

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2013

J. Arkko
A. Lindem
Ericsson
B. Paterson
Cisco Systems
October 23, 2012

Prefix Assignment in a Home Network
draft-arkko-homenet-prefix-assignment-03

Abstract

This memo describes a prefix assignment mechanism for home networks. It is expected that home gateway routers are allocated an IPv6 prefix through DHCPv6 Prefix Delegation (PD) or that a prefix is made available through other means. This prefix needs to be divided among the multiple subnets in a home network. This memo describes a mechanism for such division, or assignment, via OSPFv3. This is an alternative design to also using DHCPv6 PD for the assignment. The memo is input to the working group so that it can make a decision on which type of design to pursue. It is expected that a routing-protocol based assignment uses a minimal amount of prefixes.

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 April 26, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Requirements language	4
3. Role of Prefix Assignment	4
4. Router Behavior	5
4.1. Sending Router Advertisements	7
4.2. DNS Discovery	7
5. Design Choices	8
5.1. DNS Discovery	8
5.2. Prefix Assignment	8
6. Prefix Assignment in OSPFv3	9
6.1. Aggregated Prefix TLV	10
6.2. Assigned Prefix TLV	11
6.3. OSPFv3 Prefix Assignment	12
6.3.1. Making a New Assignment	15
6.3.2. Checking for Conflicts Across the Entire Network	15
6.3.3. Deprecating an Assigned Prefix	16
6.3.4. Verifying and Making a Local Assignment	16
7. ULA Generation	16
8. Hysteresis	18
9. Manageability Considerations	19
10. Security Considerations	19
11. IANA Considerations	19
12. Timer Constants	19
13. References	20
13.1. Normative References	20
13.2. Informative References	20
Appendix A. Changes in Version -02	21
Appendix B. Changes in Version -03	21
Appendix C. Acknowledgments	21
Authors' Addresses	21

1. Introduction

This memo describes a prefix assignment mechanism for home networks. It is expected that home gateway routers are allocated an IPv6 prefix through DHCPv6 Prefix Delegation (PD) [RFC3633], or that a prefix is made available by some other means. Manual configuration may be needed in some networks, for instance when the ISP does not support DHCPv6-based prefix delegation. In other cases, such as networks that have do not yet have an Internet connection, Unique Local Address (ULA) [RFC4193] prefixes can be automatically generated. For the purposes of this document, we refer to the prefix reserved for a home network as a prefix allocation.

A prefix allocation needs to be divided among the multiple subnets in a home network. For the purposes of this document, we refer to this process as prefix assignment. This memo describes a mechanism for prefix assignment via OSPFv3 [RFC5340].

The OSPv3-based mechanism is an alternative design to also using DHCPv6 PD for the prefix assignment in the internal network. This memo has been written so that the working group can make a decision on which type of design to pursue. The main benefit of using a routing protocol to handle the prefix assignment is that it can provide a more efficient use of address space than hierarchical assignment through DHCPv PD. This may be important for home networks that only get a /60 prefix allocation from their ISPs.

The rest of this memo is organized as follows. Section 2 defines the usual keywords, Section 3 explains the main requirements for prefix assignments, Section 4 describes how a home gateway router makes assignments when it itself has multiple subnets, and Section 5 and Section 6 describe how the assignment can be performed in a distributed manner via OSPFv3 in the entire home network. Finally, Section 7 specifies the procedures for automatic generation of ULA prefixes, Section 8 explains the hysteresis principles applied to prefix assignment and de-assignment, Section 9 explains what administrative interfaces are useful for advanced users that wish to manually interact with the mechanisms, Section 10 discusses the security aspects of the design, Section 11 explains the necessary IANA actions, and Section 12 defines the necessary timer constants.

An analysis of a mechanism reminiscent of the one described in this specification has been published in the SIGCOMM IPv6 Workshop [SIGCOMM.IPV6]. Further analysis is encouraged.

2. Requirements language

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

3. Role of Prefix Assignment

Given a prefix shorter than /64 for the entire home network, this prefix needs to be subdivided so that every subnet is given its own /64 prefix. In many cases there will be just one subnet, the internal network interface attached to the router. But it is also common to have two or more internal network interfaces with intentionally separate networks. For instance, "private" and "guest" SSIDs are automatically configured in many current home network routers. When all the network interfaces that require a prefix are attached to the same router, the prefix assignment problem is simple, and procedures outlined in Section 4 can be employed.

In a more complex setting there are multiple routers in the internal network. There are various possible reasons why this might be necessary [I-D.ietf-homenet-arch]. For instance, networks that cannot be bridged together should be routed, speed differences between wired and wireless interfaces make the use of the same broadcast domain undesirable, or simply that router devices keep being added. In any case, it then becomes necessary to assign prefixes across the entire network, and this assignment can no longer be done on a local basis as proposed in Section 4. A distributed mechanism and a protocol are required.

The key requirements for this distributed mechanism are as follows.

- o A prefix allocated to a home gateway router within the home network is used to assign /64 prefixes on each subnet that requires one.

Note that several methods may be used to allocate such an aggregated prefix.

- o The assignment mechanism should provide reasonable efficiency. As a practical benchmark, some ISPs may employ /60 allocations to individual subscribers. As a result, the assignment mechanism should avoid wasting too many prefixes so that this set of 16 /64 prefixes is not exhausted in the foreseeable future for commonly occurring network configurations.

- o In particular, the assignment of multiple prefixes to the same network from the same top-level prefix must be avoided.

Example: When a home network consists of a home gateway router connected to another router which in turn is connected to hosts, a minimum of two /64 prefixes are required in the internal network: one between the two routers, and another one for the host-side interface on the second router. But an ineffective assignment mechanism in the two routers might have both of them asking for separate assignments for this shared interface.

- o The assignments must be stable across reboots, power cycling, router software updates, and preferably, should be stable across simple network changes. Simple network changes are in this case defined as those that could be resolved through either deletion or addition of a prefix assignment. For instance, the addition of a new router without changing connections between existing routers requires just the assignment of new prefixes for the new networks that the router introduces. There are no stability requirements across more complex types of network reconfiguration events. For instance, if a network is separated into two networks connected by a newly inserted router, this may lead to renumbering all networks within the home.

In an even more complex setting there may be multiple home gateway routers and multiple connections to ISP(s). These cases are analogous to the case of a single gateway router. Each gateway will simply distribute the prefix it has, and participating routers throughout the network may assign themselves prefixes from several gateways. Multiple assignments can be made for the same interface. For example, this can be useful in a multi-homing setting.

Similarly, it is also possible that it is necessary to assign either a global prefix delegated from the ISP or a local, Unique Local Address (ULA) prefix [RFC4193]. The mechanisms in this memo are applicable to both types of prefixes. The details of the generation of ULA-based prefixes is covered in Section 7.

The mechanisms in this memo can also be used in standalone or ad hoc networks where no global prefixes or Internet connectivity are available, by distributing ULA prefixes within the network.

4. Router Behavior

This section describes how a router assigns prefixes to its directly connected interfaces. We assume that the router has prefix

allocation(s) that it can use for this assignment. Each such prefix allocation is called an aggregated prefix. Parts of the aggregated prefix may already be assigned for some purpose; a coordinated assignment from the prefix is necessary before it can actually be assigned to an interface.

Even if the assignment process is local, it still needs to follow the requirements from Section 3. This is achieved through the following rules:

- o The router MUST maintain a list of assigned prefixes on a per-interface basis. The contents of this list consists of prefixes that the router itself has assigned to the interface, as well as prefixes assigned to the interface by other routers. The latter are learned through the mechanisms described in Section 6, when they are used. Each prefix is associated with the Router ID of the router that assigned it.
- o Whenever the router finds a combination of an interface and aggregated prefix that is not used on the interface, it SHOULD make a new prefix assignment. That is, the router checks to see if an interface and aggregated prefix exists such that there are no assigned prefixes within that interface that are more specific than the aggregated prefix. In this situation the router makes an allocation from the aggregated prefix (if possible) and adds the assignment to the list of assigned prefixes on that interface.

Note: The above implies that when there are multiple aggregated prefixes, each network will be assigned multiple prefixes.

- o An assignment from an aggregated prefix MUST be checked against possible other assignments from the same aggregated prefix on the same link by neighboring routers, to avoid unnecessary assignments. Assignments MUST also be examined against all existing assignments from the same aggregated prefix across the network, to avoid collisions. Assignments are made for individual /64 prefixes. The choice of a /64 prefix among multiple free ones MUST be made randomly or based on an algorithm that takes unique hardware characteristics of the router and the interface into account. This helps avoid collisions when simultaneous assignments are made within a network.
- o In order to provide a stable assignment, the router MUST store assignments affecting directly connected interfaces and automatically generated ULA prefixes in non-volatile memory and attempt to re-use them in the future when possible. At least the 5 most recent assignments SHOULD be stored. Note that this applies to both its own assignments as well as assignments made by

others. This ensures that the same prefix assignments are made regardless of the order that different devices are brought up. To avoid attacks on flash memory write cycles, assignments made by others SHOULD be recorded only after 10 minutes have passed and the assignment is still valid.

- o Re-using a memorized assignment is possible when a aggregated prefix exists that is less specific than the prefix in the assignment (or it is the prefix itself in the assignment), and the prefix is currently unassigned.

4.1. Sending Router Advertisements

Once the router has assigned a prefix to an interface, it MUST act as a router as defined in [RFC4861] and advertise the prefix in Router Advertisements. The lifetime of the prefix SHOULD be advertised as a reasonably long period, at least 48 hours or the lifetime of the assigned prefixes, whichever is smaller.

4.2. DNS Discovery

To support a variety of IPv6-only hosts in these networks, the router needs to ensure that sufficient DNS discovery mechanisms are enabled. It is RECOMMENDED that both stateless DHCPv6 [RFC3736] and Router Advertisement options [RFC6106] are supported and turned on by default in routers.

The above requires, however, that a working DNS server is known and addressable via IPv6. The mechanism in [RFC3736] and [RFC3646] can be used for this. It is RECOMMENDED that each router attempts to discover an existing DNS server. Typically, such a server will be provided by an ISP. However, in some cases no such server can be found. For instance, an ISP may provide only IPv4 DNS server addresses, or a router deep within the home network is unaware of the IPv6 DNS servers that a home gateway router has discovered. In these situations it is RECOMMENDED that each router turns on a local DNS relay that fetches information from the IPv4 Internet (if a working IPv4 DNS server is available) or a full DNS server that fetches information from the DNS root.

As a result of these recommendations, as long as there is reachability to at least the Internet, every router within the home network will either know the IPv6 address of a DNS server or it itself runs a server that can fetch information from the Internet. As a result, the router can provide information about the server address to hosts in directly connected networks.

5. Design Choices

5.1. DNS Discovery

The DNS discovery recommendations in Section 4.2 ensure that an IPv6-only home network can resolve names. However, these recommendations are suboptimal in the sense that different routers in the home may provide different DNS servers, or multiple local DNS servers have to be run where it would have been possible to point to one, or even point to the one provided by the ISP. However, better coordination for the DNS server selection would require some form of additional communication between the routers in the home network. The authors solicit opinions from the Working Group on whether this is something that should be specified. However, the current design is easy to deploy even when not all routers within the network support Homenet specifications yet; the mechanism provides an incremental improvement to IPv6 DNS reachability even when the first Homenet router is deployed.

5.2. Prefix Assignment

The OSPFv3-based prefix assignment protocol needs to detect two types of conflicts:

1. Two or more OSPFv3 routers have assigned the same IPv6 prefix for different networks.
2. Two or more OSPFv3 routers have assigned different IPv6 prefixes for the same network.

Several design decisions were needed to construct the protocol:

1. How to determine the winner in case of a conflict?

The algorithm in Section 6 ensures that the OSPFv3 Router with the numerically lower OSPFv3 Router ID removes its assignment and schedules an advertisement of LSAs that no longer describe such an assignment. That is, the router with the highest Router ID wins in a conflict situation.

2. How to ensure that a network-wide conflict can be detected?

We chose to define new LSA extensions -- TLVs within the new Autoconfiguration LSA -- that are flooded throughout the network. Another possible design would have been to re-use existing OSPFv3 LSAs, and by assuming that if a router advertises a prefix then it has made an assignment. The advantage of the design that we chose is that we get to specify what information is needed in the

new TLVs. This is particularly important, as not all existing OSPFv3 LSAs are extensible. A downside is that assignments will not be visible, unless the router using an assignment implements this specification and advertises the new LSAs. Had we reused existing LSAs, a manual assignment for a legacy router could have been handled, as the legacy router would have advertised the prefix assigned to it.

3. How to ensure that both local and network-wide conflicts can be detected?

We chose to employ the same new Autoconfiguration LSA TLVs for this purpose, and correlate neighbors through the Router IDs and Interface IDs that they advertise in these TLVs. The OSPFv3 Router with a numerically lower OSPFv3 Router ID should accept the global IPv6 prefix from the neighbor with the highest OSPFv3 Router ID.

6. Prefix Assignment in OSPFv3

This section describes how prefix assignment in a home network can be performed in a distributed manner via OSPFv3. It is expected that the router already support the auto-configuration extensions defined in [I-D.ietf-ospf-ospfv3-autoconfig].

An overview of OSPFv3-based prefix assignment is as follows. OSPFv3 routers that are capable of auto-configuration advertise an OSPFv3 Auto-Configuration (AC) LSA [I-D.ietf-ospf-ospfv3-autoconfig] with suitable TLVs. For prefix assignment, two TLVs are used. The Aggregated Prefix TLV (Section 6.1) advertises an aggregated prefix, usually the prefix that has been delegated to the home gateway router from the ISP through DHCPv6 PD. These aggregated prefixes are necessary for running the algorithm in Section 4 for determining whether prefix assignments can and should be made.

The Assigned Prefix TLV (Section 6.2) is used to communicate assignments that routers make out of the aggregated prefixes.

An assignment can be made when the algorithm in Section 4 indicates that it can be made and no other router has claimed the same assignment. The router makes an OSPFv3 advertisement with the Assigned Prefix TLV included to let other devices know that the prefix is now in use. Unfortunately, collisions are still possible, when the algorithms on different routers happen to choose the same free /64 prefix or when more /64 prefixes are needed than are available. This situation is detected through an advertisement where a different router claims the assignment of the same prefix. In this

situation the router with the numerically lower OSPFv3 Router ID has to select another prefix and immediately withdraw any assignments and advertisements that may have been advertised in OSPFv3. See also Section 5.2 in [I-D.ietf-ospf-ospfv3-autoconfig].

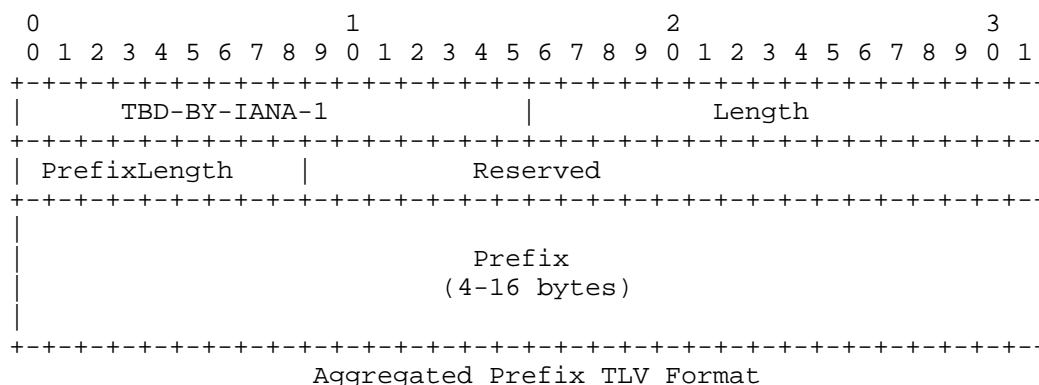
6.1. Aggregated Prefix TLV

The Aggregated Prefix TLV is defined for the OSPFv3 Auto-Configuration (AC) LSA [I-D.ietf-ospf-ospfv3-autoconfig]. It will have type TBD-BY-IANA-1 and MUST be advertised in the LSID OSPFv3 AC LSA with an LSID of 0. It MAY occur once or multiple times and the information from all TLV instances is retained. The length of the TLV is variable.

The contents of the TLV include an aggregated prefix (Prefix) and prefix length (PrefixLength). PrefixLength is the length in bits of the prefix and is an 8-bit field. The PrefixLength MUST be greater than or equal to 8 and less than or equal to 64. The prefix describes an allocation of a global or ULA prefix for the entire auto-configured home network. The Prefix is an encoding of the prefix itself as an even multiple of 32-bit words, padding with zero bits as necessary. This encoding consumes $(\text{PrefixLength} + 31) / 32$ 32-bit words and is consistent with [RFC5340]. It MUST NOT be directly assigned to any interface before following the procedures defined in this memo.

This TLV SHOULD be advertised by every home gateway router that has either a manual, DHCPv6 PD-based, or generated ULA prefix that is shorter than /64.

This TLV MUST appear inside an OSPFv3 Router Auto-Configuration LSA, and only in combination with the Router-Hardware-Fingerprint TLV [I-D.ietf-ospf-ospfv3-autoconfig] Section 5.2.2 in the same LSA.



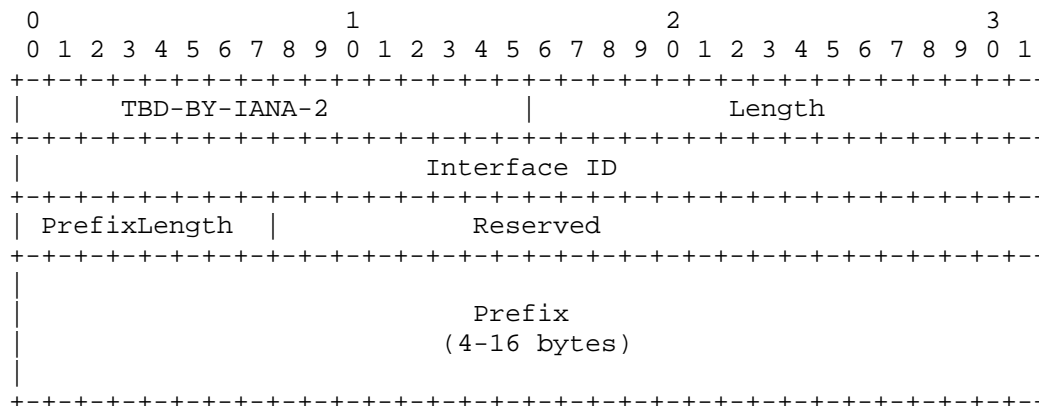
6.2. Assigned Prefix TLV

The Assigned Prefix TLV is defined for the OSPFv3 Auto-Configuration (AC) LSA [I-D.ietf-ospf-ospfv3-autoconfig]. It will have type TBD-BY-IANA-2 and MUST be advertised in the LSID OSPFv3 AC LSA with an LSID of 0. It MAY occur once or multiple times and the information from all TLV instances is retained. The length of the TLV is variable.

