allowed to access the object, even before logging into the captive portal.

Routers MAY increment the PVD ID Sequence number in order to inform host that a new PvD Additional Information object is available and should be retrieved.

4.3. PvD Additional Information Format

The PvD Additional Information is a JSON object.

The following array presents the mandatory keys which MUST be included in the object:

JSON key	Description	Type	Example
name	Human-readable service name	UTF-8 string	"Awesome Wifi"
expires	Date after which this object is not valid	[RFC3339]	"2017-07-23T06:00:00Z"
prefixes	Array of IPv6 prefixes valid for this PVD	Array of strings	["2001:db8:1::/48", "2001:db8:4::/48"]

A retrieved object which does not include a valid string associated with the "name" key at the root of the object, or a valid date associated with the "expiration" key, also at the root of the object, MUST be ignored. In such cases, an error message SHOULD be logged and/or displayed in a rate-limited fashion.

The following table presents some optional keys which MAY be included in the object.

JSON key	Description	Type	Example
localizedName	Localized user-visible service name, language can be selected based on the HTTP Accept-Language header in the request.	UTF-8 string	"Wifi Genial"
noInternet	No Internet, set when the PvD only provides restricted access to a set of services.	boolean	true
characteristics	Connectivity characteristics	JSON object	See Section 4.3.1
metered	metered, when the access volume is limited.	boolean	false

It is worth noting that the JSON format allows for extensions. Whenever an unknown key is encountered, it MUST be ignored along with its associated elements.

4.3.1. Connectivity Characteristics Information

The following set of keys can be used to signal certain characteristics of the connection towards the PvD.

They should reflect characteristics of the overall access technology which is not limited to the link the host is connected to, but rather a combination of the link technology, CPE upstream connectivity, and further quality of service considerations.

JSON key	Description	Type	Example
maxThroughput	Maximum achievable throughput	object({down(int), up(int)}) in kb/s	{"down": 10000, "up": 5000}
minLatency	Minimum achievable latency	object({down(int), up(int)}) in ms	{"down": 10, "up": 20}
rl	Maximum achievable reliability	object({down(int), up(int)}) in losses every 1000 packets	{"down": 0.1, "up": 1}

4.3.2. Private Extensions

JSON keys starting with "x-" are reserved for private use and can be utilized to provide information that is specific to vendor, user or enterprise. It is RECOMMENDED to use one of the patterns "x-FQDN-KEY" or "x-PEN-KEY" where FQDN is a fully qualified domain name or PEN is a private enterprise number [PEN] under control of the author of the extension to avoid collisions.

4.3.3. Example

Here are two examples based on the keys defined in this section.

```
{
  "name": "Foo Wireless",
  "localizedName": "Foo-France Wifi",
  "expires": "2017-07-23T06:00:00Z",
  "prefixes" : ["2001:db8:1::/48", "2001:db8:4::/48"],
  "characteristics": {
   "maxThroughput": { "down":200000, "up": 50000 },
   "minLatency": { "down": 0.1, "up": 1 }
  }
}

{
  "name": "Bar 4G",
  "localizedName": "Bar US 4G",
  "expires": "2017-07-23T06:00:00Z",
  "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
  "metered": true,
  "characteristics": {
    "maxThroughput": { "down":80000, "up": 20000 }
  }
}
```

5. Security Considerations

Although some solutions such as IPsec or SEND [RFC3971] can be used in order to secure the IPv6 Neighbor Discovery Protocol, actual deployments largely rely on link layer or physical layer security mechanisms (e.g. 802.1x [IEEE8021X]) in conjunction with RA Guard [RFC6105].

This specification does not improve the Neighbor Discovery Protocol security model, but extends the purely link-local configuration retrieval mechanisms with HTTP-over-TLS communications.

During the exchange, the server authenticity is verified by the mean of a certificate, validated based on the FQDN found in the Router Advertisement (e.g. using a list of pre-installed CA certificates, or DNSSec [RFC4035] with DNS Based Authentication of Named Entities [RFC6698]). This authentication creates a secure binding between the information provided by the trusted Router Advertisement, and the HTTP server. But this does not mean the Advertising Router and the PvD server belong to the same entity.

The IPv6 prefixes list included in the PvD Additional Information JSON object is used to validate that the prefixes included in the Router Advertisements are really part of the PvD. An adversarial router willing to fake the use of a given explicit PvD, without any access to the actual PvD, would need to perform NAT66 in order to circumvent this check.

It is also RECOMMENDED that the PvD server checks the source addresses of incoming connexions (see Section 4.1). This check ensures that the internet access provided by any router advertising a given PvD eventually reaches the internet using the actual PvD (Tunneling can still be used).

For privacy reasons, it is desirable that the PvD Additional Information object may only be retrieved by the hosts using the given PvD. Host identity SHOULD be validated based on the client address that is used during the HTTP query.

6. Privacy Considerations

TBD

7. IANA Considerations

IANA is kindly requested to allocate a new IPv6 Neighbor Discovery option number for the PvD ID Router Advertisement option.

The URI used to retrieve the PvD Additional Information JSON object is the well known URI (see [RFC5785]) with the URI suffix "pvd".

TBD: JSON keys will need a new registry.

8. Acknowledgements

Many thanks to M. Stenberg and S. Barth for their earlier work: [I-D.stenberg-mif-mpvd-dns].

Thanks also to Ray Bellis, Lorenzo Colitti, Thierry Danis, Marcus Keane, Erik Kline, Jen Lenkova, Mark Townsley, James Woodyatt and Mikael Abrahamson for useful and interesting discussions.

Finally, many thanks to Thierry Danis for his implementation work ([github]), Tom Jones for his integration effort into the Neat project and Rigil Salim for his implementation work.

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.

[RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <http://www.rfc-editor.org/info/rfc2818>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

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

9.2. Informative references

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <http://www.rfc-editor.org/info/rfc3339>.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <http://www.rfc-editor.org/info/rfc4035>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010, <http://www.rfc-editor.org/info/rfc5785>.

[RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <http://www.rfc-editor.org/info/rfc5798>.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <http://www.rfc-editor.org/info/rfc6105>.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <http://www.rfc-editor.org/info/rfc6698>.

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

[RFC7556] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <http://www.rfc-editor.org/info/rfc7556>.

[RFC8028] Baker, F. and B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC 8028, DOI 10.17487/RFC8028, November 2016, <http://www.rfc-editor.org/info/rfc8028>.

[I-D.bowbakova-rtgwg-enterprise-pa-multihoming] Baker, F., Bowers, C., and J. Linkova, "Enterprise Multihoming using Provider-Assigned Addresses without Network Prefix Translation: Requirements and Solution", draft-bowbakova-rtgwg-enterprise-pa-multihoming-01 (work in progress), October 2016.

[I-D.stenberg-mif-mpvd-dns] Stenberg, M. and S. Barth, "Multiple Provisioning Domains using Domain Name System", draft-stenberg-mif-mpvd-dns-00 (work in progress), October 2015.

[I-D.kline-mif-mpvd-api-reqs]
Kline, E., "Multiple Provisioning Domains API Requirements", draft-kline-mif-mpvd-api-reqs-00 (work in progress), November 2015.

[PEN] IANA, "Private Enterprise Numbers", <https://www.iana.org/assignments/enterprise-numbers>.

[IEEE8021X]
IEEE, "IEEE Standards for Local and Metropolitan Area Networks: Port based Network Access Control, IEEE Std", .

[github] Cisco, "IPv6-mPvD github repository", <https://github.com/IPv6-mPvD>.

Appendix A. Changelog

Note to RFC Editors: Remove this section before publication.

A.1. Version 00

Initial version of the draft. Edited by Basile Bruneau + Eric Vyncke and based on Basile’s work.

A.2. Version 01

Major rewrite intended to focus on the the retained solution based on corridors, online, and WG discussions. Edited by Pierre Pfister. The following list only includes major changes.

PvD ID is an FQDN retrieved using a single RA option. This option contains a sequence number for push-based updates, a new H-flag, and a L-flag in order to link the PvD with the IPv4 DHCP server.

A lifetime is included in the PvD ID option.

Detailed Hosts and Routers specifications.

Additional Information is retrieved using HTTP-over-TLS when the PvD ID Option H-flag is set. Retrieving the object is optional.

The PvD Additional Information object includes a validity date.

DNS-based approach is removed as well as the DNS-based encoding of the PvD Additional Information.

Major cut in the list of proposed JSON keys. This document may be extended later if need be.

Monetary discussion is moved to the appendix.

Clarification about the ’prefixes’ contained in the additional information.

Clarification about the processing of DHCPv6.

A.3. Version 02

The FQDN is now encoded with ASCII format (instead of DNS binary) in the RA option.

The PvD ID option lifetime is removed from the object.

Use well known URI "https://<PvD-ID>/.well-known/pvd"

Reference RFC3339 for JSON timestamp format.

The PvD ID Sequence field has been extended to 16 bits.

Modified host behavior for DHCPv4 and DHCPv6.

Removed IKEv2 section.

Removed mention of RFC7710 Captive Portal option. A new I.D. will be proposed to address the captive portal use case.

Appendix B. Connection monetary cost

NOTE: This section is included as a request for comment on the potential use and syntax.

The billing of a connection can be done in a lot of different ways. The user can have a global traffic threshold per month, after which his throughput is limited, or after which he/she pays each megabyte. He/she can also have an unlimited access to some websites, or an unlimited access during the weekends.

An option is to split the bill in elementary billings, which have conditions (a start date, an end date, a destination IP address...). The global billing is an ordered list of elementary billings. To know the cost of a transmission, the host goes through the list, and the first elementary billing whose the conditions are fulfilled gives the cost. If no elementary billing conditions match the request, the host MUST make no assumption about the cost.

B.1. Conditions

Here are the potential conditions for an elementary billing. All conditions MUST be fulfill.

Key	Description	Type	JSON Example
beginDate	Date before which the billing is not valid	ISO 8601	"1977-04-22T06:00:00Z"
endDate	Date after which the billing is not valid	ISO 8601	"1977-04-22T06:00:00Z"
domains	FQDNs whose the billing is limited	array(string)	["deezer.com","spotify.com"]
prefixes4	IPv4 prefixes whose the billing is limited	array(string)	["78.40.123.182/32","78.40.123.183/32"]
prefixes6	IPv6 prefixes whose the billing is limited	array(string)	["2a00:1450:4007:80e::200e/64"]

B.2. Price

Here are the different possibilities for the cost of an elementary billing. A missing key means "all/unlimited/unrestricted". If the elementary billing selected has a trafficRemaining of 0 kb, then it means that the user has no access to the network. Actually, if the last elementary billing has a trafficRemaining parameter, it means that when the user will reach the threshold, he/she will not have access to the network anymore.

Key	Description	Type	JSON Example
pricePerGb	The price per Gigabit	float (currency per Gb)	2
currency	The currency used	ISO 4217	"EUR"
throughputMax	The maximum achievable throughput	float (kb/s)	100000
trafficRemaining	The traffic remaining	float (kB)	12000000

B.3. Examples

Example for a user with 20 GB per month for 40 EUR, then reach a threshold, and with unlimited data during weekends and to example.com:

```
[
  {
    "domains": ["example.com"]
  },
  {
    "prefixes4": ["78.40.123.182/32","78.40.123.183/32"]
  },
  {
    "beginDate": "2016-07-16T00:00:00Z",
    "endDate": "2016-07-17T23:59:59Z",
  },
  {
    "beginDate": "2016-06-20T00:00:00Z",
    "endDate": "2016-07-19T23:59:59Z",
    "trafficRemaining": 12000000
  },
  {
    "throughputMax": 100000
  }
]
```

If the host tries to download data from example.com, the conditions of the first elementary billing are fulfilled, so the host takes this elementary billing, finds no cost indication in it and so deduces that it is totally free. If the host tries to exchange data with foobar.com and the date is 2016-07-14T19:00:00Z, the conditions of the first, second and third elementary billing are not fulfilled.

But the conditions of the fourth are. So the host takes this elementary billing and sees that there is a threshold, 12 GB are remaining.

Another example for a user abroad, who has 3 GB per year abroad, and then pay each MB:

```
[
  {
    "beginDate": "2016-02-10T00:00:00Z",
    "endDate": "2017-02-09T23:59:59Z",
    "trafficRemaining": 3000000
  },
  {
    "pricePerGb": 30,
    "currency": "EUR"
  }
]
```

Authors' Addresses

Pierre Pfister (editor)
Cisco
11 Rue Camille Desmoulins
Issy-les-Moulineaux 92130
France

Email: ppfister@cisco.com

David Schinazi
Apple

Email: dschinazi@apple.com

Tommy Pauly
Apple

Email: tpauly@apple.com

Eric Vyncke
Cisco
De Kleetlaan, 6
Diegem 1831
Belgium

Email: evyncke@cisco.com

Basile Bruneau
Ecole Polytechnique
Vannes 56000
France

Email: basile.bruneau@polytechnique.edu

intarea
Internet-Draft
Intended status: Standards Track
Expires: September 3, 2017

B. Bruneau
Ecole polytechnique
E. Vyncke, Ed.
P. Pfister
Cisco
D. Schinazi
T. Pauly
Apple

March 2, 2017

Proposals to discover Provisioning Domains
draft-bruneau-pvd-00

Abstract

This document describes different possibilities for hosts to retrieve additional information about their Internet access configuration. The set of configuration items required to access the Internet is called a Provisioning Domain (PvD) and is identified by a Fully Qualified Domain Name (or more generally a Uniform Resource Locator).

This document separates the way of getting the Provisioning Domain identifier, the way of getting the Provisioning Domain information and the potential information contained in the Provisioning Domain.

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 September 3, 2017.

Copyright Notice

Table of Contents

1. Introduction

It has become very common in modern networks that hosts have Internet or more specific access through different networking interfaces, tunnels, or next-hop routers. The concept of Provisioning Domain (PvD) was defined in RFC7556 [RFC7556] as a set of network configuration information which can be used by hosts in order to access the network. In this document, PvDs are associated with a Fully Qualified Domain Name (called PvD ID) which is used within the host to identify correlated sets of configuration data and also used to retrieve additional information about the services that the network provides.

Devices connected to the Internet through multiple interfaces would typically be provisioned with one PvD per interface, but it is worth noting that multiple PvDs with different PvD IDs could be provisioned on any host interface, as well as noting that the same PvD ID could be used on different interfaces in order to inform the host that both PvDs, on different interfaces, ultimately provide equivalent services.

This document proposes multiple methods which could be used in order to retrieve the PvD ID associated with a set of networking configuration as well as the methods and format in order to retrieve the associated PvD Information.

2. Terminology

PvD — a provisioning domain, usually with a set of provisioning domain information; for more information, see [RFC7556].

2.1. Requirements Language

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 RFC 2119 [RFC2119].

3. Retrieving the PvD ID

In this document, each provisioning domain is identified by a PvD ID. The PvD ID is a Fully Qualified Domain Name which belongs to the network operator to avoid conflicts among network operators. The same PvD ID can exist in several access networks if the set of configuration information is identical in all those networks (such as

in all home networks of a residential subscriber). Within a host, the PvD ID SHOULD be associated to all the configuration information associated to this PvD ID; this allows for easy update and removal of information while keeping a consistent state.

This section assumes that IPv6 Router Advertisements are used to discover the PvD ID and explains why this technique was selected.

3.1. Using One Router Advertisement per PvD

Hosts receive implicit PvDs by the means of Router Advertisements (RA).

A router MAY add a single PvD ID Option in its RAs. The PvD ID specified in this option is then associated with all the Prefix Information Options (PIO) included in the RA (albeit it is expected that only one PIO will be included in the RA). All other information contained in the RA (notably the RDNSS) are to be associated with the PvD. The set of information contained in the RA forms the bootstrap (or hint) PvD. A new RA option will be required.

When a host receives an RA which does not include a PvD ID Option, the set of information included in the RA is attached to an implicit PvD identified by the local interface ID on which the RA is received, and by the link-local address of the router sending the RA.

In the cases where a router should provide multiple independent PvDs to all hosts, including non-PvD aware hosts, it should send multiple RAs, as proposed in [I-D.bowbakova-rtgwg-enterprise-pa-multihoming] using different source link-local addresses (LLA).

Using RA allows for an early discovery of the PvD ID as it is early in the interface start-up. As RA is usually processed in the kernel, this requires a host OS upgrade. The RA SHOULD contain other PvD information as explained in section Section 4.1.

3.2. Rationale for not selecting other techniques

There are other techniques to discover the PvD ID that were not selected by the authors and reviewers, this section explains why. The design goal was to be as reliable as possible (do not depend on Internet connectivity) and as fast as possible.

3.2.1. Using DNS-SD

For each received RA including a RDNSS option as well as a DNS search list option, the host MAY retrieve the PvD ID by querying the configured DNS server for records of type PTR associated with

_pvd.<DNS search name>. If a PvD ID is configured, the DNS recursive resolver MUST reply with the PvD ID as a PTR record. NXDOMAIN is returned otherwise.

When the RDNSS address is link-local, the host MAY retrieve the PvD ID before configuring its global scope address(es).

Relying on a valid DNS service at the interface bootstrap can lead into delay to start the interface or starting without enough information: for example when the RDNSS is a non local address and there is no Internet connectivity.

3.2.2. Using Reverse DNS lookup

[I-D.stenberg-mif-mpvd-dns] proposes a solution to get the name of the PvD using a reverse DNS lookup based on the host global address(es). It merely relies on prepending a well-known prefix '_pvd' to the reverse lookup, for example ' _pvd....ip6.arpa.'.

However, the PvD information is typically provided by the network operator, whereas the reverse DNS zone could be delegated from the operator to the network user, in which case it would not work.

It also requires a fully functional global address to retrieve the information which may be too late for a correct host configuration. One advantage is that it does not require any change in the IPv6 protocol and no change in the host kernel or even in the CPE.

3.3. Linking IPv4 Information to an IPv6 PvD

The document describes IPv6-only PvD but there are multiple ways to link the set of IPv4 configuration information received by DHCPv4:

- correlation based on the data-link layer address of the source, if the IPv6 RA and the DHCPv4 response have the same data-link layer address, then the information contained in the IPv4 DHCP can be linked to the IPv6 PvD;

- correlation based on the interface when there is no data-link address on the link (such as a 3GPP link), then the information contained in the IPv4 PDP context can be linked to the IPv6 PvD (*** TO BE VERIFIED before going -01);

- correlation based on the DNS search list, if the DNS search lists are identical between the IPv6 RDNSS and the DHCPV4 response, then the information contained in the IPv4 DHCP response can be linked to the IPv6 PvD.

The correlation could be useful for some PvD information such as Internet reachability, use of captive portal, display name of the PvD, ...

In cases where the IPv4 configuration information could not be associated with a PvD, hosts MUST consider it as attached to an independent implicit PvD containing no other information than what is provided through DHCPv4.

4. Getting the PvD information

Once the PvD ID is known, it MAY be used to retrieve additional information. PvD Information is modeled as a key-value dictionary which keys are ASCII strings of arbitrary length, and values are either strings (encoding can vary), ordered list of values (recursively), or a dictionary (recursively).

The PvD Information may be retrieved from multiple sources (from the bootstrap PvD contained in the RA to the secondary/extended PvD described in this section); the PvD ID is then used to correlate the information from different sources. The way a host should operate when receiving conflicting information is TBD.

4.1. Using the PvD Bootstrap Information Option

Routers MAY transmit, in addition to the PvD ID option, a PvD Bootstrap Information option, containing a first subset of PvD information.

As there is a size limit on the amount of information a single RA can convey, it is likely that the PvD Bootstrap Information option may not contain the whole set of PvD Information. The set of PvD information included in the RA is therefore called PvD Bootstrap Information.

4.2. Downloading a JSON file over HTTPS

The host SHOULD try to download a JSON formatted file over HTTPS in order to get more PvD information.

The host MUST perform an HTTP query to https://<PvD-ID>/v1.json. If the HTTP status of the answer is greater than 400 the host MUST abandon and consider that there is no PvD. If the HTTP status of the answer is between 300 and 400 it MUST follow the redirection(s). If the HTTP status of the answer is between 200 and 300 the host MAY get a file containing a single JSON object.

The host MUST respect the cache information in the HTTP header if any and at expiration of the downloaded object, it must fetch a fresher version if any.

4.2.1. Advantages

The JSON format allows advanced structures.

It can be secured using HTTPS (and DNSSEC).

It is easier to update a file on a web server than to edit DNS records. It can be especially important if we want providers to be able to often update the remaining phone plan of the user.

4.2.2. Disadvantages

It is slower than using DNS because HTTPS uses TCP and TLS and needs more packets to be exchanged to get the file.

An additional HTTPS server must be deployed and configured.

4.3. Using DNS TXT ressource records (not selected)

This approach was not selected during the design team meeting but has kept here for reference, it will be removed after global consensus is reached.

The host could perform a DNS query for TXT resource records (RR) for the FQDN used as PvD ID. For each retrieved PvD ID, the DNS query MUST be sent to the DNS server configured from the same router advertisement as the PvD ID. Syntax of the TXT response is defined in Section 5 (Section 5).

4.3.1. Advantages

It requires a single round-time trip in order to retrieve the PvD Information.

It can be secured using DNSSEC.

4.3.2. Disadvantages

A TXT record is limited to 65535 characters in theory but large size of TXT records could require either DNS over TCP (so loosing the 1-RTT advantage) or fragmented UDP packets (which could be dropped by a bad choice of security policy). Large TXT records could also be used to mount an amplification attack.

4.3.3. Using DNS SRV ressource records

It is expected that the DNS TXT records will be sufficient for the host to configure itself with basic networking and policy configuration. Nevertheless, if further information is required, or when a different security model shall be used to access the PvD Information, a SRV Resource Record including a full URL MAY be included as a response, expecting the host to query this URL in order to retrieve additional PvD information.

5. PvD Information

PvD information is a set of key-value pairs. Keys are ASCII character strings. Values are either a character string, an ordered list of values, or an embedded dictionary. Value types and default behavior with respect to some specific keys MAY be further specified (recursively). Some keys have a default value as described in the following sections. When there is an expiration time in a PvD, then the information MUST be refreshed before the expiration time. The behavior of a host when the refresh operation is not successful is TBD.

Note, the DNS TXT key has been kept even if not selected by the design team but has been kept here for reference.

5.1. PvD Name

PvD SHOULD have a human readable name in order to be presented on a GUI. The name can also be localized.

DNS TXT key	JSON key	Description	Type	Example
n	name	User-visible service name, SHOULD be part of the bootstrap PvD	human-readable UTF-8 string	"Foobar Service"
nl10n	localizedName	Localized user-visible service name, language can be selected based on the HTTP Accept-Language header in the request.	human-readable UTF-8 string	"Service Blabla"

5.2. Trust of the bootstrap PvD

The content of the bootstrap PvD (from the original RA) cannot be trusted as it is not authenticated. But, the extended PvD can be associated with the PvD ID (as the PvD ID is used to construct the extended PvD URL) and trusted by the used of TLS. The extended PvD SHOULD therefore include the following information elements and, if they are present, the host MUST verify that the PIO of the RA fits into the master prefix list. The values of the bootstrap PvD (RDNSS, ...) are overwritten by the values contained in the extended PvD if they are present.