The contents of the TLV include an Interface ID, assigned prefix (Prefix), and prefix length (PrefixLength). The Interface ID is the same OSPFv3 Interface ID that is described in section 4.2.1 or [RFC5340]. PrefixLength is the length in bits of the prefix and is an 8-bit field. The PrefixLength value MUST be 64 in this version of the specification. The prefix describes an assignment of a global or ULA prefix for a directly connected interface in the advertising router. The Prefix is an encoding of the prefix itself as an even multiple of 32-bit words, padding with zero bits as necessary. This encoding consumes $(\text{PrefixLength} + 31) / 32$ 32-bit words and is consistent with [RFC5340].

This TLV MUST be advertised by the router that has made assignment from an aggregated prefix per Section 4.

This TLV MUST appear inside an OSPFv3 Router Auto-Configuration LSA, and only in combination with the Router-Hardware-Fingerprint TLV [I-D.ietf-ospf-ospfv3-autoconfig] Section 5.2.2 in the same LSA.



Assigned Prefix TLV Format

6.3. OSPFv3 Prefix Assignment

OSPFv3 Routers supporting the mechanisms in the memo will learn or assign a global /64 IPv6 prefix for each IPv6 interface. Since the mechanisms described herein are based on OSPFv3, Router ID assignment as described in [I-D.ietf-ospf-ospfv3-autoconfig] MUST have completed successfully.

When an OSPFv3 Router receives a global prefix through DHCPv6 prefix delegation, manual configuration, or other means, it SHOULD advertise this prefix by including the Aggregated Prefix TLV in its OSPFv3 AC LSA. This will trigger prefix assignment for auto-configured OSPFv3 routers within the routing domain including the originating OSPFv3 router.

Discussion: Note that while having multiple routers advertise the same aggregated address space (or address space that covers another router's aggregated address space) is a configuration error, it should not result in any adverse effects, as long as assignments from such space are still checked for collisions against all other assignments from the same address space.

When an OSPFv3 Router detects a change in the set of AC LSAs in its LSA database, it will run the prefix assignment algorithm. The purpose of this algorithm is to determine, for each Aggregated Prefix in the database, whether or not a new prefix needs to be assigned for each of its attached IPv6 interfaces and whether or not existing assignments need to be deprecated. The algorithm also detects and removes assignments for which there is no longer a corresponding Aggregated Prefix. Before the algorithm is run, all existing assignments in assigned prefix lists for directly connected interfaces must be marked as "invalid" and will be deleted at the end of the algorithm if they are still in this state. An assigned prefix is considered to be "valid" if all the following conditions are met:

- o A containing Aggregated Prefix TLV exists in reachable AC LSA(s).
- o An Assigned Prefix TLV that matches this assignment exactly (same prefix, same router and interface ID associated with the assignment) exist in reachable AC LSA(s).
- o Any router advertising an assignment for the same link and Aggregated Prefix has a lower Router ID than the source of this assignment.
- o If this router is the source of the assignment, any router in the network that has assigned the same prefix on a different link has a lower Router ID than this router.

Note that this definition of a "valid assignment" depends on the router running the algorithm: in particular, a router is not expected to detect prefix collisions or duplicate prefix assignments that do not concern assignments for which it is the responsible router. It is the role of the responsible router to detect these cases and update its AC LSAs accordingly. A router is, however, expected to react to these updates from other routers which translate into additions or removals of Aggregated Prefix or Assigned Prefix TLVs.

The router is expected to have made a snapshot of the LSA database before running this algorithm. The prefix assignment algorithm consists of the following steps run once per combination of Aggregated Prefix in the LSA database and directly connected OSPFv3 interface. For the purposes of this discussion, the Aggregated Prefix will be referred to as the Current Aggregated Prefix, and the interface will be referred to as the Current Interface. The following steps will be performed for each tuple (Aggregated Prefix, OSPFv3 interface):

1. The OSPFv3 Router will search all AC LSAs for an Aggregated Prefix TLV describing a prefix which contains but is not equal to the Current Aggregated Prefix. If such a prefix is found, the algorithm is skipped for the Current Aggregated Prefix as it either has or will be run for the shorter prefix.
2. The OSPFv3 router will examine its list of neighbors to find all neighbors in state greater than Init (these neighbors will be referred to as active neighbors).
3. The following three steps will serve to determine whether an assignment needs to be made on the link:

i.

The OSPFv3 router will determine whether or not it has the highest Router ID of all active OSPFv3 routers on the link.

ii.

If OSPFv3 active neighbors are present on the link, the router will determine whether any of them have already assigned an IPv6 prefix. This is done by examining the AC LSAs of all the active neighbors on the link and looking for any that include an Assigned Prefix TLV with the same OSPFv3 Router ID and Interface ID as the neighbor has. If one is found and it is a subnet of the IPv6 prefix advertised in the Aggregated Prefix TLV, the router stores this prefix and the Router ID of the router advertising it for reference in the next step. If

several such prefixes are found, only the prefix and Router ID with the numerically highest Router ID are stored.

iii.

The router will compare its Router ID with the highest Router ID among neighbors which have made an assignment on the link. If it is higher (or if no assignments have been made by any neighbors), it will determine whether or not it is already the source of an assignment for the Current Interface from the Current Aggregated Prefix.

4. There are four possibilities at this stage:

- * The router has already made an assignment on the link and has a higher Router ID than all eventual neighbors which have also made an assignment. In this case, the router's existing assignment takes precedence over all other eventual existing assignments on the link, but the router must determine whether its assignment is still valid throughout the whole network. This is described in Section 6.3.2.
- * An assignment has been made by a neighbor on the link, and the router either has not made an assignment on the link, or has a lower Router ID than the neighbor. In this case, the neighbor's assignment takes precedence over all eventual existing assignments on the link (including assignments made by the router), and the router must update the assigned prefix list of the Current Interface as well as check assignments on other interfaces for potential collisions. This is described in Section 6.3.4.
- * No assignment has been made by anyone on the link, and the router has the highest Router ID on the link. In this case, it must make an assignment from the Current Aggregated Prefix. This is described in Section 6.3.1.
- * No assignment has been made by anyone on the link, and the router does not have the highest Router ID on the link. In this case, the algorithm exits as the router is not responsible for prefix assignment on the link.

Once the algorithm has been run for each Aggregated Prefix and each interface, the router must delete all assignments that are not marked as valid on all assigned prefix lists and deprecate the corresponding addresses. If this leads to deleting an assignment that this router was responsible for, or if AC LSA origination was scheduled during the algorithm, it must originate a new AC LSA advertising the

changes. The router MUST also deprecate deleted prefixes as specified in Section 6.3.3.

6.3.1. Making a New Assignment

This procedure is executed when no assignment exists on the link and the router is responsible for making an assignment. The router MUST:

1. Examine all the AC LSAs not advertised by this router that include Assigned Prefix TLVs that are subnets of the Current Aggregated Prefix, as well as all assignments made by this router, to determine which prefixes are already assigned.
2. Examine former prefix assignments stored in non-volatile storage for the interface. Starting with the most recent assignment, if the prefix is both a subnet of the Current Aggregated Prefix and is currently unassigned, reuse the assignment for the interface.
3. If no unused former prefix assignment is found, and an unassigned /64 subnet of the Current Aggregated Prefix exists, assign that prefix to the interface.
4. If no OSPFv3 neighbors have been discovered and previous prefix assignments exist, the router can make the assignments immediately. Otherwise, the hysteresis periods specified in Section 8 are applied before making an assignment.
5. In the event that no assignment could be made to the interface, a warning must be raised. However, the router MUST remain in a state where it continues to assign prefixes through OSPFv3, as prefixes may later become available.
6. Once a global IPv6 prefix is assigned, the router will mark it as valid and schedule re-origination of the AC LSA including the Assigned Prefix TLV once all Aggregated Prefixes and interfaces have been examined.

6.3.2. Checking for Conflicts Across the Entire Network

This procedure is executed for every assignment that the router intends to make or retain as the router responsible for an assignment.

The router MUST verify that this assignment is still valid across the whole network. This assigned prefix will be referred to as the Current Assigned Prefix. The router will search for a reachable AC LSA in the LSA database that is advertised by a router with a higher Router ID and contains an Assigned Prefix equal to the Current

Assigned Prefix. If such an LSA is found, it needs to be deprecated as described in Section 6.3.3. Otherwise, the router will mark its assignment as valid.

6.3.3. Deprecating an Assigned Prefix

This procedure is executed when the router's earlier assignment of a prefix can no longer be used. The following steps MUST be followed:

1. If the the prefix was in an interface's assigned prefix list, it is removed.
2. If this router was the source of the prefix assignment, schedule re-origination of the modified AC LSA once the algorithm has finished.
3. The router MUST also deprecate the prefix, if it had been advertised in Router Advertisements on an interface. The prefix is deprecated by sending Router Advertisements with the lifetime set to 0 [RFC4861] for the prefix in question.

6.3.4. Verifying and Making a Local Assignment

This procedure is executed when an assignment by a neighbor already exists, and takes precedence over all other assignments on the link. The router must check whether or not it is already aware of this assignment. It will search for the assigned prefix matching the neighbor's assignment and Router ID in the Current Interface's assigned prefix list. If it is already present, the router will mark it as valid. Otherwise, the router will check that no assignment on any directly connected interface collides with the neighbor's assignment. If a collision is found and the colliding prefix takes priority over the neighbor's assignment (higher Router ID), the router will silently ignore the neighbor's assignment. If a collision is found but the neighbor's assignment takes priority, the old assignment is removed as described in Section 6.3.3. If the neighbor's assignment takes priority, or if no collision was found, the router will provision the interface with the prefix, add it to the list of assigned prefixes using the neighbor's Router ID and mark it as valid.

7. ULA Generation

For ULA-based prefixes, it is necessary to elect a router as the generator of such prefixes, have it perform the generation, and then employ the prefixes for local interfaces and the entire router network. This section specifies these procedures, and recommends the

generation of ULAs when no connectivity can be established otherwise. However, the use of ULAs in parallel with global IPv6 prefixes is outside the scope of this memo. The mechanisms in this memo could be used for that as well.

When an OSPFv3 Router detects a change in the set of AC LSAs in its LSA database, it will run the ULA generation algorithm. The purpose of this algorithm is to determine whether a new ULA prefix needs to be generated. There is no need for this router to generate a new ULA prefix when any of the following conditions are met:

i.

An Aggregated Prefix TLV exists in an AC LSA advertised by a reachable router in the LSA database, with either global or ULA address space.

ii.

A reachable router in the OSPFv3 topology with a higher Router ID than this OSPFv3 router exists.

iii.

This router has assignments from either IPv4 or IPv6 global address space on any interface, or there is connectivity to the global Internet.

Discussion: This rule is necessary in order to prevent autoconfiguration-capable routers from unnecessarily creating ULA address space in networks where autoconfiguration is not in use. Similarly, from an IPv6 "happy eyeballs" perspective it is desirable to not create local islands of IPv6 connectivity when there is IPv4 connectivity (even through a NAT).

If none of the above conditions are met after applying the hysteresis principles from Section 8, the router SHOULD perform the following actions:

1. Generate a new 48-bit ULA prefix as specified in [RFC4193], Section 3.2.
2. Record the new prefix in stable storage, per rules in Section 4.
3. Advertise the new prefix allocation in OSPFv3 as specified in Section 6.3.

4. Assign /64 prefixes from the new prefix for its own use, as a part of the general algorithm for making prefix assignments (also in Section 6.3).

If the router has made such an allocation, it SHOULD continue to advertise the prefix in OSPFv3 for as long as conditions i) through iii) do not apply, with the exception of the generated ULA prefix that this router is advertising.

If the router has made such an allocation, and any of the conditions become true (except for the case of the ULA prefix that the router is advertising) even after applying the hysteresis principles from Section 8, then the OSPFv3 router SHOULD withdraw the advertisement for the aggregated prefix. This is done by scheduling the re-origination of an AC LSA that does not include the Aggregated Prefix TLV with the ULA. Note that as a result of the general algorithm for making prefix assignments, any /64 prefix assignments from the ULA prefix will eventually be deprecated.

8. Hysteresis

A network may experience temporary connectivity problems, routing protocol convergence may take time, and a set of devices may be coming up at the same time due to power being turned on in a synchronous manner. Due to these reasons it is important that the prefix allocation and assignment mechanisms do not react before the situation is allowed to stabilize. To allow for this, a hysteresis principle is applied to new or withdrawn automatically generated prefixes and prefix assignments.

A new automatically generated ULA prefix SHOULD NOT be allocated before the router has waited NEW_ULA_PREFIX seconds for another prefix or reachable OSPFv3 router to appear. See Section 12 for the specific time value.

A previously automatically generated ULA prefix SHOULD NOT be taken out of use before the router has waited TERMINATE_ULA_PREFIX seconds.

A new prefix assignment within an aggregated prefix SHOULD NOT be committed before the router has waited NEW_PREFIX_ASSIGNMENT seconds for another prefix or reachable OSPFv3 router to appear. Note the exceptions to this rule in Section 6.3.1, item 4.

A previously assigned prefix SHOULD NOT be taken out of use before the router has waited TERMINATE_PREFIX_ASSIGNMENT seconds.

9. Manageability Considerations

Advanced users may wish to manage their networks without automation, and there may also be situations where manual intervention may be needed. For these purposes there **MUST** be a configuration mechanism that allows users to turn off the automatic prefix allocation and assignment on a given interface. This setting can be a part of disabling the entire routing auto-configuration [I-D.ietf-ospf-ospfv3-autoconfig].

In addition, there **SHOULD** be a configuration mechanism that allows users to specify the prefix that they would like the router to request for a given interface. This can be useful, for instance, when a router is replaced and there is a desire for the new router to be configured to ask for the same prefix as the old one, in order to avoid renumbering other devices on this network.

Finally, there **SHOULD** be mechanisms to display the prefixes assigned on each interface, and where they came from (manual configuration, DHCPv6 PD, OSPFv3).

10. Security Considerations

Security can be always added later.

11. IANA Considerations

This memo makes two allocations out of the OSPFv3 Auto- Configuration (AC) LSA TLV namespace [I-D.ietf-ospf-ospfv3-autoconfig]:

- o The Aggregated Prefix TLV in Section 6.1 takes the value TBD-BY-IANA-1 (suggested value is 2).
- o The Assigned Prefix TLV in Section 6.2 takes the value TBD-BY-IANA-2 (suggested value is 3).

12. Timer Constants

NEW_ULA_PREFIX	20 seconds
TERMINATE_ULA_PREFIX	120 seconds
NEW_PREFIX_ASSIGNMENT	20 seconds
TERMINATE_PREFIX_ASSIGNMENT	240 seconds

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.
- [RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.
- [I-D.ietf-ospf-ospfv3-autoconfig]
Lindem, A. and J. Arkko, "OSPFv3 Auto-Configuration",
draft-ietf-ospf-ospfv3-autoconfig-00 (work in progress),
October 2012.

13.2. Informative References

- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil,
"Home Networking Architecture for IPv6",
draft-ietf-homenet-arch-06 (work in progress),
October 2012.
- [I-D.chelius-router-autoconf]
Chelius, G., Fleury, E., and L. Toutain, "Using OSPFv3 for
IPv6 router autoconfiguration",
draft-chelius-router-autoconf-00 (work in progress),
June 2002.

[I-D.dimitri-zospf]

Dimitrelis, A. and A. Williams, "Autoconfiguration of routers using a link state routing protocol", draft-dimitri-zospf-00 (work in progress), October 2002.

[SIGCOMM.IPV6]

Chelius, G., Fleury, E., Sericola, B., Toutain, L., and D. Binet, "An evaluation of the NAP protocol for IPv6 router auto-configuration", ACM SIGCOMM IPv6 Workshop, Kyoto, Japan, 2007.

Appendix A. Changes in Version -02

These changes were extensive, including the definition of a new algorithm for making allocations, adding support for DNS server discovery, adding support for ULA-based address space generation, and adding specifications for a hysteresis mechanism.

Appendix B. Changes in Version -03

This version updated references to the most current ones, and changed the "usable prefix" terminology to "aggregated prefix". The requirements for turning on DNS relays or servers were also clarified.

Appendix C. Acknowledgments

The authors would like to thank to Tim Chown, Fred Baker, Mark Townsley, Lorenzo Colitti, Ole Troan, Ray Bellis, Markus Stenberg, Wassim Haddad, Joel Halpern, Samita Chakrabarti, Michael Richardson, Anders Brandt, Erik Nordmark, Laurent Toutain, and Ralph Droms for interesting discussions in this problem space. The authors would also like to point out some past work in this space, such as those in [I-D.chelius-router-autoconf] or [I-D.dimitri-zospf].

Authors' Addresses

Jari Arkko
Ericsson
Jorvas 02420
Finland

Email: jari.arkko@piuha.net

Acee Lindem
Ericsson
Cary, NC 27519
USA

Email: acee.lindem@ericsson.com

Benjamin Paterson
Cisco Systems
Paris
France

Email: benjamin@paterson.fr

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 23, 2013

F.J. Baker
Cisco Systems
February 19, 2013

Automated prefix allocation in IS-IS
draft-baker-ipv6-isis-automatic-prefix-00

Abstract

This note describes a TLV and associated mechanisms for the allocation of /64 prefixes from a less specific prefix.

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 23, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	2
2. Theory of Operation	2
2.1. Autoconfiguration TLV Advertisement	3
2.2. Subnet prefix allocation and announcement	3
2.3. Autoconfiguration TLV Withdrawal	4
2.4. Renumbering using autoconfiguration	4
3. IPv6 Autoconfiguration TLV	5
4. IANA Considerations	5
5. Security Considerations	5
6. Privacy Considerations	6
7. Acknowledgements	6
8. Change Log	6
9. References	6
9.1. Normative References	6
9.2. Informative References	6
Author's Address	7

1. Introduction

This note recommends an approach to the automated allocation of /64 prefixes within a network. This not something that will be done in a heavily-managed network, but may be appropriate in networks with light management, such as residential and SOHO networks.

1.1. Requirements Language

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

2. Theory of Operation

The premise is that some party allocates a prefix to a network, such as a PA /48 or /56. The obvious way is using DHCP-PD [RFC3633], although that is not actually required.

IS-IS [ISO.10589.1992] represents those destinations as a type-length-value field that identifies an address. For CLNS, it was designed to the ISO NSAP; by various extensions, it also handles IPv4 and IPv6 prefixes and their counterparts for other protocols. In this model, we add a TLV to advertise the delegated prefix, with the expectation that routers in the network (including pseudo-nodes) will allocate more specific prefixes from that prefix.

In short, some specified system in the network, potentially a configuration management system or the CPE router facing an upstream network, is configured with an autoconfiguration prefix, and manages the use of that prefix in the network.

2.1. Autoconfiguration TLV Advertisement

Upon recognizing that it has been configured with a prefix and that the network management policy is for autoconfiguration, the system in question advertises the autoconfiguration prefix described in Section 3 within the intended area or network.

2.2. Subnet prefix allocation and announcement

Each router advertising a Reachability TLV [RFC5308], including a pseudonode on a LAN, when it receives the Autoconfiguration TLV Advertisement, calculates and announces a more specific prefix from the advertised autoconfiguration prefix in a Reachability TLV. This prefix is chosen at random, but may not collide with any prefix currently advertised within the network and therefore in the LSP database.

There are obvious caveats here: if the autoconfiguration prefix is too long and as a result there are more LANs than there are prefixes to allocate to them, the procedure breaks down badly, and if there are just exactly enough, it may take time to converge. Hence, from an operational perspective, the autoconfiguration prefix should have enough /64 more specific prefixes, and from an implementation perspective, the randomization function must be sufficiently robust, that independent choices are unlikely to collide.

In the event of collision, which is likely to happen from time to time, it is up to the nodes advertising the prefix in question to detect and resolve the situation. Upon receiving an LSP containing its "own" prefix advertised by another router, each router waits CollisionDetect (10) seconds, to give the network ample opportunity to detect the issue. It then waits an additional random interval between zero and CollisionDetect seconds, to randomize the recovery process and maximize the chance that replacement prefixes do not collide. It then allocates a new prefix following the procedure described in this section, and re-announces its LSP, removing (and therefore withdrawing) the offending reachability TLV, and instead announcing the new one.

Subsequent procedures, such as the advertisement of Router Advertisement using the allocated prefix or DHCPv6 allocation of addresses, may start CollisionDetect seconds after the LSP has been announced if no collision has been detected. At this point, routers MAY store their /64s in non-volatile storage.

2.3. Autoconfiguration TLV Withdrawal

When the prefix advertised in the Autoconfiguration TLV expires or is withdrawn, the Autoconfiguration TLV is withdrawn from the network. Upon detection of the withdrawal, each router in the network MUST withdraw any addresses or prefixes dependent on it. If those prefixes are stored in non-volatile storage, they MUST also be removed.

2.4. Renumbering using autoconfiguration

[RFC4192] describes the process of renumbering in some detail. The discussion here is somewhat simplistic; refer to that for a more detailed discussion.

In short, "renumbering" a network is a special case of "numbering" a network. If there is one prefix in use in a network and it is withdrawn, the network will experience an outage. Hence, it is generally advisable to ensure that there are at least two prefixes in use in a network when one of them is removed. This might be accomplished by simply using multiple prefixes in the network; it might also be accomplished by deploying a second autoconfiguration prefix minutes or hours before the "old" one is removed. During that time, DNS and DHCP databases need to be updated as described in [RFC4192] to reflect the new prefix.

If an outage is acceptable, it is also possible to renumber using the same prefix. For this, the administration withdraws the prefix as described in Section 2.3 and waits until the process is complete. There are two obvious ways to determine completion:

- o Wait long enough that it is highly unlikely to have not completed, which might be the number of routers in the network diameter times the LSP update retransmission interval, or
- o Wait until the managing router's LSP database contains no Reachability TLVs that depend on the prefix.

At this point, any systems that are only using that prefix are now unreachable using global addressing.

At this point, the managing system may re-advertise the prefix as described in Section 2.1, and the routers in the network will re-allocate prefixes as described in Section 2.3.

3. IPv6 Autoconfiguration TLV

The structure of the Autoconfiguration TLV is as follows:

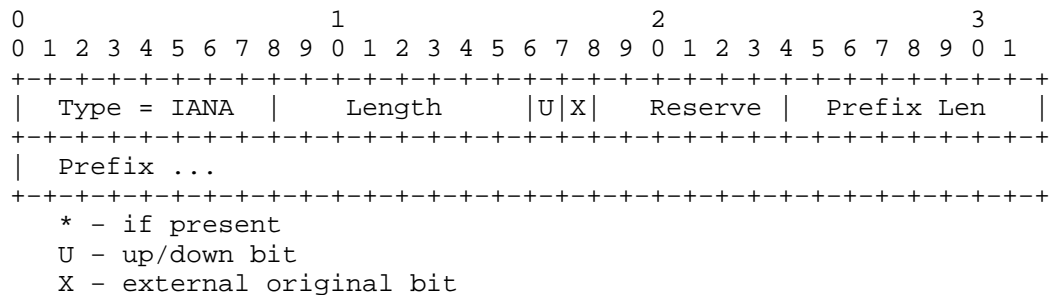


Figure 1: Autoconfiguration TLV

As is described in [RFC5305]: "The up/down bit SHALL be set to 0 when a prefix is first injected into IS-IS. If a prefix is advertised from a higher level to a lower level (e.g. level 2 to level 1), the bit SHALL be set to 1, indicating that the prefix has traveled down the hierarchy. Prefixes that have the up/down bit set to 1 may only be advertised down the hierarchy, i.e., to lower levels".

If the prefix was distributed into IS-IS from another routing protocol, the external bit SHALL be set to 1. This information is useful when distributing prefixes from IS-IS to other protocols.

The prefix is "packed" in the data structure. That is, only the required number of octets of prefix are present. This number can be computed from the prefix length octet as follows:

$$\text{prefix octets} = \text{integer of } ((\text{prefix length} + 7) / 8)$$

4. IANA Considerations

This section will request an identifying value for the TLV defined. This is deferred to the -01 version of the draft.

5. Security Considerations

To be considered.

6. Privacy Considerations

To be considered.

7. Acknowledgements

8. Change Log

Initial Version: February 2013

9. References

9.1. Normative References

- [ISO.10589.1992]
International Organization for Standardization,
"Intermediate system to intermediate system intra-domain-
routing routine information exchange protocol for use in
conjunction with the protocol for providing the
connectionless-mode Network Service (ISO 8473)", ISO
Standard 10589, 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic
Engineering", RFC 5305, October 2008.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October
2008.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF
for IPv6", RFC 5340, July 2008.

9.2. Informative References

- [I-D.baker-fun-routing-class]
Baker, F., "Routing a Traffic Class", draft-baker-fun-
routing-class-00 (work in progress), July 2011.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September
1981.
- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and
dual environments", RFC 1195, December 1990.

- [RFC1247] Moy, J., "OSPF Version 2", RFC 1247, July 1991.
- [RFC1349] Almquist, P., "Type of Service in the Internet Protocol Suite", RFC 1349, July 1992.
- [RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2597] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, September 2005.

Author's Address

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

Email: fred@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2013

F.J. Baker
Cisco Systems
February 17, 2013

Using IS-IS with Role-Based Access Control
draft-baker-ipv6-isis-dst-flowlabel-routing-00

Abstract

This note describes the changes necessary for IS-IS to route classes of IPv6 traffic that are defined by an IPv6 Flow Label and a destination prefix. This implies not routing "to a destination", but "traffic matching a classification tuple". The obvious application is data center inter-tenant routing using a form of role-based access control. If the sender doesn't know the value to insert in the flow label (the receiver's tenant ID), he in effect has no route to that destination.

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 21, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Theory of Routing	3
2.1. Dealing with ambiguity	3
3. Extensions necessary for IS-IS/IPv6	4
3.1. On Flow labels and Security	4
3.2. Flow Label sub-TLV	5
4. IANA Considerations	5
5. Security Considerations	5
6. Privacy Considerations	5
7. Acknowledgements	5
8. Change Log	5
9. References	5
9.1. Normative References	5
9.2. Informative References	6
Appendix A. Use case: Data Center Role-based Access Control . .	6
Appendix B. FIB Design	6
B.1. Staged Lookup	7
B.2. PATRICIA	7
B.2.1. Virtual Bit String	7
B.2.2. Tree Construction	8
B.2.3. Tree Lookup	8
Author's Address	9

1. Introduction

This specification builds on the extensible TLV defined in [RFC5308]. It adds to the existing Reachability TLV the (obviously optional) sub-TLV for an IPv6 Flow Label, to define routes defined by a destination prefix plus a flow label. [RFC5308] also provides an "address TLV", which enables a router to identify the prefixes in use on its interfaces. The Address TLV is not extensible; it does not permit sub-TLVs. Hence, classes of traffic defined by the destination address plus a flow label MUST be advertised using the Reachability TLV.

Advertised IS-IS TLVs that specify only a destination prefix may be understood as identifying a destination prefix used with "any" flow label, which is a very useful class of traffic to compactly represent.

1.1. Requirements Language

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

2. Theory of Routing

Both IS-IS and OSPF perform their calculations by building a lattice of routers and routes from the router performing the calculation to each router, and then use those routes to get to destinations that those routes advertise connectivity to. Following the SPF algorithm, calculation starts by selecting a starting point (typically the router doing the calculation), and successively adding {link, router} pairs until one has calculated a route to every router in the network. As each router is added, including the original router, destinations that it is directly connected to are turned into routes in the route table: "to get to 2001:db8::/32, route traffic to {interface, list of next hop routers}". For immediate neighbors to the originating router, of course, there is no next hop router; traffic is handled locally.

IS-IS [ISO.10589.1992] represents those destinations as a type-length-value field that identifies an address. For CLNS, it was designed for the ISO NSAP; by various extensions, it also handles IPv4 and IPv6 prefixes and their counterparts for other protocols. Adding a new class of traffic to route is as simple as adding a new tuple type and the supporting method routines for that class of traffic.

2.1. Dealing with ambiguity

In any routing protocol, there is the possibility of ambiguity. An area border router might, for example, summarize the routes to other areas into a small set of relatively short prefixes, which have more specific routes within the area. Traditionally, we have dealt with that using a "longest match first" rule. If the same datagram matches more than one destination prefix advertised within the area, we follow the route to the longest matching prefix.

When routing a class of traffic, we follow an analogous "most specific match" rule; we follow the route for the most specific matching tuple. In cases of simple overlap, such as routing to 2001:db8::/32 or 2001:db8:1::/48, that is exactly analogous; we choose one of the two routes.

It is possible, however, to construct an ambiguous case in which neither class subsumes the other. For example, presume that

- o A is a prefix,
- o B is a more-specific prefix within A, and
- o C is a specific flow label value

The two classes "routes to A using flow label C" and "routes to B using any flow label" are ambiguous: a datagram to B using the flow label C matches both classes, and it is not clear in the data plane what decision to make. Solving this requires the addition of a third route in the FIB corresponding to the class for routes to B using flow label C, which is more-specific than either of the first two, and can be given routing guidance based on metrics or other policy in the usual way.

3. Extensions necessary for IS-IS/IPv6

Section 2 of [RFC5308] defines the "IPv6 Reachability TLV", and carries in it destination prefix advertisements. It has the capability of extension, using sub-TLVs. The extension needed is to add a sub-TLV for each additional item in the tuple. We interpret the lack of a given sub-TLV as "any"; by definition, S=0 implies any source address, any DSCP, and any flow label. If S=1, there will be one or more additional sub-TLVs following the sub-TLV format specified there.

3.1. On Flow labels and Security

According to section 6 of [RFC2460], a Flow Label is a 20 bit number which

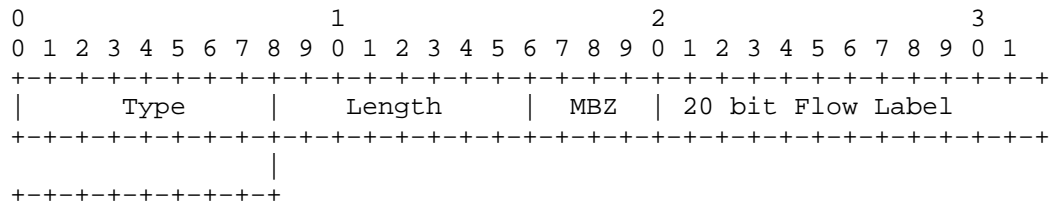
"may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers".

The possible use case mentioned in an appendix is egress routing. Other RFCs suggest other possible use cases.

In this model, the flow label is used to prove that the datagram's sender has specific knowledge of its intended receiver. No proof is requested; this is left for higher layer exchanges such as IPsec or TLS. However, if the information is distributed privately, such as through DHCP/DHCPv6, the network can presume that a system that marks traffic with the right flow label has a good chance of being authorized to communicate with its peer.

The key consideration, in this context, is that the flow label is a 20 bit number. As such, an advertised route requiring a given flow label value is calling for an exact match of all 20 bits of the label value.

3.2. Flow Label sub-TLV



Flow Label Sub-TLV

Flow Label Type: assigned by IANA

TLV Length: Length of the sub-TLV in octets

Flow Label: 20 bits of Flow Label value

MBZ: unused, MUST be zero when generated, ignored on receipt.

4. IANA Considerations

This section will request an identifying value for the TLV defined. This is deferred to the -01 version of the draft.

5. Security Considerations

To be considered.

6. Privacy Considerations

To be considered.

7. Acknowledgements

8. Change Log

Initial Version: February 2013

9. References

9.1. Normative References

[ISO.10589.1992]

International Organization for Standardization,
"Intermediate system to intermediate system intra-domain-
routing routine information exchange protocol for use in
conjunction with the protocol for providing the
connectionless-mode Network Service (ISO 8473)", ISO
Standard 10589, 1992.

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

[RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version
6 (IPv6) Specification", RFC 2460, December 1998.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October
2008.

9.2. Informative References

[PATRICIA]

Morrison, D.R., "Practical Algorithm to Retrieve
Information Coded in Alphanumeric", Journal of the ACM
15(4) pp514-534, October 1968.

Appendix A. Use case: Data Center Role-based Access Control

Consider a data center in which IPv6 is deployed throughout using internet routing technologies instead of tunnels, and the Flow Label is used to identify tenants, as discussed in Section 3.1. Hosts are required, by configuration if necessary, to know their own tenant number and the numbers of any tenants they are authorized to communicate with. When they originate a datagram, they send it to their peer's destination address and label it with their peer's tenant id. They, or their router on their behalf, advertise their own addresses as traffic classes

{destination prefix, Tenant Flow Label }

The net effect is that traffic is routed among tenants that are authorized to communicate, but not among tenants that are not authorized to communicate - there is no route. This is done without tunnels, access lists, or other data plane overhead; the overhead is in the control plane, equipping authorized parties to communicate.

Appendix B. FIB Design

While the design of the Forwarding Information Base is not a matter for standardization, as it only has to work correctly, not

interoperate with something else, the design of a FIB for this type of lookup may differ from approaches used in destination routing. We describe two possible approaches from the perspective of a proof of concept. These are a staged lookup and a single FIB.

B.1. Staged Lookup

A FIB can be designed as a staged lookup. Given that it is unlikely that any given destination would support very many tenants, a simple list or small hash may be sufficient; one looks up the destination, and having found it, validates the flow label used. In such a design, it is necessary to have the option of "any" flow label in addition to the set of specified flow labels, as it is legal and correct to advertise routes that do not have flow labels.

B.2. PATRICIA

One approach is a [PATRICIA] Tree. This is a relative of a Trie, but unlike a Trie, need not use every bit in classification, and does not need the bits used to be contiguous. It depends on treating the bit string as a set of slices of some size, potentially of different sizes. Slice width is an implementation detail; since the algorithm is most easily described using a slice of a single bit, that will be presumed in this description.

B.2.1. Virtual Bit String

It is quite possible to view the fields in a datagram header incorporated into the classification tuple as a virtual bit string such as is shown in Figure 1. This bit string has various regions within it. Some vary and are therefore useful in a radix tree lookup. Some may be essentially constant - all global IPv6 addresses at this writing are within 2000::/3, for example, so while it must be tested to assure a match, incorporating it into the radix tree may not be very helpful in classification. Others are ignored; if the destination is a remote /64, we really don't care what the EID is. In addition, due to variation in prefix length and other details, the widths of those fields vary among themselves. The algorithm the FIB implements, therefore, must efficiently deal with the fact of a discontinuous lookup key.

```

+-----+-----+-----+-----+-----+-----+
|Destination Prefix|Source Prefix|DSCP|Flow Label|
+-----+-----+-----+-----+-----+-----+
Common|Varying|Ignored|Common|Varying|Ignored|Varying or ignored

```

Figure 1: Treating a traffic class as a virtual bit string

B.2.2. Tree Construction

The tree is constructed by recursive slice-wise decomposition. At each stage, the input is a set of classes to be classified. At each stage, the result is the addition of a lookup node in the tree that identifies the location of its slice in the virtual bit string (which might be a bit number), the width of the slice to be inspected, and an enumerated set of results. Each result is a similar set of classes, and is analyzed in a similar manner.

The analysis is performed by enumerating which bits that have not already been considered are best suited to classification. For a slice of N bits, one wants to select a slide that most evenly divides the set of classes into 2^N subsets. If one or more bits in the slice is ignored in some of the classes, those classes must be included in every subset, as the actual classification of them will depend on other bits.

```
Input: {2001:db8::/32, ::/0, *, *}
      {2001:db8:1::/48, ::/0, AF41, *}
      {2001:db8:1::/48, ::/0, AF42, *}
      {2001:db8:1::/48, ::/0, AF43, *}
```

Common parts: Destination prefix 2001:dba, source prefix, and label
 Varying parts: DSCP and the third set of sixteen bits in the destination prefix

One possible decomposition:

(1) slice = DSCP

enumerated cases:

- (a) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF41, *} }
- (b) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF42, *} }
- (c) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF43, *} }

(2) slice = third sixteen bit field in destination

This divides each enumerated case into those containing 0001 and "everything else", which would imply 2001:db8::/32

(1) DSCP

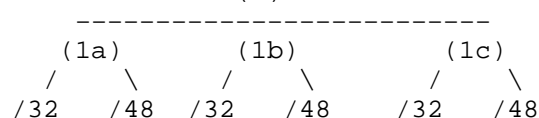


Figure 2: Example PATRICIA Tree

B.2.3. Tree Lookup

To look something up in a PATRICIA Tree, one starts at the root of the tree and performs the indicated comparisons recursively walking down the tree until one reaches a terminal node. When the enumerated subset is empty or contains only a single class, classification

stops. Either classification has failed (there was no matching class, or one has presumably found the indicated class. At that point, every bit in the virtual bit string must be compared to the classifier; classification is accepted on a perfect match.