DNS TXT key	JSON key	Description	Type	Example
mp6	masterIpv6Prefix	All the IPv6 prefixes linked to this PvD (such as a /29 for the ISP).	Array of IPv6 prefixes	["2001:db8::/32"]

5.3. Reachability

The following set of keys can be used to specify the set of services for which the respective PvD should be used. If present they MUST be honored by the client, i.e., if the PvD is marked as not usable for Internet access (walled garden), then it MUST NOT be used for Internet access. If the usability is limited to a certain set of domain or address prefixes (typical VPN access), then a different PvD MUST be used for other destinations.

DNS TXT key	JSON key	Description	Type	Example
s	noInternet	Internet inaccessible	boolean	true
lp	loginPortal	Presence of a login portal	boolean	false
z	dnsZones	DNS zones accessible and searchable	array of DNS zone	["foo.com","sub.bar.com"]
6	prefixes6	IPv6-prefixes accessible via this PvD	array of IPv6 prefixes	["2001:db8:a::/48","2001:db8:b:c::/64"]
4	prefixes4	IPv4-prefixes accessible	array of IPv4 prefixes in CIDR reachable via this PvD	["192.0.2.0/24","2.3.0.0/16"]

5.4. Connectivity Characteristics

NOTE: open question to the authors/reviewers: should this document include this section or is it useless?

The following set of keys can be used to signal certain characteristics of the connection towards the PvD.

They should reflect characteristics of the overall access technology which is not limited to the link the host is connected to, but rather a combination of the link technology, CPE upstream connectivity, and further quality of service considerations.

DNS TXT key	JSON key	Description	Type	Example
tp	throughputMax	Maximum achievable throughput (e.g. CPE downlink/uplink)	object({down (int), up(int)}) in kb/s	{"down": 10000, "up": 5000}
lt	latencyMin	Minimum achievable latency	object({down (int), up(int)}) in ms	{"down": 10, "up": 20}
rl	reliabilityMax	Maximum achievable reliability	object({down (int), up(int)}) in 1/1000	{"down": 1000, "up": 800}
cp	captiveUrl	Captive portal	URL of the portal	"https://example.com"
nat	nat	IPv4 NAT in place	boolean	true
srh	segmentRoutingHeader	The IPv6 Segment Routing Header to be used between the IPv6 header and any other headers when using this PvD	Binary string	...
srhDNS	segmentRoutingHeaderDnsFQDN	The DNS FQDN which is used to retrieved the actual IPv6 Segment Routing Header to be used between the IPv6 header and	Ascii string	srh.pvd-foo.example.org

		any other headers when using this PvD			
cost	cost	Cost of using the connection	object	See Section 5.5	

5.5. Connection monetary cost

NOTE: This section is included as a request for comment on the potential use and syntax.

The billing of a connection can be done in a lot of different ways. The user can have a global traffic threshold per month, after which his throughput is limited, or after which he/she pays each megabyte. He/she can also have an unlimited access to some websites, or an unlimited access during the week-ends.

We propose to split the final billing in elementary billings, which have conditions (a start date, an end date, a destination IP address...). The global billing is an ordered list of elementary billings. To know the cost of a transmission, the host goes through the list, and the first elementary billing whose the conditions are fulfilled gives the cost. If no elementary billing conditions match the request, the host MUST NOT make any assumption about the cost.

5.5.1. Conditions

Here are the potential conditions for an elementary billing. All conditions MUST be fulfilled.

Note: the final version should use shorter key names.

Key	Description	Type	Example
beginDate	Date before which the billing is not valid	ISO 8601	"1977-04-22T06:00:00Z"
endDate	Date after which the billing is not valid	ISO 8601	"1977-04-22T06:00:00Z"
domains	FQDNs whose the billing is limited	array(string)	["deezer.com","spotify.com"]
prefixes4	IPv4 prefixes whose the billing is limited	array(string)	["78.40.123.182/32","78.40.123.183/32"]
prefixes6	IPv6 prefixes whose the billing is limited	array(string)	["2a00:1450:4007:80e::200e/64"]

5.5.2. Price

Here are the different possibilities for the cost of an elementary billing. A missing key means "all/unlimited/unrestricted". If the elementary billing selected has a trafficRemaining of 0 kb, then it means that the user has no access to the network. Actually, if the last elementary billing has a trafficRemaining parameter, it means that when the user will reach the threshold, he/she will not have access to the network anymore.

Key	Description	Type	Example
pricePerGb	The price per Gigabit	float (currency per Gb)	2
currency	The currency used	ISO 4217	"EUR"
throughputMax	The maximum achievable throughput	float (kb/s)	1000
trafficRemaining	The traffic remaining	float (kb)	96000000

5.5.3. Examples

Example for a user with 20 GB per month for 40 EUR, then reach a threshold, and with unlimited data during week-ends and to the server "deezer":

```
[
  {
    "domains": ["deezer.com"]
  },
  {
    "prefixes4": ["78.40.123.182/32","78.40.123.183/32"]
  },
  {
    "beginDate": "2016-07-16T00:00:00Z",
    "endDate": "2016-07-17T23:59:59Z",
  },
  {
    "beginDate": "2016-06-20T00:00:00Z",
    "endDate": "2016-07-19T23:59:59Z",
    "trafficRemaining": 96000000
  },
  {
    "throughputMax": 1000
  }
]
```

If the host tries to download data from deezer.com, the conditions of the first elementary billing are fulfilled, so the host takes this elementary billing, finds no cost indication in it and so deduces that it is totally free. If the host tries to exchange data with youtube.com and the date is 2016-07-14T19:00:00Z, the conditions of the first, second and third elementary billing are not fulfilled.

But the conditions of the fourth are. So the host takes this elementary billing and sees that there is a threshold, 12 GB are remaining.

Another example for a user abroad, who has 3 GB per year abroad, and then pay each MB:

```
[
  {
    "beginDate": "2016-02-10T00:00:00Z",
    "endDate": "2017-02-09T23:59:59Z",
    "trafficRemaining": 9200000
  },
  {
    "pricePerGb": 30,
    "currency": "EUR"
  }
]
```

5.6. Private Extensions

keys starting with "x-" are reserved for private use and can be utilized to provide vendor-, user- or enterprise-specific information. It is RECOMMENDED to use one of the patterns "x-FQDN-KEY" or "x-PEN-KEY" where FQDN is a fully qualified domain name or PEN is a private enterprise number [PEN] under control of the author of the extension to avoid collisions.

5.7. Examples

5.7.1. Using JSON

```
{
  "name": "Orange France",
  "localizedName": "Orange France",
  "dnsServers": ["8.8.8.8", "8.8.4.4"],
  "throughputMax": {
    "down": 100000,
    "up": 20000
  },
  "cost": [
    {
      "domains": ["deezer.com"]
    },
    {
      "prefixes4": ["78.40.123.182/32","78.40.123.183/32"]
    },
    {
      "beginDate": "2016-07-16T00:00:00Z",
      "endDate": "2016-07-17T23:59:59Z",
    },
    {
      "beginDate": "2016-06-20T00:00:00Z",
      "endDate": "2016-07-19T23:59:59Z",
      "trafficRemaining": 96000000
    },
    {
      "throughputMax": 1000
    }
  ]
}
```

5.7.2. Using DNS TXT records

```
n=Orange France
r=8.8.8.8,8.8.4.4
tp=100000,20000
cost+0+domains=deezer.com
cost+1+prefixes4=78.40.123.182/32,78.40.123.183/32
cost+2+beginDate=2016-07-16T00:00:00Z
cost+2+endDate=2016-07-17T23:59:59Z
cost+3+beginDate=2016-06-20T00:00:00Z
cost+3+endDate=2016-07-19T23:59:59Z
cost+3+trafficRemaining=96000000
cost+4+throughputMax=1000
```

6. Security Considerations

While the PvD ID can be forged easily, if the host retrieve the extended PvD via TLS, then the host can trust the content of the extended PvD and verifies that the RA prefix(es) are indeed included in the extended PvD.

7. Acknowledgements

Many thanks to M. Stenberg and S. Barth: Section 5.3, Section 5.4 and Section 5.6 are from their document [I-D.stenberg-mif-mpvd-dns].

Thanks also to Ray Bellis, Lorenzo Colitti, Erik Kline, Mark Townsley and James Woodyatt for useful and interesting brainstorming sessions.

8. References

8.1. 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>.

[RFC7556] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <http://www.rfc-editor.org/info/rfc7556>.

8.2. Informative references

[I-D.bowbakova-rtgwg-enterprise-pa-multihoming] Baker, F., Bowers, C., and J. Linkova, "Enterprise Multihoming using Provider-Assigned Addresses without Network Prefix Translation: Requirements and Solution", draft-bowbakova-rtgwg-enterprise-pa-multihoming-01 (work in progress), October 2016.

[I-D.stenberg-mif-mpvd-dns] Stenberg, M. and S. Barth, "Multiple Provisioning Domains using Domain Name System", draft-stenberg-mif-mpvd-dns-00 (work in progress), October 2015.

[PEN] IANA, "Private Enterprise Numbers", <https://www.iana.org/assignments/enterprise-numbers>.

Authors' Addresses

Basile Bruneau
Ecole polytechnique
Vannes 56000
France

Email: basile.bruneau@polytechnique.edu

Eric Vyncke (editor)
Cisco
De Kleetlaan, 6
Diegem 1831
Belgium

Email: evyncke@cisco.com

Pierre Pfister
Cisco
11 Rue Camille Desmoulins
Issy-les-Moulineaux 92130
France

Email: ppfister@cisco.com

David Schinazi
Apple

Email: dschinazi@apple.com

Tommy Pauly
Apple

Email: tpauly@apple.com

IPv6 Operations Working Group (v6ops)
Internet-Draft
Intended status: Best Current Practice
Expires: September 14, 2017

F. Gont
SI6 Networks / UTN-FRH
G. Doering
SpaceNet AG
M. Garcia Corbo
SITRANS
G. Gont
SI6 Networks
March 13, 2017

On the Dynamic/Automatic Configuration of IPv6 Hosts
draft-gont-v6ops-host-configuration-01

Abstract

IPv6 has two different mechanisms for dynamic/automatic host configuration: SLAAC and DHCPv6. These two mechanisms allow for the configuration of IPv6 addresses and a number of network parameters. While there is overlap in the parameters that can be configured via these two protocols, different implementations support only subsets of such parameters with either mechanism, or have no support for DHCPv6 at all. This document analyzes a problem that arises from this situation, and mandates that all host implementations support RFC 6105 (DNS options for SLAAC) and the stateless DHCPv6 functionality in RFC 3315.

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 September 14, 2017.

Table of Contents

1. Introduction

IPv6 has two different mechanisms for dynamic/automatic host configuration: Stateless Address Autoconfiguration (SLAAC) [RFC4862] and Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC3315]. SLAAC allows for distributed address assignment (where each host automatically configures its own IPv6 addresses) and basic network configuration (such as recursive DNS servers and DNS search lists). On the other hand, DHCPv6 provides for centralized address assignment (the DHCPv6 server leases IPv6 addresses to hosts) and richer network configuration (NTP servers, web proxys, etc.).

Traditionally, SLAAC has been seen as a more lightweight mechanism, suitable for resource-constrained devices, while DHCPv6 has been seen more as heavy-weight and full-fledged mechanism. We note that this

distinction is rather questionable, and is essentially meaningless for typical mobile devices or home appliances.