In the example in Figure 2, if a packet {2001:db8:1:2:3:4:5:6, 2001:db8:2:3:4:5:6:7, AF41, 0} arrives, we start at the root. Since it is an AF41 packet, we deduce that case (1a) applies, and since the destination has 0001 in the third sixteen bit field of the destination address, we are comparing to {2001:db8:1::/48, ::/0, AF41, *}. Since the destination address is within 2001:db8:1::/48, classification as that succeeds.

Author's Address

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

Email: fred@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2013

F.J. Baker
Cisco Systems
February 17, 2013

IPv6 Source/Destination Routing using IS-IS
draft-baker-ipv6-isis-dst-src-routing-00

Abstract

This note describes the changes necessary for IS-IS to route classes of IPv6 traffic that are defined by a source prefix and a destination prefix. This implies not routing "to a destination", but "traffic matching a classification tuple". The obvious application is egress routing - routing traffic using a given prefix to an upstream network that will not drop traffic using that prefix using BCP 38 filters.

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 21, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Theory of Routing	3
2.1. Dealing with ambiguity	3
3. Extensions necessary for IS-IS/IPv6	4
3.1. Source Prefix sub-TLV	4
4. IANA Considerations	5
5. Security Considerations	5
6. Privacy Considerations	5
7. Acknowledgements	5
8. Change Log	5
9. References	5
9.1. Normative References	5
9.2. Informative References	5
Appendix A. Use case: Egress Routing	6
Appendix B. FIB Design	7
B.1. Linux Source-Address Forwarding	7
B.1.1. One FIB per source prefix	7
B.1.2. One FIB per source prefix plus a general FIB	8
B.2. PATRICIA	9
B.2.1. Virtual Bit String	9
B.2.2. Tree Construction	9
B.2.3. Tree Lookup	10
Author's Address	10

1. Introduction

This specification builds on the extensible TLV defined in [RFC5308]. It adds to the existing Reachability TLV the (obviously optional) sub-TLV for an IPv6 Source Prefix, to define routes defined by a source and a destination prefix. Note that `::/0`, "any IPv6 Address", is a prefix, so this may be used for default routes as well as more specific routes.

IS-IS TLVs that specify only a destination prefix remain legal. They may be understood as identifying a route to a destination prefix from

"any" source, which is a very useful class of traffic to compactly represent.

1.1. Requirements Language

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

2. Theory of Routing

Both IS-IS and OSPF perform their calculations by building a lattice of routers and routes from the router performing the calculation to each router, and then use those routes to get to destinations that those routes advertise connectivity to. Following the SPF algorithm, calculation starts by selecting a starting point (typically the router doing the calculation), and successively adding {link, router} pairs until one has calculated a route to every router in the network. As each router is added, including the original router, destinations that it is directly connected to are turned into routes in the route table: "to get to 2001:db8::/32, route traffic to {interface, list of next hop routers}". For immediate neighbors to the originating router, of course, there is no next hop router; traffic is handled locally.

IS-IS [ISO.10589.1992] represents those destinations as a type-length-value field that identifies an address. For CLNS, it was designed to the ISO NSAP; by various extensions, it also handles IPv4 and IPv6 prefixes and their counterparts for other protocols. Adding a new class of traffic to route is as simple as adding a new tuple type and the supporting method routines for that class of traffic.

2.1. Dealing with ambiguity

In any routing protocol, there is the possibility of ambiguity. An area border router might, for example, summarize the routes to other areas into a small set of relatively short prefixes, which have more specific routes within the area. Traditionally, we have dealt with that using a "longest match first" rule. If the same datagram matches more than one destination prefix advertised within the area, we follow the route to the longest matching prefix.

When routing a class of traffic, we follow an analogous "most specific match" rule; we follow the route for the most specific matching tuple. In cases of simple overlap, such as routing to 2001:db8::/32 or 2001:db8:1::/48, that is exactly analogous; we choose one of the two routes.

4. IANA Considerations

This section will request an identifying value for the TLV defined. This is deferred to the -01 version of the draft.

5. Security Considerations

To be considered.

6. Privacy Considerations

To be considered.

7. Acknowledgements

8. Change Log

Initial Version: February 2013

9. References

9.1. Normative References

- [ISO.10589.1992]
International Organization for Standardization,
"Intermediate system to intermediate system intra-domain-
routing routine information exchange protocol for use in
conjunction with the protocol for providing the
connectionless-mode Network Service (ISO 8473)", ISO
Standard 10589, 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October
2008.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF
for IPv6", RFC 5340, July 2008.

9.2. Informative References

- [I-D.baker-fun-routing-class]
Baker, F., "Routing a Traffic Class", draft-baker-fun-
routing-class-00 (work in progress), July 2011.

[PATRICIA]

Morrison, D.R., "Practical Algorithm to Retrieve Information Coded in Alphanumeric", Journal of the ACM 15(4) pp514-534, October 1968.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[RFC1247] Moy, J., "OSPF Version 2", RFC 1247, July 1991.

[RFC1349] Almquist, P., "Type of Service in the Internet Protocol Suite", RFC 1349, July 1992.

[RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2597] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

Appendix A. Use case: Egress Routing

Using this technology for egress routing is straightforward. Presume a multihomed edge (residential or enterprise) network with multiple egress points to the various ISPs. These ISPs allocate PA prefixes to the network. Due to BCP 38 [RFC2827], the network must presume that its upstream ISPs will filter out any traffic presented to them that does not use their PA prefix.

Within the network, presume that a /64 prefix from each of those PA prefixes is allocated on each LAN, and that hosts generate and use multiple addresses on each interface.

Within the network, we permit any host to communicate with any other. Hence, routing advertisements within the network use traditional destination routing, which is understood to be advertising the traffic class

{destination, ::/0}.

From the egresses, the firewall or its neighboring router injects a default route for traffic "from" its PA prefix:

{::/0, PA prefix}.

Routing is calculated as normal, with the exception that traffic following a default route will select that route based on the source address. Traffic will never be lost to BCP 38 filters, because by definition the only traffic sent to the ISP is using the PA prefix assigned by the ISP. In addition, while hosts can use spoofed addresses outside of their PA prefixes to attack each other, they cannot send traffic using spoofed addresses to their upstream networks; such traffic has no route.

Appendix B. FIB Design

While the design of the Forwarding Information Base is not a matter for standardization, as it only has to work correctly, not interoperate with something else, the design of a FIB for this type of lookup may differ from approaches used in destination routing. We describe one possible approach that is known to work, from the perspective of a proof of concept.

B.1. Linux Source-Address Forwarding

The University of Waikato has added to the Linux Advanced Routing & Traffic Control facility the ability to maintain multiple FIBs, one for each of a set of prefixes. Implementing source/destination routing using this mechanism is not difficult.

The router must know what source prefixes might be used in its domain. This may be by configuration or, at least in concept, learned from the routing protocols themselves. In whichever way that is done, one can imagine two fundamental FIB structures to serve N source prefixes; N FIBs, one per prefix, or N+1 FIBs, one per prefix plus one for destinations for which the source prefix is unspecified.

B.1.1. One FIB per source prefix

In an implementation with one FIB per source prefix, the routing algorithm has two possibilities.

- o If it calculates a route to a prefix (such as a default route) associated with a given source prefix, it stores the route in the FIB for the relevant source prefix.
- o If it calculates a route for which the source prefix is unspecified, it stores that route in all N FIBs.

When forwarding a datagram, the IP forwarder looks at the source address of the datagram to determine which FIB it should use. If it

is from an address for which there is no FIB, the forwarder discards the datagram as containing a forged source address. If it is from an address within one of the relevant prefixes, it looks up the destination in the indicated FIB and forwards it in the usual way.

The argument for this approach is simplicity: there is one place to look in making a forwarding decision for any given datagram. The argument against it is memory space; it is likely that the FIBs will be similar, but every destination route not associated with a source prefix is duplicated in each FIB. In addition, since it automatically removes traffic whose source address is not among the configured list, it limits the possibility of user software using improper addresses.

B.1.1.2. One FIB per source prefix plus a general FIB

In an implementation with N+1 FIBs, the algorithm is slightly more complex.

- o If it calculates a route to a prefix (such as a default route) associated with a given source prefix, it stores the route in the FIB for the relevant source prefix.
- o If it calculates a route for which the source prefix is unspecified, it stores that route in the FIB that is not associated with a source prefix.

When forwarding a datagram, the IP forwarder looks at the source address of the datagram to determine which FIB it should use. If it is from one of the configured prefixes, it looks the destination up in the indicated FIB. In any event it also looks the destination up in the "unspecified source address" FIB. If the destination is found in only one of the two, the indicated route is followed. If the destination is found in both, the more specific route is followed.

The argument for this approach is memory space; if a large percentage of routes are only in the general FIB, such as when egress routing is used for the default route and all other routes are internal, the other FIBs are likely to be very small - perhaps only a single default route. The argument against this approach is complexity: most lookups if not all will be done in a prefix-specific FIB and in the general FIB.

B.2. PATRICIA

One approach is a [PATRICIA] Tree. This is a relative of a Trie, but unlike a Trie, need not use every bit in classification, and does not need the bits used to be contiguous. It depends on treating the bit string as a set of slices of some size, potentially of different sizes. Slice width is an implementation detail; since the algorithm is most easily described using a slice of a single bit, that will be presumed in this description.

B.2.1. Virtual Bit String

It is quite possible to view the fields in a datagram header incorporated into the classification tuple as a virtual bit string such as is shown in Figure 1. This bit string has various regions within it. Some vary and are therefore useful in a radix tree lookup. Some may be essentially constant - all global IPv6 addresses at this writing are within 2000::/3, for example, so while it must be tested to assure a match, incorporating it into the radix tree may not be very helpful in classification. Others are ignored; if the destination is a remote /64, we really don't care what the EID is. In addition, due to variation in prefix length and other details, the widths of those fields vary among themselves. The algorithm the FIB implements, therefore, must efficiently deal with the fact of a discontinuous lookup key.

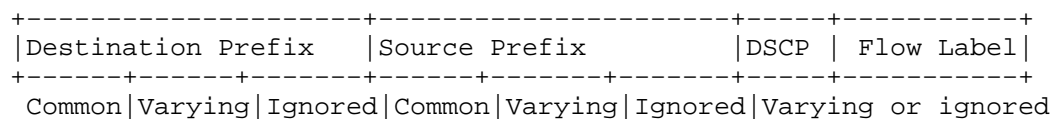


Figure 1: Treating a traffic class as a virtual bit string

B.2.2. Tree Construction

The tree is constructed by recursive slice-wise decomposition. At each stage, the input is a set of classes to be classified. At each stage, the result is the addition of a lookup node in the tree that identifies the location of its slice in the virtual bit string (which might be a bit number), the width of the slice to be inspected, and an enumerated set of results. Each result is a similar set of classes, and is analyzed in a similar manner.

The analysis is performed by enumerating which bits that have not already been considered are best suited to classification. For a slice of N bits, one wants to select a slide that most evenly divides the set of classes into 2^N subsets. If one or more bits in the slice is ignored in some of the classes, those classes must be

included in every subset, as the actual classification of them will depend on other bits.

```
Input: {2001:db8::/32, ::/0, *, *}
       {2001:db8:1::/48, ::/0, AF41, *}
       {2001:db8:1::/48, ::/0, AF42, *}
       {2001:db8:1::/48, ::/0, AF43, *}
```

Common parts: Destination prefix 2001:dba, source prefix, and label

Varying parts: DSCP and the third set of sixteen bits in the destination prefix

One possible decomposition:

(1) slice = DSCP

enumerated cases:

- (a) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF41, *} }
- (b) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF42, *} }
- (c) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF43, *} }

(2) slice = third sixteen bit field in destination

This divides each enumerated case into those containing 0001 and "everything else", which would imply 2001:db8::/32

(1) DSCP

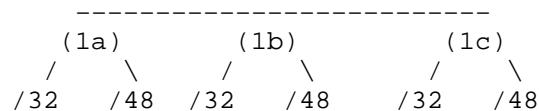


Figure 2: Example PATRICIA Tree

B.2.3. Tree Lookup

To look something up in a PATRICIA Tree, one starts at the root of the tree and performs the indicated comparisons recursively walking down the tree until one reaches a terminal node. When the enumerated subset is empty or contains only a single class, classification stops. Either classification has failed (there was no matching class, or one has presumably found the indicated class. At that point, every bit in the virtual bit string must be compared to the classifier; classification is accepted on a perfect match.

In the example in Figure 2, if a packet {2001:db8:1:2:3:4:5:6, 2001:db8:2:3:4:5:6:7, AF41, 0} arrives, we start at the root. Since it is an AF41 packet, we deduce that case (1a) applies, and since the destination has 0001 in the third sixteen bit field of the destination address, we are comparing to {2001:db8:1::/48, ::/0, AF41, *}. Since the destination address is within 2001:db8:1::/48, classification as that succeeds.

Author's Address

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

Email: fred@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2013

F.J. Baker
Cisco Systems
February 17, 2013

Using OSPFv3 with Role-Based Access Control
draft-baker-ipv6-ospf-dst-flowlabel-routing-00

Abstract

This note describes the changes necessary for OSPFv3 to route classes of IPv6 traffic that are defined by an IPv6 Flow Label and a destination prefix. This implies not routing "to a destination", but "traffic matching a classification tuple". The obvious application is data center inter-tenant routing using a form of role-based access control. If the sender doesn't know the value to insert in the flow label (the receiver's tenant ID), he in effect has no route to that destination.

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 21, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	2
2. Theory of Routing	3
2.1. Dealing with ambiguity	3
3. Extensions necessary for OSPFv3	4
3.1. On Flow Labels and security	4
3.2. Flow Label TLV	5
4. IANA Considerations	5
5. Security Considerations	5
6. Privacy Considerations	5
7. Acknowledgements	5
8. Change Log	5
9. References	5
9.1. Normative References	5
9.2. Informative References	6
Appendix A. Use case: Data Center Role-based Access Control . .	6
Appendix B. FIB Design	6
B.1. Staged Lookup	7
B.2. PATRICIA	7
B.2.1. Virtual Bit String	7
B.2.2. Tree Construction	8
B.2.3. Tree Lookup	8
Author's Address	9

1. Introduction

This specification builds on the extensible LSAs defined in [I-D.baker-ipv6-ospf-extensible.txt]. It adds the option for an IPv6 Flow Label, to define routes defined by a destination prefix plus a flow label.

1.1. Requirements Language

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

2. Theory of Routing

Both IS-IS and OSPF perform their calculations by building a lattice of routers and routes from the router performing the calculation to each router, and then use those routes to get to destinations that those routes advertise connectivity to. Following the SPF algorithm, calculation starts by selecting a starting point (typically the router doing the calculation), and successively adding {link, router} pairs until one has calculated a route to every router in the network. As each router is added, including the original router, destinations that it is directly connected to are turned into routes in the route table: "to get to 2001:db8::/32, route traffic to {interface, list of next hop routers}". For immediate neighbors to the originating router, of course, there is no next hop router; traffic is handled locally.

2.1. Dealing with ambiguity

In any routing protocol, there is the possibility of ambiguity. An area border router might, for example, summarize the routes to other areas into a small set of relatively short prefixes, which have more specific routes within the area. Traditionally, we have dealt with that using a "longest match first" rule. If the same datagram matches more than one destination prefix advertised within the area, we follow the route to the longest matching prefix.

When routing a class of traffic, we follow an analogous "most specific match" rule; we follow the route for the most specific matching tuple. In cases of simple overlap, such as routing to 2001:db8::/32 or 2001:db8:1::/48, that is exactly analogous; we choose one of the two routes.

It is possible, however, to construct an ambiguous case in which neither class subsumes the other. For example, presume that

- o A is a prefix,
- o B is a more-specific prefix within A,
- o C is a specific flow label value

The two classes "routes to A using flow label C" and "routes to B using any flow label" are ambiguous: a datagram to B using the flow label C matches both classes, and it is not clear in the data plane what decision to make. Solving this requires the addition of a third route in the FIB corresponding to the class for routes to B using flow label C, which is more-specific than either of the first two, and can be given routing guidance based on metrics or other policy in the usual way.

3. Extensions necessary for OSPFv3

The several extensible LSAs defined in [I-D.baker-ipv6-ospf-extensible.txt] require one additional option to accomplish source/destination routing: the flow label in use by the destination. This is defined here.

In addition, should (as one might expect is normal) destination-only intra-area-prefix, inter-area-prefix, and AS-external-prefix LSAs be encountered, we need a rule for interpretation. The rule is that they are treated exactly as the extensible version if the flow label TLV is omitted, which is to say, that any flow label value is accepted.

3.1. On Flow Labels and security

According to section 6 of [RFC2460], a Flow Label is a 20 bit number which

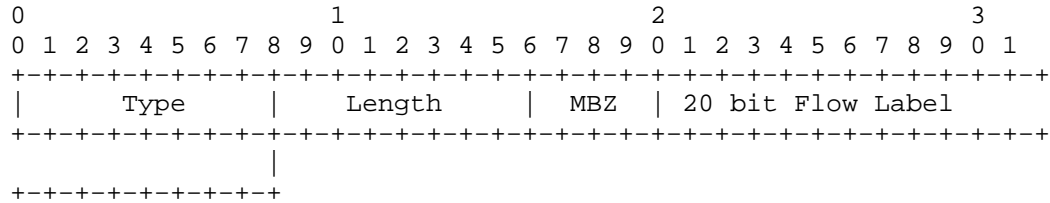
"may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers".

The possible use case mentioned in an appendix is egress routing. Other RFCs suggest other possible use cases.

In this model, the flow label is used to prove that the datagram's sender has specific knowledge of its intended receiver. No proof is requested; this is left for higher layer exchanges such as IPsec or TLS. However, if the information is distributed privately, such as through DHCP/DHCPv6, the network can presume that a system that marks traffic with the right flow label has a good chance of being authorized to communicate with its peer.

The key consideration, in this context, is that the flow label is a 20 bit number. As such, an advertised route requiring a given flow label value is calling for an exact match of all 20 bits of the label value.

3.2. Flow Label TLV



Flow Label TLV

Flow Label Type: assigned by IANA

TLV Length: Length of the TLV in octets

Flow Label: 20 bits of Flow Label value

MBZ: unused, MUST be zero when generated, ignored on receipt.

4. IANA Considerations

This section will request an identifying value for the TLV defined. This is deferred to the -01 version of the draft.

5. Security Considerations

To be considered.

6. Privacy Considerations

To be considered.

7. Acknowledgements

8. Change Log

Initial Version: February 2013

9. References

9.1. Normative References

- [I-D.baker-ipv6-ospf-extensible.txt]
Baker, F., "Extensible OSPF LSAs", February 2013.
- [ISO.10589.1992]

International Organization for Standardization,
"Intermediate system to intermediate system intra-domain-
routing routine information exchange protocol for use in
conjunction with the protocol for providing the
connectionless-mode Network Service (ISO 8473)", ISO
Standard 10589, 1992.

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

[RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version
6 (IPv6) Specification", RFC 2460, December 1998.

9.2. Informative References

[PATRICIA]

Morrison, D.R., "Practical Algorithm to Retrieve
Information Coded in Alphanumeric", Journal of the ACM
15(4) pp514-534, October 1968.

Appendix A. Use case: Data Center Role-based Access Control

Consider a data center in which IPv6 is deployed throughout using internet routing technologies instead of tunnels, and the Flow Label is used to identify tenants, as discussed in Section 3.1. Hosts are required, by configuration if necessary, to know their own tenant number and the numbers of any tenants they are authorized to communicate with. When they originate a datagram, they send it to their peer's destination address and label it with their peer's tenant id. They, or their router on their behalf, advertise their own addresses as traffic classes

{destination prefix, Tenant Flow Label }

The net effect is that traffic is routed among tenants that are authorized to communicate, but not among tenants that are not authorized to communicate - there is no route. This is done without tunnels, access lists, or other data plane overhead; the overhead is in the control plane, equipping authorized parties to communicate.

Appendix B. FIB Design

While the design of the Forwarding Information Base is not a matter for standardization, as it only has to work correctly, not interoperate with something else, the design of a FIB for this type of lookup may differ from approaches used in destination routing. We describe two possible approaches from the perspective of a proof of concept. These are a staged lookup and a single FIB.

B.1. Staged Lookup

A FIB can be designed as a staged lookup. Given that it is unlikely that any given destination would support very many tenants, a simple list or small hash may be sufficient; one looks up the destination, and having found it, validates the flow label used. In such a design, it is necessary to have the option of "any" flow label in addition to the set of specified flow labels, as it is legal and correct to advertise routes that do not have flow labels.

B.2. PATRICIA

One approach is a [PATRICIA] Tree. This is a relative of a Trie, but unlike a Trie, need not use every bit in classification, and does not need the bits used to be contiguous. It depends on treating the bit string as a set of slices of some size, potentially of different sizes. Slice width is an implementation detail; since the algorithm is most easily described using a slice of a single bit, that will be presumed in this description.

B.2.1. Virtual Bit String

It is quite possible to view the fields in a datagram header incorporated into the classification tuple as a virtual bit string such as is shown in Figure 1. This bit string has various regions within it. Some vary and are therefore useful in a radix tree lookup. Some may be essentially constant - all global IPv6 addresses at this writing are within 2000::/3, for example, so while it must be tested to assure a match, incorporating it into the radix tree may not be very helpful in classification. Others are ignored; if the destination is a remote /64, we really don't care what the EID is. In addition, due to variation in prefix length and other details, the widths of those fields vary among themselves. The algorithm the FIB implements, therefore, must efficiently deal with the fact of a discontinuous lookup key.

```

+-----+-----+-----+-----+
|Destination Prefix |Source Prefix           |DSCP | Flow Label|
+-----+-----+-----+-----+
Common|Varying|Ignored|Common|Varying|Ignored|Varying or ignored

```

Figure 1: Treating a traffic class as a virtual bit string

B.2.2. Tree Construction

The tree is constructed by recursive slice-wise decomposition. At each stage, the input is a set of classes to be classified. At each stage, the result is the addition of a lookup node in the tree that identifies the location of its slice in the virtual bit string (which might be a bit number), the width of the slice to be inspected, and an enumerated set of results. Each result is a similar set of classes, and is analyzed in a similar manner.

The analysis is performed by enumerating which bits that have not already been considered are best suited to classification. For a slice of N bits, one wants to select a slide that most evenly divides the set of classes into 2^N subsets. If one or more bits in the slice is ignored in some of the classes, those classes must be included in every subset, as the actual classification of them will depend on other bits.

```
Input: {2001:db8::/32, ::/0, *, *}
      {2001:db8:1::/48, ::/0, AF41, *}
      {2001:db8:1::/48, ::/0, AF42, *}
      {2001:db8:1::/48, ::/0, AF43, *}
```

Common parts: Destination prefix 2001:dba, source prefix, and label
 Varying parts: DSCP and the third set of sixteen bits in the destination prefix

One possible decomposition:

(1) slice = DSCP

enumerated cases:

- (a) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF41, *} }
- (b) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF42, *} }
- (c) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF43, *} }

(2) slice = third sixteen bit field in destination

This divides each enumerated case into those containing 0001 and "everything else", which would imply 2001:db8::/32

(1) DSCP

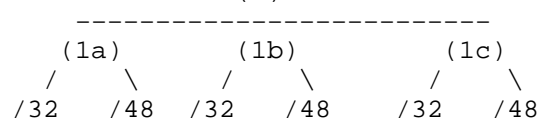


Figure 2: Example PATRICIA Tree

B.2.3. Tree Lookup

To look something up in a PATRICIA Tree, one starts at the root of the tree and performs the indicated comparisons recursively walking down the tree until one reaches a terminal node. When the enumerated subset is empty or contains only a single class, classification

stops. Either classification has failed (there was no matching class, or one has presumably found the indicated class. At that point, every bit in the virtual bit string must be compared to the classifier; classification is accepted on a perfect match.

In the example in Figure 2, if a packet {2001:db8:1:2:3:4:5:6, 2001:db8:2:3:4:5:6:7, AF41, 0} arrives, we start at the root. Since it is an AF41 packet, we deduce that case (1a) applies, and since the destination has 0001 in the third sixteen bit field of the destination address, we are comparing to {2001:db8:1::/48, ::/0, AF41, *}. Since the destination address is within 2001:db8:1::/48, classification as that succeeds.

Author's Address

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

Email: fred@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2013

F.J. Baker
Cisco Systems
February 17, 2013

IPv6 Source/Destination Routing using OSPFv3
draft-baker-ipv6-ospf-dst-src-routing-00

Abstract

This note describes the changes necessary for OSPFv3 to route classes of IPv6 traffic that are defined by a source prefix and a destination prefix. This implies not routing "to a destination", but "traffic matching a classification tuple". The obvious application is egress routing - routing traffic using a given prefix to an upstream network that will not drop traffic using that prefix using BCP 38 filters.

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 21, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	2
2. Theory of Routing	3
2.1. Dealing with ambiguity	3
3. Extensions necessary for IPv6 Source/Destination Routing in OSPFv3	4
3.1. IPv6 Source Prefix TLV	4
4. IANA Considerations	4
5. Security Considerations	5
6. Privacy Considerations	5
7. Acknowledgements	5
8. Change Log	5
9. References	5
9.1. Normative References	5
9.2. Informative References	5
Appendix A. Use case: Egress Routing	6
Appendix B. FIB Design	6
B.1. Linux Source-Address Forwarding	7
B.1.1. One FIB per source prefix	7
B.1.2. One FIB per source prefix plus a general FIB	7
B.2. PATRICIA	8
B.2.1. Virtual Bit String	8
B.2.2. Tree Construction	9
B.2.3. Tree Lookup	10
Author's Address	10

1. Introduction

This specification builds on the extensible LSAs defined in [I-D.baker-ipv6-ospf-extensible.txt]. It adds the option for an IPv6 Source Prefix, to define routes defined by a source and a destination prefix.

1.1. Requirements Language

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

2. Theory of Routing

Both IS-IS and OSPF perform their calculations by building a lattice of routers and routes from the router performing the calculation to each router, and then use those routes to get to destinations that those routes advertise connectivity to. Following the SPF algorithm, calculation starts by selecting a starting point (typically the router doing the calculation), and successively adding {link, router} pairs until one has calculated a route to every router in the network. As each router is added, including the original router, destinations that it is directly connected to are turned into routes in the route table: "to get to 2001:db8::/32, route traffic to {interface, list of next hop routers}". For immediate neighbors to the originating router, of course, there is no next hop router; traffic is handled locally.

2.1. Dealing with ambiguity

In any routing protocol, there is the possibility of ambiguity. An area border router might, for example, summarize the routes to other areas into a small set of relatively short prefixes, which have more specific routes within the area. Traditionally, we have dealt with that using a "longest match first" rule. If the same datagram matches more than one destination prefix advertised within the area, we follow the route to the longest matching prefix.

When routing a class of traffic, we follow an analogous "most specific match" rule; we follow the route for the most specific matching tuple. In cases of simple overlap, such as routing to 2001:db8::/32 or 2001:db8:1::/48, that is exactly analogous; we choose one of the two routes.

It is possible, however, to construct an ambiguous case in which neither class subsumes the other. For example, presume that

- o A is a prefix,
- o B is a more-specific prefix within A,
- o C is a different prefix, and
- o D is a more-specific prefix of C.

The two classes {A, D, *, *} and {B, C, *, *} are ambiguous: a datagram within {B, D, *, *} matches both classes, and it is not clear in the data plane what decision to make. Solving this requires the addition of a third route in the FIB corresponding to the class {B, D, *, *}, which is more-specific than either of the first two, and can be given routing guidance based on metrics or other policy in the usual way.

3. Extensions necessary for IPv6 Source/Destination Routing in OSPFv3

The several extensible LSAs defined in [I-D.baker-ipv6-ospf-extensible.txt] require one additional option to accomplish source/destination routing: the source prefix. This is defined here.

In addition, should (as one might expect is normal) destination-only intra-area-prefix, inter-area-prefix, and AS-external-prefix LSAs be encountered, we need a rule for interpretation. The rule is that they are treated exactly as the extensible version if the source prefix option is not specified or is specified to be ::/0 (any IPv6 address).

3.1. IPv6 Source Prefix TLV

The IPv6 Source Prefix TLV MAY be used with the IPv6 Destination Prefix TLV, but MUST NOT be used with the IPv4 Source Prefix TLV or the IPv4 Destination Prefix TLV.

0																1																2																3															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																						
Type																Length																Prefix Length																Prefix															

Source Prefix TLV

Source Prefix Type: assigned by IANA

TLV Length: Length of the TLV in octets

Prefix Length: Length of the prefix in bits, in the range 0..128

Prefix: (source prefix length + 7)/8 octets of prefix

4. IANA Considerations

This section will request an identifying value for the TLV defined. This is deferred to the -01 version of the draft.

5. Security Considerations

To be considered.

6. Privacy Considerations

To be considered.

7. Acknowledgements

8. Change Log

Initial Version: February 2013

9. References

9.1. Normative References

[ISO.10589.1992]

International Organization for Standardization,
"Intermediate system to intermediate system intra-domain-
routing routine information exchange protocol for use in
conjunction with the protocol for providing the
connectionless-mode Network Service (ISO 8473)", ISO
Standard 10589, 1992.

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

9.2. Informative References

[I-D.baker-ipv6-ospf-extensible.txt]

Baker, F., "Extensible OSPF LSAs", February 2013.

[PATRICIA]

Morrison, D.R., "Practical Algorithm to Retrieve
Information Coded in Alphanumeric", Journal of the ACM
15(4) pp514-534, October 1968.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering:
Defeating Denial of Service Attacks which employ IP Source
Address Spoofing", BCP 38, RFC 2827, May 2000.

Appendix A. Use case: Egress Routing

Using this technology for egress routing is straightforward. Presume a multihomed edge (residential or enterprise) network with multiple egress points to the various ISPs. These ISPs allocate PA prefixes to the network. Due to BCP 38 [RFC2827], the network must presume that its upstream ISPs will filter out any traffic presented to them that does not use their PA prefix.

Within the network, presume that a /64 prefix from each of those PA prefixes is allocated on each LAN, and that hosts generate and use multiple addresses on each interface.

Within the network, we permit any host to communicate with any other. Hence, routing advertisements within the network use traditional destination routing, which is understood to be advertising the traffic class

{destination, ::/0}.

From the egresses, the firewall or its neighboring router injects a default route for traffic "from" its PA prefix:

{::/0, PA prefix}.

Routing is calculated as normal, with the exception that traffic following a default route will select that route based on the source address. Traffic will never be lost to BCP 38 filters, because by definition the only traffic sent to the ISP is using the PA prefix assigned by the ISP. In addition, while hosts can use spoofed addresses outside of their PA prefixes to attack each other, they cannot send traffic using spoofed addresses to their upstream networks; such traffic has no route.

Appendix B. FIB Design

While the design of the Forwarding Information Base is not a matter for standardization, as it only has to work correctly, not interoperate with something else, the design of a FIB for this type of lookup may differ from approaches used in destination routing. We describe one possible approach that is known to work, from the perspective of a proof of concept.

B.1. Linux Source-Address Forwarding

The University of Waikato has added to the Linux Advanced Routing & Traffic Control facility the ability to maintain multiple FIBs, one for each of a set of prefixes. Implementing source/destination routing using this mechanism is not difficult.

The router must know what source prefixes might be used in its domain. This may be by configuration or, at least in concept, learned from the routing protocols themselves. In whichever way that is done, one can imagine two fundamental FIB structures to serve N source prefixes; N FIBs, one per prefix, or N+1 FIBs, one per prefix plus one for destinations for which the source prefix is unspecified.

B.1.1. One FIB per source prefix

In an implementation with one FIB per source prefix, the routing algorithm has two possibilities.

- o If it calculates a route to a prefix (such as a default route) associated with a given source prefix, it stores the route in the FIB for the relevant source prefix.
- o If it calculates a route for which the source prefix is unspecified, it stores that route in all N FIBs.

When forwarding a datagram, the IP forwarder looks at the source address of the datagram to determine which FIB it should use. If it is from an address for which there is no FIB, the forwarder discards the datagram as containing a forged source address. If it is from an address within one of the relevant prefixes, it looks up the destination in the indicated FIB and forwards it in the usual way.

The argument for this approach is simplicity: there is one place to look in making a forwarding decision for any given datagram. The argument against it is memory space; it is likely that the FIBs will be similar, but every destination route not associated with a source prefix is duplicated in each FIB. In addition, since it automatically removes traffic whose source address is not among the configured list, it limits the possibility of user software using improper addresses.

B.1.2. One FIB per source prefix plus a general FIB

In an implementation with N+1 FIBs, the algorithm is slightly more complex.

- o If it calculates a route to a prefix (such as a default route) associated with a given source prefix, it stores the route in the FIB for the relevant source prefix.
- o If it calculates a route for which the source prefix is unspecified, it stores that route in the FIB that is not associated with a source prefix.

When forwarding a datagram, the IP forwarder looks at the source address of the datagram to determine which FIB it should use. If it is from one of the configured prefixes, it looks the destination up in the indicated FIB. In any event it also looks the destination up in the "unspecified source address" FIB. If the destination is found in only one of the two, the indicated route is followed. If the destination is found in both, the more specific route is followed.

The argument for this approach is memory space; if a large percentage of routes are only in the general FIB, such as when egress routing is used for the default route and all other routes are internal, the other FIBs are likely to be very small - perhaps only a single default route. The argument against this approach is complexity: most lookups if not all will be done in a prefix-specific FIB and in the general FIB.

B.2. PATRICIA

One approach is a [PATRICIA] Tree. This is a relative of a Trie, but unlike a Trie, need not use every bit in classification, and does not need the bits used to be contiguous. It depends on treating the bit string as a set of slices of some size, potentially of different sizes. Slice width is an implementation detail; since the algorithm is most easily described using a slice of a single bit, that will be presumed in this description.

B.2.1. Virtual Bit String

It is quite possible to view the fields in a datagram header incorporated into the classification tuple as a virtual bit string such as is shown in Figure 1. This bit string has various regions within it. Some vary and are therefore useful in a radix tree lookup. Some may be essentially constant - all global IPv6 addresses at this writing are within 2000::/3, for example, so while it must be tested to assure a match, incorporating it into the radix tree may not be very helpful in classification. Others are ignored; if the destination is a remote /64, we really don't care what the EID is. In addition, due to variation in prefix length and other details, the widths of those fields vary among themselves. The algorithm the FIB implements, therefore, must efficiently deal with the fact of a discontinuous lookup key.

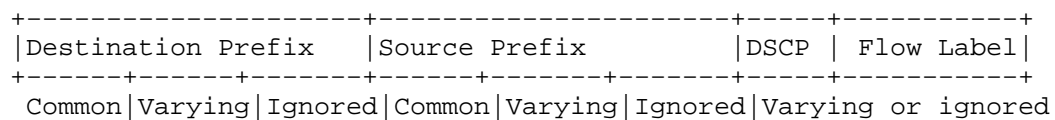


Figure 1: Treating a traffic class as a virtual bit string

B.2.2. Tree Construction

The tree is constructed by recursive slice-wise decomposition. At each stage, the input is a set of classes to be classified. At each stage, the result is the addition of a lookup node in the tree that identifies the location of its slice in the virtual bit string (which might be a bit number), the width of the slice to be inspected, and an enumerated set of results. Each result is a similar set of classes, and is analyzed in a similar manner.

The analysis is performed by enumerating which bits that have not already been considered are best suited to classification. For a slice of N bits, one wants to select a slide that most evenly divides the set of classes into 2^N subsets. If one or more bits in the slice is ignored in some of the classes, those classes must be included in every subset, as the actual classification of them will depend on other bits.

```

Input:{2001:db8::/32, ::/0, *, *}
      {2001:db8:1::/48, ::/0, AF41, *}
      {2001:db8:1::/48, ::/0, AF42, *}
      {2001:db8:1::/48, ::/0, AF43, *}
Common parts: Destination prefix 2001:dba, source prefix, and label
Varying parts: DSCP and the third set of sixteen bits in the
                destination prefix
One possible decomposition:
(1) slice = DSCP

```

enumerated cases:

- (a) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF41, *} }
- (b) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF42, *} }
- (c) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF43, *} }
- (2) slice = third sixteen bit field in destination

This divides each enumerated case into those containing 0001 and "everything else", which would imply 2001:db8::/32

(1) DSCP

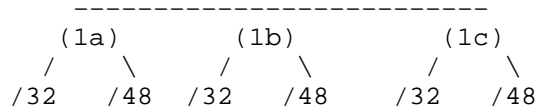


Figure 2: Example PATRICIA Tree

B.2.3. Tree Lookup

To look something up in a PATRICIA Tree, one starts at the root of the tree and performs the indicated comparisons recursively walking down the tree until one reaches a terminal node. When the enumerated subset is empty or contains only a single class, classification stops. Either classification has failed (there was no matching class, or one has presumably found the indicated class. At that point, every bit in the virtual bit string must be compared to the classifier; classification is accepted on a perfect match.

In the example in Figure 2, if a packet {2001:db8:1:2:3:4:5:6, 2001:db8:2:3:4:5:6:7, AF41, 0} arrives, we start at the root. Since it is an AF41 packet, we deduce that case (1a) applies, and since the destination has 0001 in the third sixteen bit field of the destination address, we are comparing to {2001:db8:1::/48, ::/0, AF41, *}. Since the destination address is within 2001:db8:1::/48, classification as that succeeds.