Among the possible configuration information that can be conveyed with both SLAAC and DHCPv6 is DNS related configuration: recursive DNS servers and DNS search lists. Configuring this information is probably as vital in practice as configuring IPv6 addresses, since for obvious reasons both humans and popular applications operate on names (rather than on IPv6 addresses). The ability to convey this information has always been part of DHCPv6, while for the SLAAC case, support was added in a separate document that standardizes "IPv6 Router Advertisement Options for DNS Configuration" [RFC6106].

Unfortunately, different host and router implementations provide support for only a subset of these options. For example, some host implementations (e.g., Android) support SLAAC DNS options [RFC6106], but do not support stateless DHCPv6. On the other hand, other host implementations (e.g., Microsoft Windows) support stateless DHCPv6, but do not support [RFC6106]. Similarly, some router implementations support [RFC6106], while others do not.

This represents a problem for IPv6 deployment, since:

1. in order to support most popular IPv6 host implementations, IPv6 networks are required to support *both* SLAAC and DHCPv6.

2. some router implementations do not support [RFC6106] and hence support for the SLAAC DNS options may be impossible or require yet an additional network element or network service to support [RFC6106]

We note that, in most cases, this problem is currently masked by the fact that most IPv6 deployments are actually dual-stack, and hence hosts can currently rely DNS-related information being obtained via IPv4-based DHCP. However, at the point such deployments disable IPv4 to become IPv6-only, the aforementioned problems will become evident, possibly as a surprise to network operators.

2. Terminology

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. Current Requirements regarding RDNSS and Stateless DHCPv6

Section 7.2.1 of [RFC6434] ("IPv6 Node Requirements") states:

> IPv6 nodes use DHCP [RFC3315] to obtain address configuration information (see Section 5.9.5) and to obtain additional (non-address) configuration. If a host implementation supports applications or other protocols that require configuration that is only available via DHCP, hosts SHOULD implement DHCP.

Since DNS information is (in theory) also available via RA messages, the aforementioned text essentially makes support for stateless DHCPv6 optional.

Regarding SLAAC DNS options, [RFC6434] states, in Section 7.3,

- Implementations SHOULD implement the DNS RA option [RFC6106].

which certainly is not clear whether it is referring to hosts, routers, or both.

In any case, we note that [RFC6434] has been published on the "Informational" track, and hence implementations may completely ignore this RFC while still claiming full-compliance with all the relevant IETF standards.

[RFC7084] ("Basic Requirements for IPv6 Customer Edge Routers") requires support for "DNS_SERVERS [RFC3646]" option and the SLAAC DNS options in the IPv6 CE Routers. As with [RFC6434], it was published on the "Informational" track.

4. Requirements for IPv6 Hosts

IPv6 hosts MUST support the SLAAC DNS options specified in [RFC6106], and the stateless DHCPv6 mechanism specified in [RFC3315].

5. Requirements for IPv6 Routers

IPv6 routers MUST support the SLAAC DNS options specified in [RFC6106].

6. IANA Considerations

This document has no actions for IANA. The RFC-Editor should remove this section prior to publication of this document as an RFC.

7. Security Considerations

Host implementations supporting SLAAC are subject to a number of attacks based on forged ICMPv6 Router Advertisement [RFC4861] messages. Such attacks can be mitigated by means of RA-Guard [RFC6105] [RFC7113]. Hosts supporting DHCPv6 are subject to a number of attacks based on forged DHCPv6-server messages. Such attacks can be mitigated by means of DHCPv6-Shield [RFC7610].

8. Acknowledgements

The authors would like to thank Chuck Anderson, Brian Carpenter, Nick Hilliard, Philip Homburg, Mark Smith, Barbara Stark, and several participants of the v6ops wg (TBD) for providing valuable comments on earlier versions of this document.

Fernando Gont would like to thank Nelida Garcia and Jorge Oscar Gont for their love and support, and Ivan Arce and Diego Armando Maradona for their inspiration.

9. References

9.1. 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>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, DOI 10.17487/RFC6106, November 2010, <http://www.rfc-editor.org/info/rfc6106>.

9.2. Informative References

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <http://www.rfc-editor.org/info/rfc6105>.

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

[RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <http://www.rfc-editor.org/info/rfc7084>.

[RFC7113] Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, DOI 10.17487/RFC7113, February 2014, <http://www.rfc-editor.org/info/rfc7113>.

[RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers", BCP 199, RFC 7610, DOI 10.17487/RFC7610, August 2015, <http://www.rfc-editor.org/info/rfc7610>.

Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: http://www.si6networks.com

Gert Doering
SpaceNet AG
Joseph-Dollinger-Bogen 14
Muenchen D-80807
Germany

Email: gert@space.net

Madeleine Garcia Corbo
Servicios de Informacion del Transporte
Neptuno 358
Havana City 10400
Cuba

Email: madelen.garcia16@gmail.com

Guillermo Gont
SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: ggont@si6networks.com
URI: https://www.si6networks.com

Network Working Group
Internet-Draft
Obsoletes: 7048 (if approved)
Intended status: Informational
Expires: August 31, 2017

J. Palet Martinez
Consulintel, S.L.
February 27, 2017

Basic Requirements for IPv6 Customer Edge Routers
draft-palet-v6ops-rfc7084-bis-01

Abstract

This document specifies requirements for an IPv6 Customer Edge (CE) router. Specifically, the current version of this document focuses on the basic provisioning of an IPv6 CE router and the provisioning of IPv6 hosts attached to it. The document also covers several transition technologies, as required in a world where IPv4 addresses are no longer available, so hosts in the customer LANs with IPv4-only or IPv6-only applications or devices, requiring to communicate with IPv4-only services at the Internet, are able to do so. The document obsoletes RFC 7084.

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 31, 2017.

Copyright Notice

Table of Contents

1. Introduction

This document defines basic IPv6 features for a residential or small-office router, referred to as an "IPv6 CE router", in order to establish an industry baseline for features to be implemented on such a router.

These routers typically also support IPv4, at least in the LAN side.

This document specifies how an IPv6 CE router automatically provisions its WAN interface, acquires address space for provisioning of its LAN interfaces, and fetches other configuration information

from the service provider network. Automatic provisioning of more complex topology than a single router with multiple LAN interfaces is out of scope for this document. In some cases manual provisioning may be acceptable, when intended for a small number of customers.

See [RFC4779] for a discussion of options available for deploying IPv6 in service provider access networks.

This document also covers the IP transition technologies required in a world where IPv4 addresses are no longer available, so the service providers need to provision IPv6-only WAN access, while at the same time ensuring that IPv4-only or IPv6-only devices or applications in the customer LANs can still reach IPv4-only devices or applications in Internet, which still don’t have IPv6 support.

1.1. Requirements Language

Take careful note: Unlike other IETF documents, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are not used as described in RFC 2119 [RFC2119]. This document uses these keywords not strictly for the purpose of interoperability, but rather for the purpose of establishing industry-common baseline functionality. As such, the document points to several other specifications (preferable in RFC or stable form) to provide additional guidance to implementers regarding any protocol implementation required to produce a successful CE router that interoperates successfully with a particular subset of currently deploying and planned common IPv6 access networks.

2. Terminology

End-User Network	one or more links attached to the IPv6 CE router that connect IPv6 hosts.
IPv6 Customer Edge Router	a node intended for home or small-office use that forwards IPv6 packets not explicitly addressed to itself. The IPv6 CE router connects the end-user network to a service provider network. In other documents, the CE is named as CPE (Customer Premises Equipment or Customer Provided Equipment). In the context of this document, both terminologies are synonymous.

IPv6 Host	any device implementing an IPv6 stack receiving IPv6 connectivity through the IPv6 CE router.
LAN Interface	an IPv6 CE router’s attachment to a link in the end-user network. Examples are Ethernet (simple or bridged), 802.11 wireless, or other LAN technologies. An IPv6 CE router may have one or more network-layer LAN interfaces.
Service Provider	an entity that provides access to the Internet. In this document, a service provider specifically offers Internet access using IPv6, and it may also offer IPv4 Internet access. The service provider can provide such access over a variety of different transport methods such as FTTH, DSL, cable, wireless, LTE, and others.
WAN Interface	an IPv6 CE router’s attachment to a link used to provide connectivity to the service provider network; example link technologies include Ethernet (simple or bridged), PPP links, Frame Relay, or ATM networks, as well as Internet-layer (or higher-layer) "tunnels", such as tunnels over IPv4 or IPv6 itself.

3. Usage Scenarios

The IPv6 CE router described in this document is expected to be used typically in any of the following scenarios:

1. Residential/household users. Common usage is any kind of Internet access (web, email, streaming, online gaming, etc.).

2. Residential with Small Office/Home Office (SOHO). Same usage as for the first scenario.

3. Small Office/Home Office (SOHO). Same usage as for the first scenario.

4. Small and Medium Enterprise (SME). Same usage as for the first scenario.

5. Residential/household with advanced requirements. Same basic usage as for the first scenario, however there may be

requirements for exporting services to the WAN (IP cameras, web, DNS, email, VPN, etc.).

6. Small and Medium Enterprise (SME) with advanced requirements. Same basic usage as for the first scenario, however there may be requirements for exporting services to the WAN (IP cameras, web, DNS, email, VPN, etc.).

The above list is not intended to be comprehensive of all the possible usage scenarios, just the main ones. In fact, combinations of the above usages are also possible, for example a residential with SOHO and advanced requirements.

The mechanisms for exporting IPv6 services are commonly "naturally" available in any IPv6 router, as when using GUA, unless they are blocked by firewall rules, which may require some manual configuration by means of a GUI and/or CLI.

However in the case of IPv4, because the usage of private addresses and NAT, it typically requires some degree of manual configuration such as setting up a DMZ, virtual servers, or port/protocol forwarding. In general, CE routers already provide GUI and/or CLI to manually configure them, or the possibility to setup the CE in bridge mode, so another CE behind it, takes care of that. It is out of the scope of this document the definition of any requirements for that.

The main difference for an IPv6 CE router to support one or several of the above indicated scenarios, is related to the packet processing capabilities, performance, even other details such as the number of WAN/LAN interfaces, their maximum speed, memory for keeping tables or tracking connections, etc. So, it is out of the scope of this document to classify them.

For example, an SME may have just 10 employees (micro-SME), which commonly will be considered same as a SOHO, but a small SME can have up to 50 employees, or 250 for a medium one. Depending on the IPv6 CE router capabilities or even how it is being configured (for instance, using SLAAC or DHCPv6), it may support even a higher number of employees if the traffic in tha LANs is low, or switched by another device(s), or the WAN bandwith requirements are low, etc. The actual bandwidth capabilities of access with technologies such as FTTH, cable and even LTE, allows the support of such usages, and indeed is a very common situation that access networks and the CE provided by the service provider are the same for SMEs and residential users.

There is also no difference in terms of who actually provides the IPv6 CE router. In most of the cases is the service provider, and in

fact is responsible, typically, of provisioning/managing at least the WAN side. However, commonly the user has access to configure the LAN interfaces, firewall, DMZ, and many other aspects. In fact, in many cases, the user must supply, or at least can replace the IPv6 CE router, which makes even more relevant that all the IPv6 CE routers, support the same requirements defined in this document, .

The IPv6 CE router described in this document is not intended for usage in other scenarios such as bigger Enterprises, Data Centers, Content Providers, etc. So, even if the documented requirements meet their needs, may have additional requirements, which are out of the scope of this document.