Author's Address

Fred Baker
 Cisco Systems
 Santa Barbara, California 93117
 USA

Email: fred@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2013

F.J. Baker
Cisco Systems
February 17, 2013

Extensible OSPF LSAs
draft-baker-ipv6-ospf-extensible-00

Abstract

This note describes the changes necessary for OSPFv3 to route extensible classes of traffic. This implies not routing "to a destination", but "traffic matching a classification tuple" which includes a destination but may also include other attributes such as the source address, DSCP, or Flow Label.

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 21, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Theory of Routing	3
2.1. Dealing with ambiguity	3
3. Extensions necessary for OSPFv3	4
3.1. OSPF optional data extensions	4
3.1.1. IPv6 Destination Prefix TLV	4
3.1.2. IPv6 Forwarding Address TLV	5
3.1.3. Referenced Advertising Router TLV	5
3.1.4. Metric TLV	6
3.1.5. External Route Tag TLV	6
3.1.6. Referenced Link State ID TLV	7
3.2. OSPF extensible LSAs	7
3.2.1. Extensible-Inter-area-prefix-LSA	8
3.2.2. Extensible-AS-external-LSA	9
3.2.3. Extensible-Intra-Area-Prefix-LSA	9
4. IANA Considerations	10
5. Security Considerations	10
6. Privacy Considerations	11
7. Acknowledgements	11
8. Change Log	11
9. References	11
9.1. Normative References	11
9.2. Informative References	11
Appendix A. FIB Design	11
A.1. Linux Source-Address Forwarding	12
A.1.1. One FIB per source prefix	12
A.1.2. One FIB per source prefix plus a general FIB	13
A.2. PATRICIA	13
A.2.1. Virtual Bit String	13
A.2.2. Tree Construction	14
A.2.3. Tree Lookup	15
Author's Address	15

1. Introduction

In related documents, the author proposes extensions to OSPF and IS-IS for the routing of IPv6 traffic using more than the destination address as the definition of a class of traffic to be routed. These include the possibility of source/destination routing, and especially egress routing, routing based on the destination plus the DSCP value such as is discussed in [RFC4915], and routing using the destination plus the IPv6 Flow Label for a form of Role Based Access Control - if the sender doesn't know the flow label value that the receiver is using, which it would learn from the network administrator through

configuration, DHCP, or some other means, it in effect has no route to the destination.

These capabilities, in OSPFv3, are have as a premise an extensible LSA; an LSA that contains the necessary elements of any LSA as discussed in section 4.4.1 of [RFC5340], a destination address, and a set of options. This document describes extensible inter-area-prefix-LSAs, intra-area-prefix-LSAs, and AS-external-LSAs. Additional options are defined in other documents.

Existing OSPF LSAs that specify only a destination prefix may be understood as identifying a destination prefix and "any" other option, whether it be source address, flow label, or something else. This is also a useful class of traffic to compactly represent, so existing LSA types are not deprecated, merely added to.

1.1. Requirements Language

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

2. Theory of Routing

Both IS-IS and OSPF perform their calculations by building a routes from the router performing the calculation to each router, and then use those routes to get to destinations that those routes advertise connectivity to. Following the SPF algorithm, calculation starts by selecting a starting point (typically the router doing the calculation), and successively adding {link, router} pairs until one has calculated a route to every router in the network. As each router is added, including the original router, destinations that it is directly connected to are turned into routes in the route table: "to get to 2001:db8::/32, route traffic to {interface, list of next hop routers}". For immediate neighbors to the originating router, of course, there is no next hop router; traffic is handled locally.

2.1. Dealing with ambiguity

In any routing protocol, there is the possibility of ambiguity. An area border router might, for example, summarize the routes to other areas into a small set of relatively short prefixes, which have more specific routes within the area. Traditionally, we have dealt with that using a "longest match first" rule. If the same datagram matches more than one destination prefix advertised within the area, we follow the route to the longest matching prefix.

When routing a class of traffic, we follow an analogous "most specific match" rule; we follow the route for the most specific matching tuple. In cases of simple overlap, such as routing to 2001:db8::/32 or 2001:db8:1::/48, that is exactly analogous; we choose one of the two routes.

It is possible, however, to construct an ambiguous case in which neither class subsumes the other. For example, presume that

- o A is a prefix,
- o B is a more-specific prefix within A,
- o C is a different prefix, and
- o D is a more-specific prefix of C.

The two classes {A, D, *, *} and {B, C, *, *} are ambiguous: a datagram within {B, D, *, *} matches both classes, and it is not clear in the data plane what decision to make. Solving this requires the addition of a third route in the FIB corresponding to the class {B, D, *, *}, which is more-specific than either of the first two, and can be given routing guidance based on metrics or other policy in the usual way.

3. Extensions necessary for OSPFv3

Changing OSPF to provide for this type of change requires cloning many of the existing LSAs: the inter-area-prefix-LSAs, the AS-external-LSAs, and the intra-area-prefix LSA. This can be done specifically with the information we have thought about, or designed for extensibility. We choose extensibility.

3.1. OSPF optional data extensions

This section defines a number of optional type-length-value (TLV) information elements that may be included in an extensible LSA. In an extensible LSA, elements not included are not considered in classification and as a result are in effect wild-carded.

3.1.1. IPv6 Destination Prefix TLV

The IPv6 Destination Prefix TLV MAY be used with the IPv6 Source Prefix TLV, but MUST NOT be used with the IPv4 Source Prefix TLV or the IPv4 Destination Prefix TLV.

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1			

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      | Prefix Length  |      Prefix
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Destination Prefix TLV

Destination Prefix Type: assigned by IANA

TLV Length: Length of the TLV in octets

Prefix Length: Length of the prefix in bits, in the range 0..128

Prefix: (Destination prefix length + 7)/8 octets of prefix

3.1.2. IPv6 Forwarding Address TLV

The IPv6 Forwarding Address TLV is only used in the Extensible-AS-external-LSA, and is optional.

```

0                                     1                                     2                                     3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      | 128 bit IPv6 Address
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

IPv6 Forwarding Address TLV

Flow Label Type: assigned by IANA

TLV Length: Length of the TLV in octets

IPv6 Address: A fully qualified IPv6 address (128 bits). If included, data traffic for the advertised destination will be forwarded to this address. It MUST NOT be set to the IPv6 Unspecified Address (0:0:0:0:0:0:0:0) or an IPv6 Link-Local Address (Prefix FE80/10). While OSPFv3 routes are normally installed with link-local addresses, an OSPFv3 implementation advertising a forwarding address MUST advertise a global IPv6 address. This global IPv6 address may be the next-hop gateway for an external prefix or may be obtained through some other method (e.g., configuration).

3.1.3. Referenced Advertising Router TLV

```

0                                     1                                     2                                     3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      | Referenced Advertising Router
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Referenced Advertising Router TLV

Flow Label Type: assigned by IANA

TLV Length: Length of the TLV in octets

Referenced Link State ID: With the Referenced Link State ID TLV (Referenced LS Type and Referenced Link State ID), Identifies the router-LSA or network-LSA with which the IPv6 traffic classes should be associated. If Referenced LS Type is 0x2001, the prefixes are associated with a router-LSA, Referenced Link State ID should be 0, and Referenced Advertising Router should be the originating router's Router ID. If Referenced LS Type is 0x2002, the prefixes are associated with a network-LSA, Referenced Link State ID should be the Interface ID of the link's Designated Router, and Referenced Advertising Router should be the Designated Router's Router ID.

3.1.4. Metric TLV

```

0                                     1                                     2                                     3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      |      Metric      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| PrefixOptions | Information elements for the traffic class
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Metric TLV

Flow Label Type: assigned by IANA

TLV Length: Length of the TLV in octets

Metric: The cost of this traffic class. Expressed in the same units as the interface costs in router-LSAs.

Information Elements This information element will be followed by zero or more information elements that describe the traffic class. the traffic class will have been fully described when parsing reaches the end of the LSA or finds a new Metric TLV.

3.1.5. External Route Tag TLV

The External Route Tag TLV is only used in the Extensible-AS-external-LSA, and is optional.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      |      External Route Tag      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

External Route Tag TLV

Flow Label Type: assigned by IANA

TLV Length: Length of the TLV in octets

Route Tag: A 32-bit field that MAY be used to communicate additional information between AS boundary routers.

3.1.6. Referenced Link State ID TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      |      Referenced LS Type      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Referenced Link State ID                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Referenced Link State ID TLV

Flow Label Type: assigned by IANA

TLV Length: Length of the TLV in octets

Referenced LS Type: The LSType of the associate LSA.

Referenced Link State ID: If included, additional information concerning the advertised external route can be found in the LSA having LS type equal to "Referenced LS Type", Link State ID equal to "Referenced Link State ID", and Advertising Router the same as that specified in the Extensible-AS-external-LSA's link-state header. This additional information is not used by the OSPF protocol itself. It may be used to communicate information between AS boundary routers. The precise nature of such information is outside the scope of this specification.

3.2. OSPF extensible LSAs

This section defines the extensible Extensible-Inter-Area-Prefix-LSA, Extensible-AS-external-LSA, and Extensible-Intra-Area-Prefix LSA.

3.2.1. Extensible-Inter-area-prefix-LSA

Extensible-Inter-area-prefix-LSAs have LS type equal to [IANA?]. These LSAs are equivalent to OSPFv2's type 3 summary-LSAs (see Section 12.4.3 of [RFC2328]). Originated by area border routers, they describe IPv4 or IPv6 traffic classes that belong to other areas, and are encoded using the TLVs defined in Section 3.1. A separate inter-area-prefix-LSA is originated for each such traffic class. For details concerning the construction of inter-area-prefix-LSAs, see [RFC5340] Section 4.4.3.4.

For stub areas, inter-area-prefix-LSAs can also be used to describe a (per-area) default route. Default summary routes are used in stub areas instead of flooding a complete set of external routes. When describing a default summary route, the Extensible-inter-area-prefix-LSA omits the Destination Prefix information element, which has the same effect as matching 0.0.0.0/0 or ::/0.

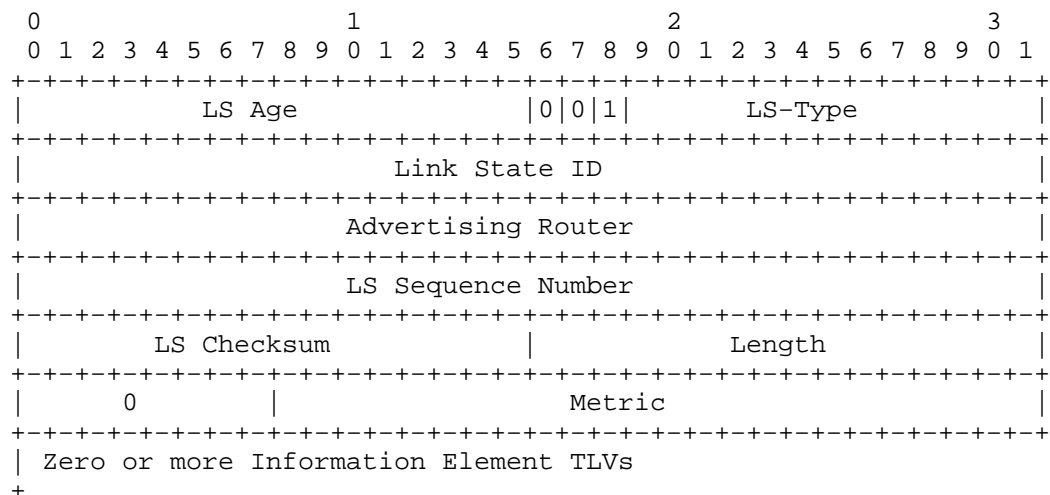


Figure 1: Extensible-Inter-area-prefix-LSA

LS-Type: To be assigned by IANA

Metric: The cost of this route. Expressed in the same units as the interface costs in router-LSAs. When the Extensible-inter-area-prefix-LSA is describing a route to a range of addresses (see [RFC5340] Appendix C.2), the cost is set to the maximum cost to any reachable component of the address range.

3.2.2. Extensible-AS-external-LSA

This is an AS-external-LSAs, but may include other information elements. Unlike the AS-external-LSAs, however, the presence of optional information is determined by the presence of the information elements, not by flags.

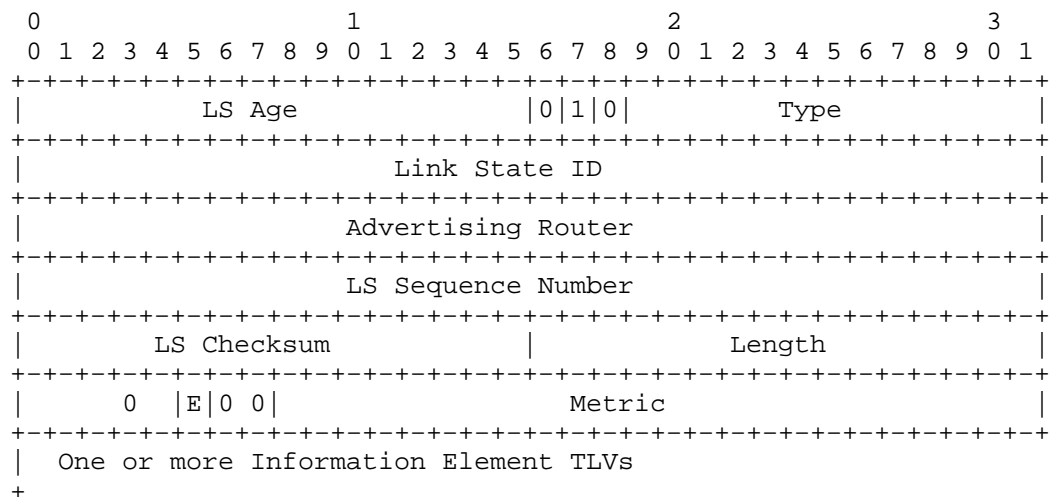


Figure 2: Extensible-AS-external-LSA

E: The type of external metric. If bit E is set, the metric specified is a Type 2 external metric. This means the metric is considered larger than any intra-AS path. If bit E is zero, the specified metric is a Type 1 external metric. This means that it is expressed in the same units as other LSAs (i.e., the same units as the interface costs in router-LSAs).

: The cost of this route. Interpretation depends on the external type indication (bit E above).

3.2.3. Extensible-Intra-Area-Prefix-LSA

This LSA MUST include a Referenced Link State ID TLV and a Referenced Advertising Router TLV immediately following the number of traffic classes. It MUST also include the indicated number of Metric TLVs, each of which is followed by the information elements that define that class of traffic, which will usually include a Destination Prefix TLV and may include a source prefix TLV, Flow Label TLV, or DSCP TLV.

Extensible-Intra-area-prefix-LSAs have LS types assigned by IANA. A router uses Extensible-intra-area-prefix-LSAs to advertise one or more traffic classes that are associated with a local router address, an attached stub network segment, or an attached transit network segment. In IPv4, the first two were accomplished via the router's router-LSA and the last via a network-LSA. In OSPF for IPv6, all addressing information that was advertised in router-LSAs and network-LSAs has been removed and is now advertised in intra-area-prefix-LSAs. For details concerning the construction of intra-area-prefix-LSA, see [RFC5340] Section 4.4.3.9.

A router can originate multiple extensible-intra-area-prefix-LSAs for each router or transit network. Each such LSA is distinguished by its unique Link State ID.

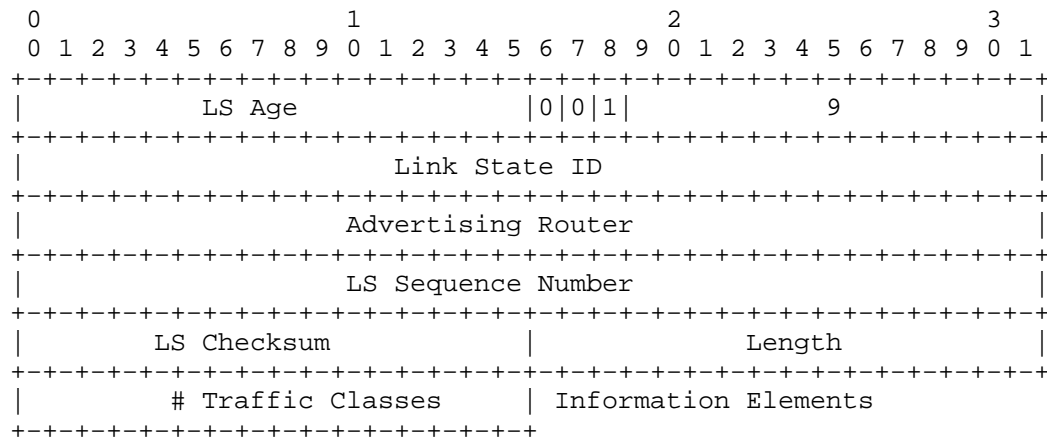


Figure 3: Extensible-Intra-Area-Prefix-LSA

Traffic Classes: The number of traffic classes that will be specified. Each traffic class has, first, a metric TLV, and then one or more other TLVs, normally including a Destination Prefix TLV.

4. IANA Considerations

This section will request LSID values for the LSAs defined, plus define a registry for optional fields. This is deferred to the -01 version of the draft.

5. Security Considerations

To be considered.

6. Privacy Considerations

To be considered.

7. Acknowledgements

8. Change Log

Initial Version: February 2013

9. References

9.1. Normative References

- [ISO.10589.1992]
International Organization for Standardization,
"Intermediate system to intermediate system intra-domain-
routing routine information exchange protocol for use in
conjunction with the protocol for providing the
connectionless-mode Network Service (ISO 8473)", ISO
Standard 10589, 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF
for IPv6", RFC 5340, July 2008.

9.2. Informative References

- [PATRICIA]
Morrison, D.R., "Practical Algorithm to Retrieve
Information Coded in Alphanumeric", Journal of the ACM
15(4) pp514-534, October 1968.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P.
Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC
4915, June 2007.

Appendix A. FIB Design

While the design of the Forwarding Information Base is not a matter for standardization, as it only has to work correctly, not interoperate with something else, the design of a FIB for this type of lookup may differ from approaches used in destination routing. We describe one possible approach that is known to work, from the perspective of a proof of concept.

A.1. Linux Source-Address Forwarding

The University of Waikato has added to the Linux Advanced Routing & Traffic Control facility the ability to maintain multiple FIBs, one for each of a set of prefixes. Implementing source/destination routing using this mechanism is not difficult.

The router must know what source prefixes might be used in its domain. This may be by configuration or, at least in concept, learned from the routing protocols themselves. In whichever way that is done, one can imagine two fundamental FIB structures to serve N source prefixes; N FIBs, one per prefix, or N+1 FIBs, one per prefix plus one for destinations for which the source prefix is unspecified.

A.1.1. One FIB per source prefix

In an implementation with one FIB per source prefix, the routing algorithm has two possibilities.

- o If it calculates a route to a prefix (such as a default route) associated with a given source prefix, it stores the route in the FIB for the relevant source prefix.
- o If it calculates a route for which the source prefix is unspecified, it stores that route in all N FIBs.

When forwarding a datagram, the IP forwarder looks at the source address of the datagram to determine which FIB it should use. If it is from an address for which there is no FIB, the forwarder discards the datagram as containing a forged source address. If it is from an address within one of the relevant prefixes, it looks up the destination in the indicated FIB and forwards it in the usual way.

The argument for this approach is simplicity: there is one place to look in making a forwarding decision for any given datagram. The argument against it is memory space; it is likely that the FIBs will be similar, but every destination route not associated with a source prefix is duplicated in each FIB. In addition, since it automatically removes traffic whose source address is not among the configured list, it limits the possibility of user software using improper addresses.

A.1.2. One FIB per source prefix plus a general FIB

In an implementation with N+1 FIBs, the algorithm is slightly more complex.

- o If it calculates a route to a prefix (such as a default route) associated with a given source prefix, it stores the route in the FIB for the relevant source prefix.
- o If it calculates a route for which the source prefix is unspecified, it stores that route in the FIB that is not associated with a source prefix.

When forwarding a datagram, the IP forwarder looks at the source address of the datagram to determine which FIB it should use. If it is from one of the configured prefixes, it looks the destination up in the indicated FIB. In any event it also looks the destination up in the "unspecified source address" FIB. If the destination is found in only one of the two, the indicated route is followed. If the destination is found in both, the more specific route is followed.

The argument for this approach is memory space; if a large percentage of routes are only in the general FIB, such as when egress routing is used for the default route and all other routes are internal, the other FIBs are likely to be very small - perhaps only a single default route. The argument against this approach is complexity: most lookups if not all will be done in a prefix-specific FIB and in the general FIB.

A.2. PATRICIA

One approach is a [PATRICIA] Tree. This is a relative of a Trie, but unlike a Trie, need not use every bit in classification, and does not need the bits used to be contiguous. It depends on treating the bit string as a set of slices of some size, potentially of different sizes. Slice width is an implementation detail; since the algorithm is most easily described using a slice of a single bit, that will be presumed in this description.

A.2.1. Virtual Bit String

It is quite possible to view the fields in a datagram header incorporated into the classification tuple as a virtual bit string such as is shown in Figure 4. This bit string has various regions within it. Some vary and are therefore useful in a radix tree lookup. Some may be essentially constant - all global IPv6 addresses at this writing are within 2000::/3, for example, so while it must be tested to assure a match, incorporating it into the radix tree may

not be very helpful in classification. Others are ignored; if the destination is a remote /64, we really don't care what the EID is. In addition, due to variation in prefix length and other details, the widths of those fields vary among themselves. The algorithm the FIB implements, therefore, must efficiently deal with the fact of a discontinuous lookup key.

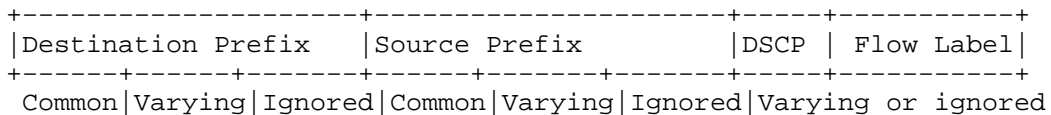


Figure 4: Treating a traffic class as a virtual bit string

A.2.2. Tree Construction

The tree is constructed by recursive slice-wise decomposition. At each stage, the input is a set of classes to be classified. At each stage, the result is the addition of a lookup node in the tree that identifies the location of its slice in the virtual bit string (which might be a bit number), the width of the slice to be inspected, and an enumerated set of results. Each result is a similar set of classes, and is analyzed in a similar manner.

The analysis is performed by enumerating which bits that have not already been considered are best suited to classification. For a slice of N bits, one wants to select a slide that most evenly divides the set of classes into 2^N subsets. If one or more bits in the slice is ignored in some of the classes, those classes must be included in every subset, as the actual classification of them will depend on other bits.

```

Input:{2001:db8::/32, ::/0, *, *}
      {2001:db8:1::/48, ::/0, AF41, *}
      {2001:db8:1::/48, ::/0, AF42, *}
      {2001:db8:1::/48, ::/0, AF43, *}
Common parts: Destination prefix 2001:dba, source prefix, and label
Varying parts: DSCP and the third set of sixteen bits in the
                destination prefix
One possible decomposition:
(1) slice = DSCP
    enumerated cases:
(a) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF41, *} }
(b) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF42, *} }
(c) { {2001:db8::/32, ::/0, *, *}, {2001:db8:1::/48, ::/0, AF43, *} }
(2) slice = third sixteen bit field in destination
    This divides each enumerated case into those containing 0001 and
    "everything else", which would imply 2001:db8::/32

```

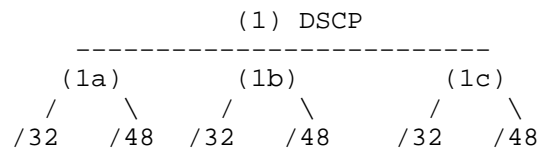


Figure 5: Example PATRICIA Tree

A.2.3. Tree Lookup

To look something up in a PATRICIA Tree, one starts at the root of the tree and performs the indicated comparisons recursively walking down the tree until one reaches a terminal node. When the enumerated subset is empty or contains only a single class, classification stops. Either classification has failed (there was no matching class, or one has presumably found the indicated class. At that point, every bit in the virtual bit string must be compared to the classifier; classification is accepted on a perfect match.

In the example in Figure 5, if a packet {2001:db8:1:2:3:4:5:6, 2001:db8:2:3:4:5:6:7, AF41, 0} arrives, we start at the root. Since it is an AF41 packet, we deduce that case (1a) applies, and since the destination has 0001 in the third sixteen bit field of the destination address, we are comparing to {2001:db8:1::/48, ::/0, AF41, *}. Since the destination address is within 2001:db8:1::/48, classification as that succeeds.

Author's Address

Fred Baker
 Cisco Systems
 Santa Barbara, California 93117
 USA

Email: fred@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 29, 2013

C. Grundemann
C. Donley
CableLabs
J. Brzozowski
Comcast Cable Communications
L. Howard
Time Warner Cable
V. Kuarsingh
Rogers Communications
February 25, 2013

A Near Term Solution for Home IP Networking (HIPnet)
draft-grundemann-homenet-hipnet-01

Abstract

Home networks are becoming more complex. With the launch of new services such as home security, IP video, Smart Grid, etc., many Service Providers are placing additional IPv4/IPv6 routers on the subscriber network. This document describes a self-configuring home router that is capable of operating in such an environment, and that requires no user interaction to configure it. Compliant with draft-ietf-homenet-arch, it uses existing protocols in new ways without the need for a routing protocol.

Requirements Language

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

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 29, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
1.1. Requirements Language	4
2. Terminology	4
3. Architecture	5
3.1. Current End-User Network Architecture	5
3.2. HIPNet End-User Network Architecture	6
4. Network Detection	8
4.1. Edge Detection	8
4.2. Directionless Home Routers	9
5. Routing and Addressing	10
5.1. Recursive Prefix Delegation	11
5.2. Prefix Sub-Delegation Requirements	12
5.3. Multiple Address Family Support	13
5.4. Hierarchical Routing	13
6. Multiple ISPs	13
6.1. Backup Connection	14
6.2. Multi-homing	14
6.2.1. Multihoming Requirements	16
7. Multicast Support	17
7.1. Service Discovery	17
7.2. Multicast Proxy Support	17
7.3. Multicast Requirements	17
8. Firewall Support	18
8.1. Requirements	19
9. IANA Considerations	19
10. Security Considerations	19
11. Acknowledgements	19
12. References	20
12.1. Normative References	20
12.2. Informative References	20

Authors' Addresses	22
------------------------------	----

1. Introduction

This document expands upon [I-D.ietf-v6ops-6204bis] to describe IPv6/IPv4 features for a residential or small-office router, referred to as a HIPnet router. Consistent with [I-D.ietf-homenet-arch], it focuses on network technology evolution to support increasingly large residential/SoHo networks. While the primary focus is on IPv6 support, this document also describes how to leverage IPv6 to configure IPv4 in a manner better than nested NATs in operation on many networks today.

This document specifies how a HIPnet router automatically detects both the edge of the customer network and its upstream interface, how it subdivides an IPv6 prefix to distribute to downstream routers, and how it leverages IPv6 address assignment to distribute IPv4 addresses. It also discusses how such a router can operate with a backup ISP or limited multihoming across two ISPs.

1.1. Requirements Language

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

2. Terminology

End-User Network	one or more links attached to the HIPnet router that connect IPv6 and IPv4 hosts.
Home IP Network (HIPnet) Router	a node intended for home or small-office use that forwards packets not explicitly addressed to itself.
Customer Edge Router (CER)	a HIPnet router that connects the end-user network to a service provider network.
Internal Router	an additional HIPnet router deployed in the home or small-office network that is not attached to a service provider network. Note that this is a functional role; it is expected that there will not be a difference in hardware or software between a CER and IR, except in such cases when a CER has a dedicated non-Ethernet WAN interface (e.g. DSL/cable/ LTE modem) that would preclude it from operating as an IR.

Down Interface	a HIPnet router's attachment to a link in the end-user network on which it distributes addresses and/or prefixes. Examples are Ethernet (simple or bridged), 802.11 wireless, or other LAN technologies. A HIPnet router may have one or more network-layer down interfaces.
downstream router	a router directly connected to a HIPnet router's Down Interface.
Service Provider	an entity that provides access to the Internet. In this document, a service provider specifically offers Internet access using IPv6, and may also offer IPv4 Internet access. The service provider can provide such access over a variety of different transport methods such as DSL, cable, wireless, and others.
Up Interface	a HIPnet router's attachment to a link where it receives one or more IP addresses and/or prefixes. This is also the link to which the HIPnet router points its default route.
depth	the number of layers of routers in a network. A single router network would have a depth of 1, while a router behind a router behind a router would have a depth of 3.
width	The number of routers that can be directly subtended to an upstream router. A network with three directly attached routers behind the CER would have a width of 3.

3. Architecture

3.1. Current End-User Network Architecture

An end-user network will likely support both IPv4 and IPv6. A typical end-user network consists of a "plug and play" router with IPv4 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.

Rewriting addresses on the edge of the network allows for some rudimentary multihoming, even though using NATs for multihoming does not preserve connections during a fail-over event [RFC4864].

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

3.2. HIPNet End-User Network Architecture

The end-user network architecture should provide equivalent or better capabilities and functionality than the current architecture. However, as end-user networks become more complex, the HIPnet architecture needs to support more complicated networks. Figure 1 illustrates the model topology for the end-user network.

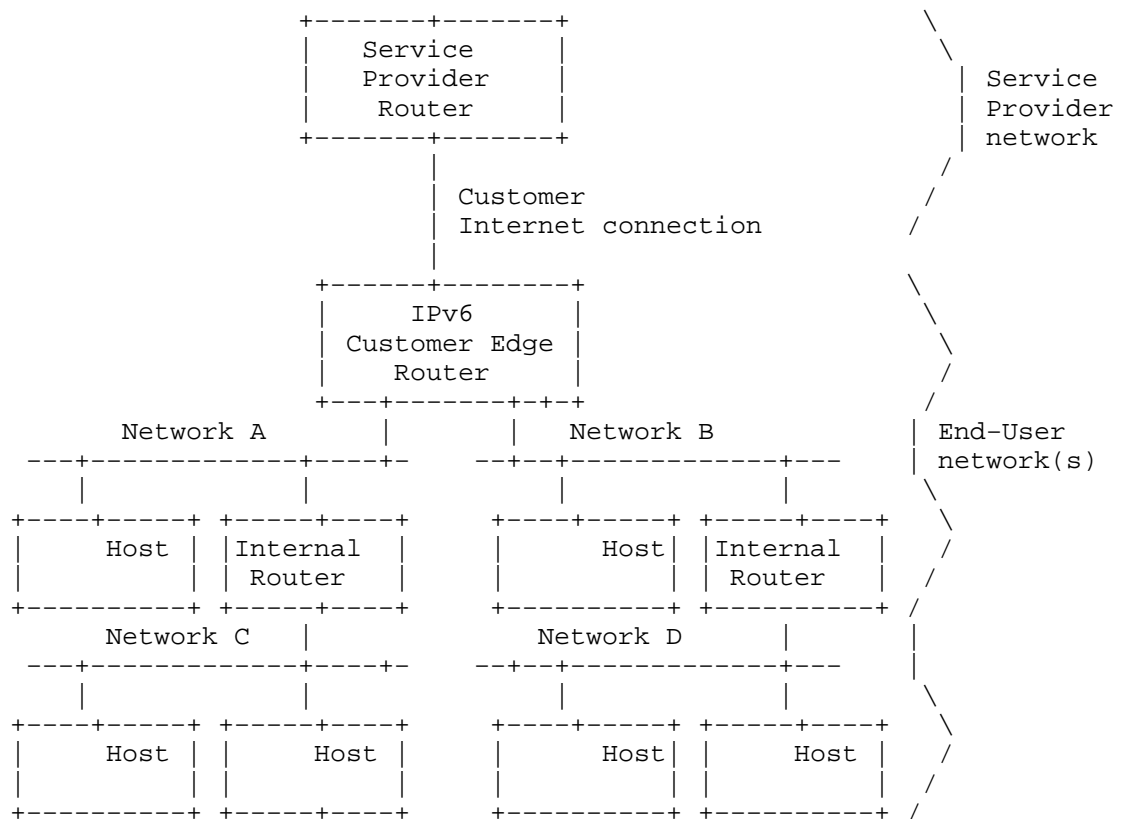


Figure 1: Example End-User Network

This architecture describes the following:

- o Prefix subdelegation supporting multiple subnets and routers
- o Border Detection
- o Router directionality supporting a hierarchical network
- o Multicast forwarding rules to support common service discovery protocols

While routers described in this document may be manually configured in an arbitrary topology with or without a dynamic routing protocol, this document only addresses automatic provisioning and configuration.

4. Network Detection

In multirouter home networks, routers have to determine where they fit in the topology - whether they are at the edge or internal, and which interface is up (that is, which interface points out of the network).

4.1. Edge Detection

Customer Edge Routers (CER) will often be required to behave differently from Internal Routers (IR) in several capacities. Some examples include: Firewall settings, IPv4 NAT, ULA generation (if supported), name services, multicast forwarding differences, and others. This is a functional role, and will not typically be differentiated by hardware/software (i.e. end users will not purchase a specific CER model of router distinct from IR models).

There are three methods that a router can use to determine if it is a CER for its given network:

"/48 Check" Service providers will provide IPv6 WAN addresses (DHCPv6 IA_NA) and IPv6 prefixes (DHCPv6 IA_PD) from different pools of addresses. The largest IPv6 prefix that we can expect to be delegated to a home router is a /48. Combining these two observations, a home router can compare the WAN address assigned to it with the prefix delegated to it to determine if it is attached directly to a service provider network. If the router is a CER, the WAN address will be from a different /48 than the prefix. If the router is an IR, the WAN address will be from the same /48 as the prefix. In this way, the router can determine if it is receiving an "external" prefix from a service provider or an "internal" prefix from another home router.

CER_ID A home router can use the CER_ID DHCPv6 option defined in [I-D.donley-dhc-cer-id-option] to determine if it is a CER or an IR. ISPs will not set the CER_ID option, but the first CPE router sets its address in the option and other routers forward the completed CER_ID to subdelegated routers.

Physical Some routers will have a physical differentiator built into them by design that will indicate that they are a CER. Examples include mobile routers, DSL routers, and cable eRouters. In the case of a mobile router, the presence of an active cellular connection indicates that the router is at the customer edge. Likewise, for an eRouter, the presence of an active DOCSIS(R) link tells the router that it is at the customer edge.

HIPnet routers can (and likely will) use more than one of the above

techniques in combination to determine the edge. For example, an internal router will check for the CER_ID option, but will also use the 48 check in case its upstream router does not support CER_ID.

4.2. Directionless Home Routers

As home networks grow in complexity and scale, it will become more common for end users to make mistakes with the physical connections between multiple routers in their home or small office. This is likely to produce loops and improper uplink connections. While we can safely assume that home networks will become more complex over time, we cannot make the same assumption of the users of home networks. Therefore, home routers will need to mitigate these physical topology problems and create a working multi-router home network dynamically, without any end user intervention.

Legacy home routers with a physically differentiated uplink port are "directional;" they are pre-set to route from the 'LAN' or Internal ports to a single, pre-defined uplink port labeled "WAN" or "Internet". This means that an end-user can make a cabling mistake which renders the router unusable (e.g. connecting two router's uplink ports together). On the other hand, in enterprise and service provider networks, routers are "directionless;" that is to say they do not have a pre-defined 'uplink' port. While directional routers have a pre-set routing path, directionless routers are required to determine routing paths dynamically. Dynamic routing is often achieved through the implementation of a dynamic routing protocol, which all routers in a given network or network segment must support equally. This section introduces an alternative to dynamic routing protocols (such as OSPF) for creating routing paths on the fly in directionless home routers.

Note that some routers (e.g. those with a dedicated wireless/DSL/DOCSIS WAN interface) may continue to operate as directional routers. The HIPnet mechanism described below is intended for general-purpose routers.

The HIPnet mechanism uses address acquisition as described in [I-D.ietf-v6ops-6204bis] and various tiebreakers to determine directionality (up vs. down) and by so doing, creates a logical hierarchy (cf. [I-D.chakrabarti-homenet-prefix-alloc]) from any arbitrary physical topology:

1. After powering on, the HIPnet router sends Router Solicitations (RS) ([RFC4861] on all interfaces (except Wi-Fi*))

2. Other routers respond with Router Advertisements (RA)
3. Router adds any interface on which it receives an RA to the candidate 'up' list
4. The router initiates DHCPv6 PD on all candidate 'up' interfaces. If no RAs are received, the router generates a /48 ULA prefix.
5. The router evaluates the offers received (in order of preference):
 - a) Valid GUA preferred (preferred/valid lifetimes >0)
 - b) Internal prefix preferred over external (for failover - see Section [6.1])
 - c) Largest prefix (e.g. /56 preferred to /60)
 - d) Link type/bandwidth (e.g. Ethernet vs. MoCA)
 - e) First response (wait 1 s after first response for additional offers)
 - f) Lowest numerical prefix
6. The router chooses the winning offer as its Up Interface.

Once directionality is established, the router continues to listen for RAs on all interfaces but doesn't acquire addresses on Down Interfaces. If the router initially receives only a ULA address on its Up Interface and GUA addressing becomes available on one of its Down Interfaces, it restarts the process. If the router stops receiving RAs on its Up Interface, it restarts the process.

In all cases, the router's Up Interface becomes its uplink interface; the router acts as a DHCP client on this interface. The router's remaining interfaces are Down Interfaces; it acts as a DHCP server on these interfaces. Also, per [I-D.ietf-v6ops-6204bis], the router only sends RAs on Down Interfaces.

*Note: By default, Wi-Fi interfaces are considered to point "down." This requires manual configuration to enable a wireless uplink, which is preferred to avoid accidental or unwanted linking with nearby wireless networks.

5. Routing and Addressing

HIPnet routers use DHCPv6 prefix sub-delegation ([RFC3633]) to

recursively build a hierarchical network ([I-D.chakrabarti-homenet-prefix-alloc]). This approach requires no new protocols to be supported on any home routers.

Default router settings: Only CER instantiates guest network. Wifi defaults to 'down' direction, default route uses wired interface. Firewall considers Wifi an inside port. Wi-Fi bridged with first wired Down Interface.

5.1. Recursive Prefix Delegation

Once directionality is established, the home router will acquire a WAN IPv6 address and an IPv6 prefix per [I-D.ietf-v6ops-6204bis]. As HIPnet routers (other than the CER) do not know their specific location in the hierarchical network, all HIPnet routers use the same generic rules for recursive prefix delegation to facilitate route aggregation, multihoming, and IPv4 support (described below). This methodology expounds upon that previously described in [I-D.chakrabarti-homenet-prefix-alloc].

The process can be illustrated in the following way:

1. Per [I-D.ietf-v6ops-6204bis], the HIPnet router assigns a separate /64 from its delegated prefix(es) for each of its Down Interfaces in numerical order, starting from the numerically lowest.
2. If the received prefix is too small to number all Down Interfaces, the router collapses them into a single interface, assigns a single /64 to that interface, and logs an error message.