4. Architecture

4.1. Current IPv4 End-User Network Architecture

An end-user network will likely support both IPv4 and IPv6. It is not expected that an end user will change their existing network topology with the introduction of IPv6. There are some differences in how IPv6 works and is provisioned; these differences have implications for the network architecture. A typical IPv4 end-user network consists of a "plug and play" router with NAT functionality and a single link behind it, connected to the service provider network.

A typical IPv4 NAT deployment by default blocks all incoming connections. Opening of ports is typically allowed using a Universal Plug and Play Internet Gateway Device (UPnP IGD) [UPnP-IGD] or some other firewall control protocol.

Another consequence of using private address space in the end-user network is that it provides stable addressing; that is, it never changes even when you change service providers, and the addresses are always there even when the WAN interface is down or the customer edge router has not yet been provisioned.

Many existing routers support dynamic routing (which learns routes from other routers), and advanced end-users can build arbitrary, complex networks using manual configuration of address prefixes combined with a dynamic routing protocol.

4.2. IPv6 End-User Network Architecture

The end-user network architecture for IPv6 should provide equivalent or better capabilities and functionality than the current IPv4 architecture.

The end-user network is a stub network. Figure 1 illustrates the model topology for the end-user network.

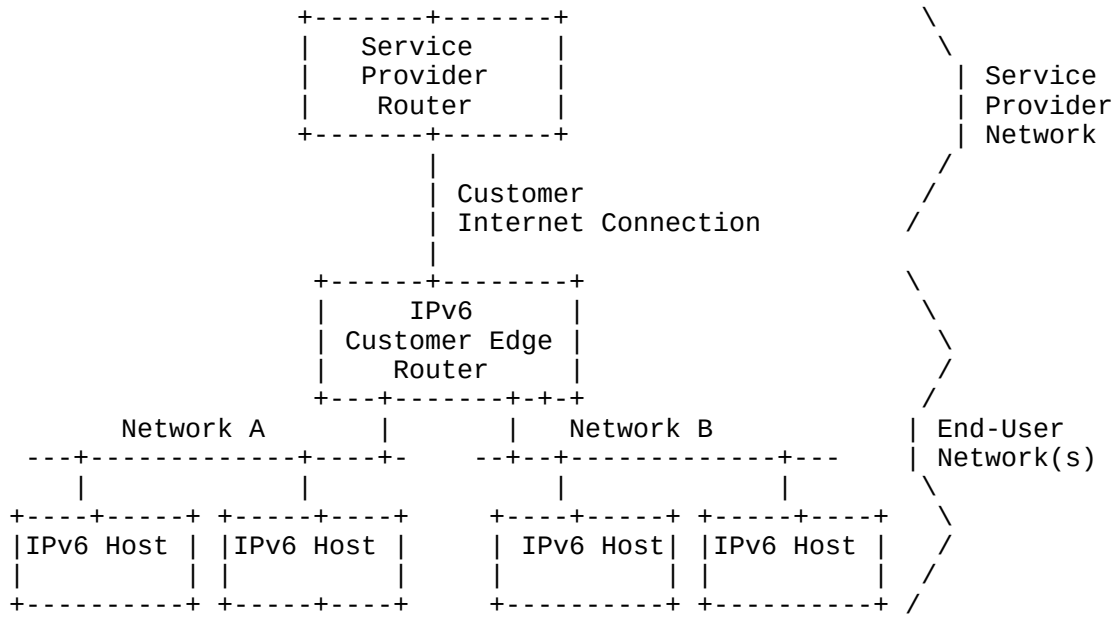


Figure 1: An Example of a Typical End-User Network

This architecture describes the:

- Basic capabilities of an IPv6 CE router
- Provisioning of the WAN interface connecting to the service provider
- Provisioning of the LAN interfaces

For IPv6 multicast traffic, the IPv6 CE router may act as a Multicast Listener Discovery (MLD) proxy [RFC4605] and may support a dynamic multicast routing protocol.

The IPv6 CE router may be manually configured in an arbitrary topology with a dynamic routing protocol. Automatic provisioning and configuration are described for a single IPv6 CE router only.

4.2.1. Local Communication

Link-local IPv6 addresses are used by hosts communicating on a single link. Unique Local IPv6 Unicast Addresses (ULAs) [RFC4193] are used by hosts communicating within the end-user network across multiple links, but without requiring the application to use a globally routable address. The IPv6 CE router defaults to acting as the demarcation point between two networks by providing a ULA boundary, a multicast zone boundary, and ingress and egress traffic filters.

At the time of this writing, several host implementations do not handle the case where they have an IPv6 address configured and no IPv6 connectivity, either because the address itself has a limited topological reachability (e.g., ULA) or because the IPv6 CE router is not connected to the IPv6 network on its WAN interface. To support host implementations that do not handle multihoming in a multi-prefix environment [RFC7157], the IPv6 CE router should not, as detailed in the requirements below, advertise itself as a default router on the LAN interface(s) when it does not have IPv6 connectivity on the WAN interface or when it is not provisioned with IPv6 addresses. For local IPv6 communication, the mechanisms specified in [RFC4191] are used.

ULA addressing is useful where the IPv6 CE router has multiple LAN interfaces with hosts that need to communicate with each other. If the IPv6 CE router has only a single LAN interface (IPv6 link), then link-local addressing can be used instead.

Coexistence with IPv4 requires any IPv6 CE router(s) on the LAN to conform to these recommendations, especially requirements ULA-5 and L-4 below.

5. Requirements

5.1. General Requirements

The IPv6 CE router is responsible for implementing IPv6 routing; that is, the IPv6 CE router must look up the IPv6 destination address in its routing table to decide to which interface it should send the packet.

In this role, the IPv6 CE router is responsible for ensuring that traffic using its ULA addressing does not go out the WAN interface and does not originate from the WAN interface.

G-1: An IPv6 CE router is an IPv6 node according to the IPv6 Node Requirements specification [RFC6434].

G-2: The IPv6 CE router MUST implement ICMPv6 according to [RFC4443]. In particular, point-to-point links MUST be handled as described in Section 3.1 of [RFC4443].

G-3: The IPv6 CE router MUST NOT forward any IPv6 traffic between its LAN interface(s) and its WAN interface until the router has successfully completed the IPv6 address and the delegated prefix acquisition process.

G-4: By default, an IPv6 CE router that has no default router(s) on its WAN interface MUST NOT advertise itself as an IPv6 default router on its LAN interfaces. That is, the "Router Lifetime" field is set to zero in all Router Advertisement messages it originates [RFC4861].

G-5: By default, if the IPv6 CE router is an advertising router and loses its IPv6 default router(s) and/or detects loss of connectivity on the WAN interface, it MUST explicitly invalidate itself as an IPv6 default router on each of its advertising interfaces by immediately transmitting one or more Router Advertisement messages with the "Router Lifetime" field set to zero [RFC4861].

5.2. WAN-Side Configuration

The IPv6 CE router will need to support connectivity to one or more access network architectures. This document describes an IPv6 CE router that is not specific to any particular architecture or service provider and that supports all commonly used architectures.

IPv6 Neighbor Discovery and DHCPv6 protocols operate over any type of IPv6-supported link layer, and there is no need for a link-layer-specific configuration protocol for IPv6 network-layer configuration options as in, e.g., PPP IP Control Protocol (IPCP) for IPv4. This section makes the assumption that the same mechanism will work for any link layer, be it Ethernet, the Data Over Cable Service Interface Specification (DOCSIS), PPP, or others.

WAN-side requirements:

W-1: When the router is attached to the WAN interface link, it MUST act as an IPv6 host for the purposes of stateless [RFC4862] or stateful [RFC3315] interface address assignment.

W-2: The IPv6 CE router MUST generate a link-local address and finish Duplicate Address Detection according to [RFC4862] prior to sending any Router Solicitations on the interface. The

source address used in the subsequent Router Solicitation MUST be the link-local address on the WAN interface.

W-3: Absent other routing information, the IPv6 CE router MUST use Router Discovery as specified in [RFC4861] to discover a default router(s) and install a default route(s) in its routing table with the discovered router's address as the next hop.

W-4: The router MUST act as a requesting router for the purposes of DHCPv6 prefix delegation ([RFC3633]).

W-5: The IPv6 CE router MUST use a persistent DHCP Unique Identifier (DUID) for DHCPv6 messages. The DUID MUST NOT change between network-interface resets or IPv6 CE router reboots.

W-6: The WAN interface of the CE router SHOULD support a Port Control Protocol (PCP) client as specified in [RFC6887] for use by applications on the CE router. The PCP client SHOULD follow the procedure specified in Section 8.1 of [RFC6887] to discover its PCP server. This document takes no position on whether such functionality is enabled by default or mechanisms by which users would configure the functionality. Handling PCP requests from PCP clients in the LAN side of the CE router is out of scope.

Link-layer requirements:

WLL-1: If the WAN interface supports Ethernet encapsulation, then the IPv6 CE router MUST support IPv6 over Ethernet [RFC2464].

WLL-2: If the WAN interface supports PPP encapsulation, the IPv6 CE router MUST support IPv6 over PPP [RFC5072].

WLL-3: If the WAN interface supports PPP encapsulation, in a dual-stack environment with IPCP and IPV6CP running over one PPP logical channel, the Network Control Protocols (NCPs) MUST be treated as independent of each other and start and terminate independently.

Address assignment requirements:

WAA-1: The IPv6 CE router MUST support Stateless Address Autoconfiguration (SLAAC) [RFC4862].

WAA-2: The IPv6 CE router MUST follow the recommendations in Section 4 of [RFC5942], and in particular the handling of the L flag in the Router Advertisement Prefix Information option.

WAA-3: The IPv6 CE router MUST support DHCPv6 [RFC3315] client behavior.

WAA-4: The IPv6 CE router MUST be able to support the following DHCPv6 options: Identity Association for Non-temporary Address (IA_NA), Reconfigure Accept [RFC3315], and DNS_SERVERS [RFC3646]. The IPv6 CE router SHOULD be able to support the DNS Search List (DNSSL) option as specified in [RFC3646].

WAA-5: The IPv6 CE router SHOULD implement the Network Time Protocol (NTP) as specified in [RFC5905] to provide a time reference common to the service provider for other protocols, such as DHCPv6, to use. If the CE router implements NTP, it requests the NTP Server DHCPv6 option [RFC5908] and uses the received list of servers as primary time reference, unless explicitly configured otherwise. LAN side support of NTP is out of scope for this document.

WAA-6: If the IPv6 CE router receives a Router Advertisement message (described in [RFC4861]) with the M flag set to 1, the IPv6 CE router MUST do DHCPv6 address assignment (request an IA_NA option).

WAA-7: If the IPv6 CE router does not acquire a global IPv6 address(es) from either SLAAC or DHCPv6, then it MUST create a global IPv6 address(es) from its delegated prefix(es) and configure those on one of its internal virtual network interfaces, unless configured to require a global IPv6 address on the WAN interface.

WAA-8: The CE router MUST support the SOL_MAX_RT option [RFC7083] and request the SOL_MAX_RT option in an Option Request Option (ORO).

WAA-9: As a router, the IPv6 CE router MUST follow the weak host (Weak End System) model [RFC1122]. When originating packets from an interface, it will use a source address from another one of its interfaces if the outgoing interface does not have an address of suitable scope.

WAA-10: The IPv6 CE router SHOULD implement the Information Refresh Time option and associated client behavior as specified in [RFC4242].

Prefix delegation requirements:

WPD-1: The IPv6 CE router MUST support DHCPv6 prefix delegation requesting router behavior as specified in [RFC3633] (Identity Association for Prefix Delegation (IA_PD) option).

WPD-2: The IPv6 CE router MAY indicate as a hint to the delegating router the size of the prefix it requires. If so, it MUST ask for a prefix large enough to assign one /64 for each of its interfaces, rounded up to the nearest nibble, and SHOULD be configurable to ask for more.

WPD-3: The IPv6 CE router MUST be prepared to accept a delegated prefix size different from what is given in the hint. If the delegated prefix is too small to address all of its interfaces, the IPv6 CE router SHOULD log a system management error. [RFC6177] covers the recommendations for service providers for prefix allocation sizes.

WPD-4: By default, the IPv6 CE router MUST initiate DHCPv6 prefix delegation when either the M or O flags are set to 1 in a received Router Advertisement (RA) message. Behavior of the CE router to use DHCPv6 prefix delegation when the CE router has not received any RA or received an RA with the M and the O bits set to zero is out of scope for this document.

WPD-5: Any packet received by the CE router with a destination address in the prefix(es) delegated to the CE router but not in the set of prefixes assigned by the CE router to the LAN must be dropped. In other words, the next hop for the prefix(es) delegated to the CE router should be the null destination. This is necessary to prevent forwarding loops when some addresses covered by the aggregate are not reachable [RFC4632].

(a) The IPv6 CE router SHOULD send an ICMPv6 Destination Unreachable message in accordance with Section 3.1 of [RFC4443] back to the source of the packet, if the packet is to be dropped due to this rule.

WPD-6: If the IPv6 CE router requests both an IA_NA and an IA_PD option in DHCPv6, it MUST accept an IA_PD option in DHCPv6 Advertise/Reply messages, even if the message does not contain any addresses, unless configured to only obtain its WAN IPv6 address via DHCPv6; see [RFC7550].

WPD-7: By default, an IPv6 CE router MUST NOT initiate any dynamic routing protocol on its WAN interface.

WPD-8: The IPv6 CE router SHOULD support the [RFC6603] Prefix Exclude option.

5.3. LAN-Side Configuration

The IPv6 CE router distributes configuration information obtained during WAN interface provisioning to IPv6 hosts and assists IPv6 hosts in obtaining IPv6 addresses. It also supports connectivity of these devices in the absence of any working WAN interface.

An IPv6 CE router is expected to support an IPv6 end-user network and IPv6 hosts that exhibit the following characteristics:

1. Link-local addresses may be insufficient for allowing IPv6 applications to communicate with each other in the end-user network. The IPv6 CE router will need to enable this communication by providing globally scoped unicast addresses or ULAs [RFC4193], whether or not WAN connectivity exists.

2. IPv6 hosts should be capable of using SLAAC and may be capable of using DHCPv6 for acquiring their addresses.

3. IPv6 hosts may use DHCPv6 for other configuration information, such as the DNS_SERVERS option for acquiring DNS information.

Unless otherwise specified, the following requirements apply to the IPv6 CE router’s LAN interfaces only.

ULA requirements:

ULA-1: The IPv6 CE router SHOULD be capable of generating a ULA prefix [RFC4193].

ULA-2: An IPv6 CE router with a ULA prefix MUST maintain this prefix consistently across reboots.

ULA-3: The value of the ULA prefix SHOULD be configurable.

ULA-4: By default, the IPv6 CE router MUST act as a site border router according to Section 4.3 of [RFC4193] and filter packets with local IPv6 source or destination addresses accordingly.

ULA-5: An IPv6 CE router MUST NOT advertise itself as a default router with a Router Lifetime greater than zero whenever all of its configured and delegated prefixes are ULA prefixes.

LAN requirements:

L-1: The IPv6 CE router MUST support router behavior according to Neighbor Discovery for IPv6 [RFC4861].

L-2: The IPv6 CE router MUST assign a separate /64 from its delegated prefix(es) (and ULA prefix if configured to provide ULA addressing) for each of its LAN interfaces.

L-3: An IPv6 CE router MUST advertise itself as a router for the delegated prefix(es) (and ULA prefix if configured to provide ULA addressing) using the "Route Information Option" specified in Section 2.3 of [RFC4191]. This advertisement is independent of having or not having IPv6 connectivity on the WAN interface.

L-4: An IPv6 CE router MUST NOT advertise itself as a default router with a Router Lifetime [RFC4861] greater than zero if it has no prefixes configured or delegated to it.

L-5: The IPv6 CE router MUST make each LAN interface an advertising interface according to [RFC4861].

L-6: In Router Advertisement messages ([RFC4861]), the Prefix Information option’s A and L flags MUST be set to 1 by default.

L-7: The A and L flags’ ([RFC4861]) settings SHOULD be user configurable.

L-8: The IPv6 CE router MUST support a DHCPv6 server capable of IPv6 address assignment according to [RFC3315] OR a stateless DHCPv6 server according to [RFC3736] on its LAN interfaces.

L-9: Unless the IPv6 CE router is configured to support the DHCPv6 IA_NA option, it SHOULD set the M flag to zero and the O flag to 1 in its Router Advertisement messages [RFC4861].

L-10: The IPv6 CE router MUST support providing DNS information in the DHCPv6 DNS_SERVERS and DOMAIN_LIST options [RFC3646].

L-11: The IPv6 CE router MUST support providing DNS information in the Router Advertisement Recursive DNS Server (RDNSS) and DNS Search List options. Both options are specified in [RFC6106].

L-12: The IPv6 CE router SHOULD make available a subset of DHCPv6 options (as listed in Section 5.3 of [RFC3736]) received from the DHCPv6 client on its WAN interface to its LAN-side DHCPv6 server.

L-13: If the delegated prefix changes, i.e., the current prefix is replaced with a new prefix without any overlapping time period, then the IPv6 CE router MUST immediately advertise the old prefix with a Preferred Lifetime of zero and a Valid Lifetime of either a) zero or b) the lower of the current Valid Lifetime and two hours (which must be decremented in real time) in a Router Advertisement message as described in Section 5.5.3, (e) of [RFC4862].

L-14: The IPv6 CE router MUST send an ICMPv6 Destination Unreachable message, code 5 (Source address failed ingress/egress policy) for packets forwarded to it that use an address from a prefix that has been invalidated.

L-15: The IPv6 CE router SHOULD provide HNCP (Home Networking Control Protocol) services, as specified in [RFC7788].

5.4. Transition Technologies Support

5.4.1. 464XLAT

464XLAT [RFC6877] is a simple and scalable technique to quickly deploy limited IPv4 access service to IPv6-only edge networks without encapsulation.

The CE router SHOULD support 464XLAT functionality. If 464XLAT is supported, it MUST be implemented according to [RFC6877]. The following CE Requirements also apply:

464XLAT requirements:

464XLAT-1: The IPv6 CE router MUST perform IPv4 Network Address Translation (NAT) on IPv4 traffic translated using the CLAT, unless a dedicated /64 prefix has been acquired using DHCPv6-PD [RFC3633].

464XLAT-2: The CE router MUST implement [RFC7050] in order to discover the PLAT-side translation IPv6 prefix. Alternatively MUST support draft-cui-intarea-464xlat-prefix-dhcp-00.

464XLAT-3: The CE router MUST implement a DNS proxy as described in [RFC5625].

464XLAT-4: The CE router MUST support the DHCPv4-over-DHCPv6 (DHCP 4o6) transport described in [RFC7341].

5.4.2. MAP-E

MAP-E [RFC7597] is a mechanism for transporting IPv4 packets across an IPv6 network using IP encapsulation, including a generic mechanism for mapping between IPv6 addresses and IPv4 addresses as well as transport-layer ports.

The CE router SHOULD support MAP-E functionality. If MAP-E is supported, it MUST be implemented according to [RFC7597]. The following CE Requirements also apply:

MAP-E requirements:

MAPE-1: The CE router MUST support configuration of MAP-E via the MAP-E DHCPv6 options [RFC7598]. The IPv6 CE router MAY use other mechanisms to configure MAP-E parameters. Such mechanisms are outside the scope of this document.

MAPE-2: The CE router MUST support the DHCPv6 S46 priority option described in [RFC8026].

MAPE-3: The CE router MUST support the DHCPv4-over-DHCPv6 (DHCP 4o6) transport described in [RFC7341].

MAPE-4: The IPv6 CE router MUST perform IPv4 Network Address Translation (NAT) on IPv4 traffic encapsulated using MAP-E.

5.4.3. MAP-T

MAP-T [RFC7599] is a mechanism similar to MAP-E, differing from it in that MAP-T uses IPv4-IPv6 translation, rather than encapsulation, as the form of IPv6 domain transport.

The CE router SHOULD support MAP-T functionality. If MAP-T is supported, it MUST be implemented according to [RFC7599]. The following CE Requirements also apply:

MAP-T requirements:

MAPT-1: The CE router MUST support configuration of MAP-T via the MAP-E DHCPv6 options [RFC7598]. The IPv6 CE router MAY use other mechanisms to configure MAP-E parameters. Such mechanisms are outside the scope of this document.

MAPT-2: The CE router MUST support the DHCPv6 S46 priority option described in [RFC8026].

MAPT-3: The CE router MUST support the DHCPv4-over-DHCPv6 (DHCP 4o6) transport described in [RFC7341].

MAPT-4: The IPv6 CE router MUST perform IPv4 Network Address Translation (NAT) on IPv4 traffic translated using MAP-T.

5.4.4. 6rd

6rd [RFC5969] specifies an automatic tunneling mechanism tailored to advance deployment of IPv6 to end users via a service provider's IPv4 network infrastructure. Key aspects include automatic IPv6 prefix delegation to sites, stateless operation, simple provisioning, and service that is equivalent to native IPv6 at the sites that are served by the mechanism. It is expected that such traffic is forwarded over the CE router's native IPv4 WAN interface and not encapsulated in another tunnel.

The CE router SHOULD support 6rd functionality. If 6rd is supported, it MUST be implemented according to [RFC5969]. The following CE Requirements also apply:

6rd requirements:

6RD-1: The IPv6 CE router MUST support 6rd configuration via the 6rd DHCPv4 Option 212. If the CE router has obtained an IPv4 network address through some other means such as PPP, it SHOULD use the DHCPINFORM request message [RFC2131] to request the 6rd DHCPv4 Option. The IPv6 CE router MAY use other mechanisms to configure 6rd parameters. Such mechanisms are outside the scope of this document.

6RD-2: If the IPv6 CE router is capable of automated configuration of IPv4 through IPCP (i.e., over a PPP connection), it MUST support user-entered configuration of 6rd.

6RD-3: If the CE router supports configuration mechanisms other than the 6rd DHCPv4 Option 212 (user-entered, TR-069 [TR-069], etc.), the CE router MUST support 6rd in "hub and spoke" mode. 6rd in "hub and spoke" requires all IPv6 traffic to go to the 6rd Border Relay. In effect, this requirement removes the "direct connect to 6rd" route defined in Section 7.1.1 of [RFC5969].

6RD-4: A CE router MUST allow 6rd and native IPv6 WAN interfaces to be active alone as well as simultaneously in order to support coexistence of the two technologies during an incremental transition period such as a transition from 6rd to native IPv6.

6RD-5: Each packet sent on a 6rd or native WAN interface MUST be directed such that its source IP address is derived from the delegated prefix associated with the particular interface from which the packet is being sent (Section 4.3 of [RFC3704]).

6RD-6: The CE router MUST allow different as well as identical delegated prefixes to be configured via each (6rd or native) WAN interface.

6RD-7: In the event that forwarding rules produce a tie between 6rd and native IPv6, by default, the IPv6 CE router MUST prefer native IPv6.

5.4.5. 6in4

6in4 [RFC4213] specifies a tunneling mechanism to allow end-users to manually configure IPv6 support via a service provider's IPv4 network infrastructure.

The CE router SHOULD support 6in4 functionality. If 6rd is implemented, 6in4 MUST be supported as well. If 6in4 is supported, it MUST be implemented according to [RFC4213]. The following CE Requirements also apply:

6in4 requirements:

6IN4-1: The IPv6 CE router SHOULD support 6in4 automated configuration by means of the 6rd DHCPv4 Option 212. If the CE router has obtained an IPv4 network address through some other means such as PPP, it SHOULD use the DHCPINFORM request message [RFC2131] to request the 6rd DHCPv4 Option. The IPv6 CE router MAY use other mechanisms to configure 6in4 parameters. Such mechanisms are outside the scope of this document.

6IN4-2: If the IPv6 CE router is capable of automated configuration of IPv4 through IPCP (i.e., over a PPP connection), it MUST support user-entered configuration of 6in4.

6IN4-3: If the CE router supports configuration mechanisms other than the 6rd DHCPv4 Option 212 (user-entered, TR-069 [TR-069], etc.), the CE router MUST support 6in4 in "hub and spoke" mode. 6in4 in "hub and spoke" requires all IPv6 traffic to go to the 6rd Border Relay. In effect, this requirement removes the "direct connect to 6rd" route defined in Section 7.1.1 of [RFC5969].

6IN4-4: A CE router MUST allow 6in4 and native IPv6 WAN interfaces to be active alone as well as simultaneously in order to support coexistence of the two technologies during an incremental transition period such as a transition from 6in4 to native IPv6.

6IN4-5: Each packet sent on a 6in4 or native WAN interface MUST be directed such that its source IP address is derived from the delegated prefix associated with the particular interface from which the packet is being sent (Section 4.3 of [RFC3704]).

6IN4-6: The CE router MUST allow different as well as identical delegated prefixes to be configured via each (6in4 or native) WAN interface.

6IN4-7: In the event that forwarding rules produce a tie between 6in4 and native IPv6, by default, the IPv6 CE router MUST prefer native IPv6.

5.4.6. Dual-Stack Lite (DS-Lite)

Dual-Stack Lite [RFC6333] enables both continued support for IPv4 services and incentives for the deployment of IPv6. It also de-couples IPv6 deployment in the service provider network from the rest of the Internet, making incremental deployment easier. Dual-Stack Lite enables a broadband service provider to share IPv4 addresses among customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) and Network Address Translation (NAT). It is expected that DS-Lite traffic is forwarded over the CE router’s native IPv6 WAN interface, and not encapsulated in another tunnel.

The IPv6 CE router SHOULD implement DS-Lite functionality. If DS-Lite is supported, it MUST be implemented according to [RFC6333]. This document takes no position on simultaneous operation of Dual-Stack Lite and native IPv4. The following CE router requirements also apply:

DS-Lite requirements:

DSLITE-1: The CE router MUST support configuration of DS-Lite via the DS-Lite DHCPv6 option [RFC6334]. The IPv6 CE router MAY use other mechanisms to configure DS-Lite parameters. Such mechanisms are outside the scope of this document.

DSLITE-2: The CE router MUST support the DHCPv6 S46 priority option described in [RFC8026].

DSLITE-3: The CE router MUST support the DHCPv4-over-DHCPv6 (DHCP 4o6) transport described in [RFC7341].

DSLITE-4: The IPv6 CE router MUST NOT perform IPv4 Network Address Translation (NAT) on IPv4 traffic encapsulated using DS-Lite.

DSLITE-5: If the IPv6 CE router is configured with an IPv4 address on its WAN interface, then the IPv6 CE router SHOULD disable the DS-Lite Basic Bridging BroadBand (B4) element.

5.4.7. Lightweight 4over6 (lw4o6)

Lw4o6 [RFC7596] specifies an extension to DS-Lite, which moves the NAPT function from the DS-Lite tunnel concentrator to the tunnel client located in the IPv6 CE router, removing the requirement for a CGN function in the tunnel concentrator and reducing the amount of centralized state.

The IPv6 CE router SHOULD implement lw4o6 functionality. If DS-Lite is implemented, lw4o6 MUST be supported as well. If lw4o6 is supported, it MUST be implemented according to [RFC7596]. This document takes no position on simultaneous operation of lw4o6 and native IPv4. The following CE router Requirements also apply:

Lw4o6 requirements:

LW6O4-1: The CE router MUST support configuration of lw4o6 via the lw4o6 DHCPv6 options [RFC7598]. The IPv6 CE router MAY use other mechanisms to configure lw4o6 parameters. Such mechanisms are outside the scope of this document.

LW6O4-2: The CE router MUST support the DHCPv6 S46 priority option described in [RFC8026].

LW6O4-3: The CE router MUST support the DHCPv4-over-DHCPv6 (DHCP 4o6) transport described in [RFC7341].

LW6O4-4: The IPv6 CE router MUST perform IPv4 Network Address Translation (NAT) on IPv4 traffic encapsulated using lw4o6.

LW6O4-5: If the IPv6 CE router is configured with an IPv4 address on its WAN interface, then the IPv6 CE router SHOULD disable the Lightweight Basic Bridging BroadBand (B4) element.

5.5. Security Considerations

It is considered a best practice to filter obviously malicious traffic (e.g., spoofed packets, "Martian" addresses, etc.). Thus, the IPv6 CE router ought to support basic stateless egress and ingress filters. The CE router is also expected to offer mechanisms to filter traffic entering the customer network; however, the method by which vendors implement configurable packet filtering is beyond the scope of this document.

Security requirements:

S-1: The IPv6 CE router SHOULD support [RFC6092]. In particular, the IPv6 CE router SHOULD support functionality sufficient for implementing the set of recommendations in [RFC6092], Section 4. This document takes no position on whether such functionality is enabled by default or mechanisms by which users would configure it.

S-2: The IPv6 CE router SHOULD support ingress filtering in accordance with BCP 38 [RFC2827]. Note that this requirement was downgraded from a MUST from RFC 6204 due to the difficulty of implementation in the CE router and the feature’s redundancy with upstream router ingress filtering.

S-3: If the IPv6 CE router firewall is configured to filter incoming tunneled data, the firewall SHOULD provide the capability to filter decapsulated packets from a tunnel.

6. Acknowledgements

This document is an update of RFC7084, whose original authors were: Hemant Singh, Wes Beebee, Chris Donley and Barbara Stark. The rest of the text on this section and the Contributors section, are the original acknowledgements and Contributors sections of the earlier version of this document.

Thanks to the following people (in alphabetical order) for their guidance and feedback:

Mikael Abrahamsson, Tore Anderson, Merete Asak, Rajiv Asati, Scott Beuker, Mohamed Boucadair, Rex Bullinger, Brian Carpenter, Tassos Chatzithomaoglou, Lorenzo Colitti, Remi Denis-Courmont, Gert Doering, Alain Durand, Katsunori Fukuoka, Brian Haberman, Tony Hain, Thomas Herbst, Ray Hunter, Joel Jaeggli, Kevin Johns, Erik Kline, Stephen Kramer, Victor Kuarsingh, Francois-Xavier Le Bail, Arifumi Matsumoto, David Miles, Shin Miyakawa, Jean-Francois Mule, Michael Newbery, Carlos Pignataro, John Pomeroy, Antonio Querubin, Daniel Roesen,

Hiroki Sato, Teemu Savolainen, Matt Schmitt, David Thaler, Mark Townsley, Sean Turner, Bernie Volz, Dan Wing, Timothy Winters, James Woodyatt, Carl Wuyts, and Cor Zwart.

This document is based in part on CableLabs' eRouter specification. The authors wish to acknowledge the additional contributors from the eRouter team:

Ben Bekele, Amol Bhagwat, Ralph Brown, Eduardo Cardona, Margo Dolas, Toerless Eckert, Doc Evans, Roger Fish, Michelle Kuska, Diego Mazzola, John McQueen, Harsh Parandekar, Michael Patrick, Saifur Rahman, Lakshmi Raman, Ryan Ross, Ron da Silva, Madhu Sudan, Dan Torbet, and Greg White.

7. Contributors

The following people have participated as co-authors or provided substantial contributions to this document: Ralph Droms, Kirk Erichsen, Fred Baker, Jason Weil, Lee Howard, Jean-Francois Tremblay, Yiu Lee, John Jason Brzozowski, and Heather Kirksey. Thanks to Ole Troan for editorship in the original RFC 6204 document.

8. References

8.1. Normative References

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <http://www.rfc-editor.org/info/rfc1122>.

[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>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <http://www.rfc-editor.org/info/rfc2131>.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <http://www.rfc-editor.org/info/rfc2464>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <http://www.rfc-editor.org/info/rfc2827>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

[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, <http://www.rfc-editor.org/info/rfc3633>.

[RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, DOI 10.17487/RFC3646, December 2003, <http://www.rfc-editor.org/info/rfc3646>.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <http://www.rfc-editor.org/info/rfc3704>.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, DOI 10.17487/RFC3736, April 2004, <http://www.rfc-editor.org/info/rfc3736>.

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

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

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, DOI 10.17487/RFC4213, October 2005, <http://www.rfc-editor.org/info/rfc4213>.

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, DOI 10.17487/RFC4242, November 2005, <http://www.rfc-editor.org/info/rfc4242>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, DOI 10.17487/RFC4443, March 2006, <http://www.rfc-editor.org/info/rfc4443>.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, DOI 10.17487/RFC4605, August 2006, <http://www.rfc-editor.org/info/rfc4605>.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <http://www.rfc-editor.org/info/rfc4632>.

[RFC4779] Asadullah, S., Ahmed, A., Popoviciu, C., Savola, P., and J. Palet, "ISP IPv6 Deployment Scenarios in Broadband Access Networks", RFC 4779, DOI 10.17487/RFC4779, January 2007, <http://www.rfc-editor.org/info/rfc4779>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC5072] Varada, S., Ed., Haskins, D., and E. Allen, "IP Version 6 over PPP", RFC 5072, DOI 10.17487/RFC5072, September 2007, <http://www.rfc-editor.org/info/rfc5072>.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, DOI 10.17487/RFC5625, August 2009, <http://www.rfc-editor.org/info/rfc5625>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <http://www.rfc-editor.org/info/rfc5905>.

[RFC5908] Gayraud, R. and B. Lourdelet, "Network Time Protocol (NTP) Server Option for DHCPv6", RFC 5908, DOI 10.17487/RFC5908, June 2010, <http://www.rfc-editor.org/info/rfc5908>.

[RFC5942] Singh, H., Beebee, W., and E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, DOI 10.17487/RFC5942, July 2010, <http://www.rfc-editor.org/info/rfc5942>.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, DOI 10.17487/RFC5969, August 2010, <http://www.rfc-editor.org/info/rfc5969>.

[RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <http://www.rfc-editor.org/info/rfc6092>.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, DOI 10.17487/RFC6106, November 2010, <http://www.rfc-editor.org/info/rfc6106>.

[RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", BCP 157, RFC 6177, DOI 10.17487/RFC6177, March 2011, <http://www.rfc-editor.org/info/rfc6177>.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <http://www.rfc-editor.org/info/rfc6333>.

[RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, DOI 10.17487/RFC6334, August 2011, <http://www.rfc-editor.org/info/rfc6334>.

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

[RFC6603] Korhonen, J., Ed., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, DOI 10.17487/RFC6603, May 2012, <http://www.rfc-editor.org/info/rfc6603>.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <http://www.rfc-editor.org/info/rfc6877>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <http://www.rfc-editor.org/info/rfc6887>.

[RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <http://www.rfc-editor.org/info/rfc7050>.

[RFC7083] Droms, R., "Modification to Default Values of SOL_MAX_RT and INF_MAX_RT", RFC 7083, DOI 10.17487/RFC7083, November 2013, <http://www.rfc-editor.org/info/rfc7083>.

[RFC7341] Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4-over-DHCPv6 (DHCP 4o6) Transport", RFC 7341, DOI 10.17487/RFC7341, August 2014, <http://www.rfc-editor.org/info/rfc7341>.

[RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596, July 2015, <http://www.rfc-editor.org/info/rfc7596>.

[RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <http://www.rfc-editor.org/info/rfc7597>.

[RFC7598] Mrugalski, T., Troan, O., Farrer, I., Perreault, S., Dec, W., Bao, C., Yeh, L., and X. Deng, "DHCPv6 Options for Configuration of Softwire Address and Port-Mapped Clients", RFC 7598, DOI 10.17487/RFC7598, July 2015, <http://www.rfc-editor.org/info/rfc7598>.

[RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <http://www.rfc-editor.org/info/rfc7599>.

[RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <http://www.rfc-editor.org/info/rfc7788>.

[RFC8026] Boucadair, M. and I. Farrer, "Unified IPv4-in-IPv6 Softwire Customer Premises Equipment (CPE): A DHCPv6-Based Prioritization Mechanism", RFC 8026, DOI 10.17487/RFC8026, November 2016, <http://www.rfc-editor.org/info/rfc8026>.

8.2. Informative References

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <http://www.rfc-editor.org/info/rfc6144>.

[RFC7157] Troan, O., Ed., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", RFC 7157, DOI 10.17487/RFC7157, March 2014, <http://www.rfc-editor.org/info/rfc7157>.

[RFC7550] Troan, O., Volz, B., and M. Siodelski, "Issues and Recommendations with Multiple Stateful DHCPv6 Options", RFC 7550, DOI 10.17487/RFC7550, May 2015, <http://www.rfc-editor.org/info/rfc7550>.

[TR-069] Broadband Forum, "CPE WAN Management Protocol", TR-069 Amendment 4, July 2011, <http://www.broadband-forum.org/technical/trlist.php>.

[UPnP-IGD] UPnP Forum, , "InternetGatewayDevice:2 Device Template Version 1.01", December 2010, <http://upnp.org/specs/gw/igd2/>.

Author’s Address

Jordi Palet Martinez
Consulintel, S.L.
Molino de la Navata, 75
La Navata - Galapagar, Madrid 28420
Spain

EMail: jordi.palet@consulintel.es
URI: http://www.consulintel.es/

Network
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2017

T. Pauly
D. Schinazi
Apple Inc.
March 12, 2017

An Update to Happy Eyeballs
draft-pauly-v6ops-happy-eyeballs-update-01

Abstract

"Happy Eyeballs" (RFC6555) is the name for a technique of reducing user-visible delays on dual-stack hosts. Since one address family (IPv4 or IPv6) may be blocked, broken, or sub-optimal on a network, clients that attempt connections for both address families in parallel have a higher chance of establishing a connection sooner. Now that this approach has been deployed at scale and measured for several years, the algorithm specification can be refined to improve its reliability and generalization.

Table of Contents

1. Introduction

"Happy Eyeballs" [RFC6555] is the name for a technique of reducing user-visible delays on dual-stack hosts. Since one address family (IPv4 or IPv6) may be blocked, broken, or sub-optimal on a network, clients that attempt connections for both address families in parallel have a higher chance of establishing a connection sooner. Now that this approach has been deployed at scale and measured for several years, the algorithm specification can be refined to improve its reliability and generalization.

This document recommends an algorithm of racing resolved addresses that has several stages of ordering and racing to avoid delays to the user whenever possible, while preferring the use of IPv6. Specifically, it discusses how to handle DNS queries when starting a connection on a dual-stack client, how to create an ordered list of addresses to which to attempt connections, and how to race the connection attempts.

1.1. Requirements Language

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

"Key words for use in RFCs to Indicate Requirement Levels" RFC 2119 [RFC2119].

2. Overview

This document defines a method of connection establishment, defined as "Happy Eyeballs Connection Setup". This approach has several distinct phases:

1. Initiation of asynchronous DNS queries [Section 3]

2. Sorting of resolved addresses [Section 4]

3. Initiation of asynchronous connection attempts [Section 5]

4. Establishment of one connection, which cancels all other attempts

Note that this document assumes that the host address preference policy favors IPv6 over IPv4. If the host is configured differently, the recommendations in this document can be easily adapted.

3. Hostname Resolution Query Handling

When a client has both IPv4 and IPv6 connectivity, and is trying to establish a connection with a named host, it needs to send out both AAAA and A DNS queries. Both queries SHOULD be made as soon after one another as possible, with the AAAA query made first, immediately followed by the A query.

Implementations MUST NOT wait for both families of answers to return before attempting connection establishment. If one query fails to return, or takes significantly longer to return, waiting for the second address family can significantly delay the connection establishment of the first one. Therefore, the client MUST treat DNS resolution as asynchronous. Note that if the platforms does not offer an asynchronous DNS API, this behavior can be simulated by making two separate synchronous queries on different threads, one per address family. If the AAAA query returns first, the first IPv6 connection attempt MUST be immediately started. If the A query returns first, the client SHOULD wait for a short time for the AAAA response. This delay will be referred to as the "Resolution Delay". The RECOMMENDED value for the Resolution Delay is 50 milliseconds. If the AAAA response is received within the Resolution Delay period, the client MUST immediately start the IPv6 connection attempt. If, at the end of the Resolution Delay period, the AAAA response has not been received but the A response has been received, the client SHOULD proceed to Sorting Addresses [Section 4] and staggered connection

attempts [Section 5] using only the IPv4 addresses returned so far. If the AAAA response arrives while these connection attempts are in progress, but before any connection has been established, then the newly received IPv6 addresses are incorporated into the list of available candidate addresses [Section 6] and the process of connection attempts will continue with the IPv6 addresses added, until one connection is established.

3.1. Handling Multiple DNS Server Addresses

If multiple DNS server addresses are configured for the current network, the client may have the option of sending its DNS queries over IPv4 or IPv6. In keeping with the Happy Eyeballs approach, queries SHOULD be sent over IPv6 first (note that this is not referring to the sending of AAAA or A queries, but rather the address of the DNS server itself). If DNS queries sent to the IPv6 address do not receive responses, that address may be marked as penalized, and queries can be sent to other DNS server addresses.

As native IPv6 deployments become more prevalent, and IPv4 addresses are exhausted, it is expected that IPv6 connectivity will have preferential treatment within networks. If a DNS server is configured to be accessible over IPv6, IPv6 should be assumed to be the preferred address family.

4. Sorting Addresses

Before attempting to connect to any of the resolved addresses, the client should define the order in which to start the attempts. Once the order has been defined, the client can use a simple algorithm for racing each option after a short delay [Section 5]. It is important that the ordered list involves all addresses from both families, as this allows the client to get the racing effect of Happy Eyeballs for the entire list, not just the first IPv4 and first IPv6 addresses.

First, the client MUST sort the addresses using Destination Address Selection ([RFC6724], Section 6).

If the client is stateful and has history of expected round-trip times (RTT) for the routes to access each address, it SHOULD add a Destination Address Selection rule between rules 8 and 9 that prefers addresses with lower RTTs. If the client keeps track of which addresses it has used in the past, it SHOULD add another destination address selection rule between the RTT rule and rule 9, which prefers used addresses over unused ones. This helps servers that use the client’s IP address for authentication, as is the case for TCP Fast Open ([RFC7413]) and some HTTP cookies. This historical data MUST

NOT be used across networks, and SHOULD be flushed on network changes.

Next, the client SHOULD modify the ordered list to interleave address families. Whichever address family is first in the list should be followed by an address of the other address family; that is, if the first address in the sorted list is IPv6, then the first IPv4 address should be moved up in the list to be second in the list. An implementation MAY want to favor one address family more by allowing multiple addresses of that family to be attempted before trying the other family. The number of contiguous addresses of the first address family will be referred to as the "First Address Family Count", and can be a configurable value.

5. Connection Attempts

Once the list of addresses has been constructed, the client will attempt to make connections. In order to avoid unreasonable network load, connection attempts SHOULD NOT be made simultaneously. Instead, one connection attempt to a single address is started first, followed by the others in the list, one at a time. Starting a new connection attempt does not affect previous attempts, as multiple connection attempts may occur in parallel. Once one of the connection attempts succeeds (generally when the TCP handshake completes), all other connections attempts that have not yet succeeded SHOULD be cancelled. Any address that was not yet attempted as a connection SHOULD be ignored.

A simple implementation can have a fixed delay for how long to wait before starting the next connection attempt. This delay is referred to as the "Connection Attempt Delay". One recommended value for this delay is 250 milliseconds. If the client has historical RTT data, it can also use the expected RTT to choose a more nuanced delay value. The recommended formula for calculating the delay after starting a connection attempt is: MAX(1.25 * RTT_MEAN + 4 * RTT_VARIANCE, 2 * RTT_MEAN), where the RTT values are based on the statistics for previous address used. If the TCP implementation leverages historical RTT data to compute SYN timeout, these algorithms should match so that a new attempt will be started at the same time as the previous is sending its second TCP SYN.

6. DNS Answer Changes during Happy Eyeballs Connection Setup

If, during the course of connection establishment, the DNS answers change either by adding resolved addresses, or removing previously resolved addresses (for example, due to expiry of the TTL on that DNS record), the client should react based on its current progress.

If an address is removed from the list that already had a connection attempt started, the connection attempt SHOULD NOT be cancelled, but rather be allowed to continue. If the removed address had not yet had a connection attempt started, it SHOULD be removed from the list of addresses to try.

If an address is added to the list, it should be sorted into the list of addresses not yet attempted according to the rules above (Section 4).

7. Summary of Configurable Values

The values that may be configured as defaults on a client for use in Happy Eyeballs are as follows:

- Resolution Delay (Section 3): The time to wait for a AAAA response after receiving an A response. RECOMMENDED at 50 milliseconds.

- First Address Family Count (Section 4): The number of addresses belonging to the first address family (such as IPv6) that should be attempted before attempting another address family. RECOMMENDED as 1, or 2 to more aggressively favor one address family.

- Connection Attempt Delay (Section 5): The time to wait between connection attempts in the absence of RTT data. RECOMMENDED at 250 milliseconds.

8. Security Considerations

This memo has no direct security considerations.

9. IANA Considerations

This memo includes no request to IANA.

10. Acknowledgments

The authors thank Dan Wing, Andrew Yourtchenko, and everyone else who worked on the original Happy Eyeballs design ([RFC6555]), Josh Graessley, Stuart Cheshire, and the rest of team at Apple that helped implement and instrument this algorithm, and Jason Fesler and Paul Saab who helped measure and refine this algorithm. The authors would also like to thank Nick Chettle, Paul Hoffman, Philip Homburg, Joe Touch and James Woodyatt for their input and contributions.

11. References

11.1. 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>.

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, DOI 10.17487/RFC6555, April 2012, <http://www.rfc-editor.org/info/rfc6555>.

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

11.2. Informative References

[RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014, <http://www.rfc-editor.org/info/rfc7413>.

Appendix A. Differences from RFC6555

"Happy Eyeballs: Success with Dual-Stack Hosts" [RFC6555] mostly concentrates on how to stagger connections to a hostname that has an AAAA and an A record. This document additionally discusses:

- how to perform DNS queries to obtain these addresses
- how to handle multiple addresses from each address family
- how to handle DNS updates while connections are being raced
- how to leverage historical information

Authors' Addresses

Tommy Pauly
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
US

Email: tpauly@apple.com

David Schinazi
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
US

Email: dschinazi@apple.com