3. The HIPnet router subdivides the IPv6 prefix received via DHCPv6 ([RFC3315]) into sub-prefixes. To support a suggested depth of three routers, with as large a width as possible, it is recommended to divide the prefix on 2-, 3-, or 4-bit boundaries. If the received prefix is not large enough, it is broken into as many /64 sub-prefixes as possible and logs an error message. By default, this document suggests that the router divide the delegated prefix based on the aggregate prefix size and the HIPnet router's number of physical Down Interfaces. This is to allow for enough prefixes to support a downstream router on each down port.
 - * If the received prefix is smaller than a /56 (e.g. a /60),
 - + 8 or more port routers divide on 3-bit boundaries (e.g. /63).

- + 7 or fewer port routers divide on 2-bit boundaries (e.g. /62).
 - * If the received prefix is a /56 or larger,
 - + 8 or more port routers divide on 4-bit boundaries (e.g. /60).
 - + 7 or fewer port routers divide on 3-bit boundaries (e.g. /59).
4. The HIPnet router delegates remaining prefixes to downstream routers per [RFC3633] in reverse numerical order, starting with the numerically highest. This is to minimize the renumbering impact of enabling an inactive interface.

For example, a four port router with two LANs (two Down Interfaces) that receives 2001:db8:0:b0::/60 would start by numbering its two Down Interfaces with 2001:db8:0:b0::/64 and 2001:db8:0:b1::/64 respectively, and then begin prefix delegation by giving 2001:db8:0:bc::/62 to the first directly attached downstream router.

5.2. Prefix Sub-Delegation Requirements

- PSD-1: The HIPnet router MUST support [I-D.ietf-v6ops-6204bis] address acquisition and LAN addressing.
- PSD-2: The HIPnet router MUST support Delegating Router behavior for the IA-PD Option [RFC3633] on all Down Interfaces.
- PSD-3: HIPnet routers MUST NOT act as both a DHCP client and server on the same link.
- PSD-4: The HIPnet router MAY support other methods of dividing the received prefix.
- PSD-5: The HIPnet router MUST delegate prefixes of the same size to downstream routers.
- PSD-6: Per [I-D.ietf-v6ops-6204bis] L-2, the HIPnet router allocates a /64 to each Down Interface. The HIPnet router SHOULD allocate these /64 interface-prefixes in numerical order, starting with the lowest.
- PSD-7: If there are insufficient /64s for each Down Interface, the HIPnet router SHOULD assign the lowest numbered /64 for all Down Interfaces and log an error message.

- PSD-8: The HIPnet router MAY reserve additional /64 interface-prefixes for interfaces that will be enabled in the future.
- PSD-9: The HIPnet router SHOULD delegate sub-prefixes to downstream routers starting from the numerically highest sub-prefix and working down in reverse numerical order.
- PSD-10: If there are not enough sub-prefixes remaining to delegate to all downstream routers, the HIPnet router SHOULD log an error message.

5.3. Multiple Address Family Support

The recursive prefix delegation method described above can be extended to support additional address types such as IPv4, additional GUAs, or ULAs. When the HIPnet router receives its prefix via DHCPv6 ([RFC3633]), it computes its 8-bit router ID (bits 56-64) from the received IA_PD. It then prepends additional prefixes received in one or more IPv6 Router Advertisements ([RFC4861]) or from the DHCPv4-assigned ([RFC2131]) IPv4 network address received on the Up Interface.

As the network is hierarchical, upstream routers know the router ID for each downstream router, and know the prefix(es) on each LAN segment. Accordingly, HIPnet routers automatically calculate downstream routes to all downstream routers.

In networks using this mechanism for IPv4 provisioning, it is suggested that the CER use addresses in the 10.0.0.0/8 ([RFC1918]) range for downstream interface provisioning.

5.4. Hierarchical Routing

The recursive prefix delegation method described above, coupled with "up detection", enables very simple hierarchical routing. By this we mean that each router installs a single default 'up' route and a more specific 'down' route for each prefix delegated to a downstream IR. Each of these 'down' routes simply points all packets destined to a given prefix to the WAN IP address of the router to which that prefix was delegated. This combination of a default 'up' route and more specific 'down' routes provides complete reachability within the home network with no need for any additional message exchange or routing protocol support.

6. Multiple ISPs

HIPnet routers can support either active/standby multihoming with

multiple ISPs or limited active/active multihoming without a routing protocol.

6.1. Backup Connection

Using the procedure described above, multi-router home networks with multiple ISP connections can easily operate in an active/standby manner, switching from one Internet connection to the other when the active connection fails. Lacking a default priority, HIPnet routers will have to default to a "first online" method of primary CER selection. In other words, by default, the first CER to come online becomes the primary CER and the second CER to turn on becomes the backup. In this text, the primary ISP is the ISP connected to the primary CER and the backup ISP is simply the ISP attached to the backup CER.

In an active/standby multi-ISP scenario, a backup CER sets its Up Interface to point to the primary CER, not the backup ISP. Hence, it does not acquire or advertise the backup ISP prefix. Instead, it discovers the internally advertised GUA prefix being distributed by the currently active primary CER.

In the case of a primary ISP failure, per [I-D.ietf-v6ops-6204bis], the CER sends an RA advertising the preferred lifetime as 0 for the ISP-provided prefix, and its router lifetime as 0. The backup CER becomes active when it sees the primary ISP GUA prefix advertised with a preferred lifetime of 0. In the case of CER failure, if the backup CER sees the Primary CER stop sending RAs altogether, the Backup CER becomes active.

When the backup CER becomes active, it obtains and advertises its own external GUA. When advertising the GUA delegated by its ISP, the backup CER sets the valid, preferred, and router lifetimes to a value greater than 0. Other routers see this and re-determine the network topology via "up" detection, placing the new CER at the root of the new hierarchical tree.

Using this approach, manual intervention may be required to transition back to the primary ISP. This prevents flapping in the event of intermittent network failures. Another alternative is to have a user-defined priority, which would facilitate pre-emption.

6.2. Multi-homing

The HIPnet algorithm also allows for limited active/active multihoming in two cases:

1. When one ISP router is used as the primary connection and the second ISP router is used for limited connectivity e.g. for a home office
2. When both ISP routers are connected to the same LAN segment at the top of the tree.

In case 1, the subscriber has a primary ISP connection and a secondary connection used for a limited special purpose. (e.g. for work VPN, video network, etc.). Devices connected under the secondary network router access the Internet through the secondary ISP. All devices still have access to all network resources in the home. Devices under the secondary connection can use the primary ISP if the secondary fails, but other devices do not use the secondary ISP.

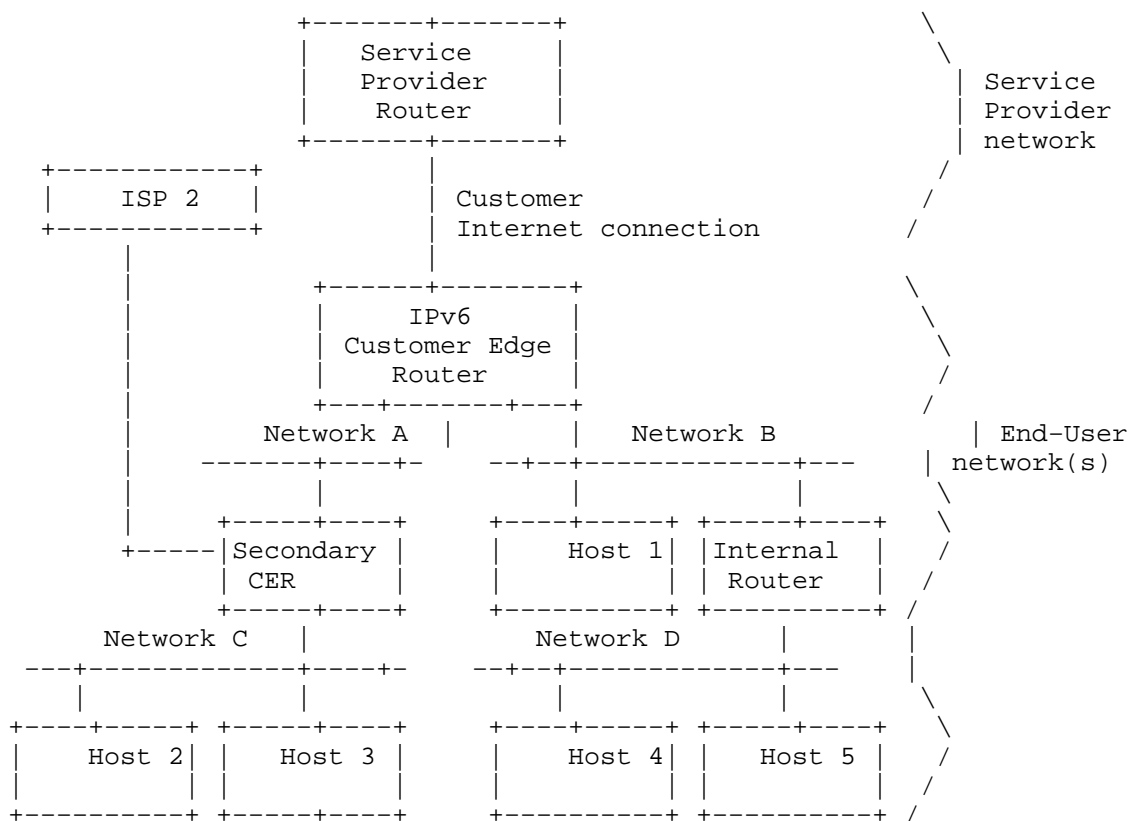


Figure 2: An Example of a multihomed End-User Network

As described above, the primary CER performs prefix sub-delegation to

create the hierarchical tree network. The secondary edge router then obtains a second prefix from ISP2 and advertises the ISP2 prefix as part of its RA. The Secondary CER thus includes sub-prefixes from both ISPs in all IA_PD messages to downstream routers with the same "router id.". In a change from the single-homing (or backup router) case, the secondary CER points its default route to ISP2, and adds an internal /48 route to its upstream internal router (e.g. R1). Devices below the the secondary CER (e.g. Host 2, Host 3) use ISP2, but have full access to all internal devices using the ISP1 prefix (and/or ULAs). If the ISP2 link fails, the secondary CER points its default route 'up' and traffic flows to ISP1. Devices not below the secondary CER (e.g. Hosts 1, 4, 5) use ISP1, but have full access to all internal devices using the ISP1 prefix (or ULAs).

In case 2, the secondary CER is installed on the same LAN segment as the primary CER. As above, it acquires a prefix from both the CER and secondary ISP. Since it is on the same LAN segment as the CER, the secondary CER does not delegate prefixes to that interface via DHCP. However, it does generate an RA for the ISP2 prefix on the LAN.

As described above, downstream routers receiving the secondary CER RA acquire an address using SLAAC and generate a prefix for sub-delegation by prepending the secondary CER prefix with the router ID generated during the receipt of the prefix from the CER. Such routers then generate their own RAs on downstream interfaces and include the secondary prefix as an IA_PD option in future prefix delegations.

6.2.1. Multihoming Requirements

- MH-1: HIPnet routers configured for active multi-ISP support MUST support DHCP address/prefix acquisition (per [I-D.ietf-v6ops-6204bis] on two interfaces (their WAN and upstream LAN interfaces).
- MH-2: HIPnet routers configured for active multi-ISP support MAY route packets based on the source IP address of incoming packets using [RFC6724] logic. This allows traffic sourced from the first ISP prefix to be directed to the first ISP, and traffic sourced from the second ISP prefix to be directed to the second ISP.
- MH-3: HIPnet routers configured for active multi-ISP support MUST advertise RAs for prefixes on all interfaces except the one from which the prefix was acquired or one directly attached to a Service Provider network.

7. Multicast Support

7.1. Service Discovery

There are several common service discovery protocols such as mDNS [I-D.cheshire-dnsext-multicastdns]/DNS-SD [I-D.cheshire-dnsext-dns-sd] and SSDP [SSDP]. In a multirouter network, service discovery needs to work across the entire home network (e.g. site-scoped rather than link-scoped). This requires that HIPnet routers forward relevant multicast traffic appropriately, to enable service discovery across the home network.

7.2. Multicast Proxy Support

In addition to multicast support for service discovery, it is recommended that HIPnet routers support external multicast traffic.

7.3. Multicast Requirements

- MULTI-1: A HIPnet router MUST discard IP multicast packets that fail a Reverse Path Forwarding Check (RPFC).
- MULTI-2: A HIPnet router that determines itself to be at the edge of a home network (e.g. via CER_ID option, /48 verification, or other mechanism) MUST NOT forward IPv4 administratively scoped (239.0.0.0/8) packets onto the WAN interface.
- MULTI-3: HIPnet Routers MUST forward IPv4 Local Scope multicast packets (239.255.0.0/16) to all LAN interfaces except the one from which they were received.
- MULTI-4: A HIPnet router that determines itself to be at the edge of a home network (e.g. via CER_ID option, /48 verification, or other mechanism) MUST NOT forward site-scope (FF05::) IPv6 multicast packets onto the WAN interface.
- MULTI-5: HIPnet routers MUST forward site-scoped (FF05::/16) IPv6 multicast packets to all LAN interfaces except the one from which they were received.
- MULTI-6: A home router MAY discard IP multicast packets sent between Down Interfaces (different VLANs).
- MULTI-7: HIPnet routers SHOULD support an IGMP/MLD proxy, as described in [RFC4605].

8. Firewall Support

In a home network, routers need to be equipped with stateful firewall capabilities. Home routers will need to provide "on by default" security where incoming traffic is limited to return traffic resulting from outgoing packets. They also need to allow users to create inbound 'pinholes' for specific purposes, such as online gaming, manually similar to those described in Simple Security ([RFC6092]). "Advanced Security" [I-D.vyncke-advanced-ipv6-security] features optionally could be added to provide intrusion detection (IDS/IPS) support.

Local Network Protection for IPv6 [RFC4864] recommends firewall functions that replace NAT security and calls for simple security. Simple Security [RFC6092] defines firewall filtering rules for IPv6 traffic. Advanced Security [I-D.vyncke-advanced-ipv6-security] supports the concept of end-to-end IPv6 reachability and uses adaptive filtering based on Intrusion Prevention System (IPS) functions.

It is recommended that the CER enable a stateful [RFC6092] firewall by default. IRs have three options described below:

IR Firewall Option 1 - Filtering Disabled: Once a home router determines that it is not the CER, it disables its firewall and allows all traffic to pass. The advantages of this approach are that it is simple and easy to troubleshoot and it facilitates whole-home service discovery and media sharing. The disadvantages are that it does not protect home devices from each other (e.g. infected machines could affect entire home network).

IR Firewall Option 2 - Simple Security + PCP: Home routers have a [RFC6092] firewall on by default, regardless of CER/IR status but IRs allow "pin-holing" using PCP [I-D.ietf-pcp-base]. CERs can restrict opening PCP pinholes on the up interface. The advantages of this approach are that it protects the home network from internal threats in other LAN segments and it mirrors legacy IPv4 router behavior. The disadvantages to this approach are that it is less predictable; it relies on application "pin-holing", a "default deny" rule that may interfere with service discovery and/or content sharing, and requires PCP clients (e.g. on PCs and CPE devices).

IR Firewall Option 3 - Advanced Security: Once a home router determines that it is not the CER, it disables its [RFC6092] firewall but activates an [I-D.vyncke-advanced-ipv6-security] firewall (IPS). The advantages to this approach are that it protects the home network from internal threats in other segments and is more predictable than Option 2, since internal traffic is allowed by default. The

disadvantages are that adaptive filtering is more complex than static filtering and typically requires a "fingerprint" subscription to work well.

It is recommended that dual-stack routers configure IPv4 support to mirror IPv6, as described above.

While this section describes default router behavior, device manufacturers are encouraged to make router security options user-configurable.

8.1. Requirements

SEC-1: The CER MUST enable a stateful [RFC6092] firewall by default.

SEC-2: HIPnet routers MUST only perform IPv4 NAT when serving as the CER.

SEC-3: By default, HIPnet routers SHOULD configure IPv4 firewalling rules to mirror IPv6.

SEC-4: HIPnet routers serving as CER SHOULD NOT enable UPnP IGD ([UPnP-IGD]) control by default.

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

Security considerations are discussed in the Firewall Support section above.

11. Acknowledgements

TBD

12. References

12.1. Normative References

- [I-D.ietf-v6ops-6204bis]
Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", draft-ietf-v6ops-6204bis-12 (work in progress), October 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [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, August 2006.
- [RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, May 2007.
- [RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.

12.2. Informative References

- [I-D.chakrabarti-homenet-prefix-alloc]
Nordmark, E., Chakrabarti, S., Krishnan, S., and W. Haddad, "Simple Approach to Prefix Distribution in Basic Home Networks", draft-chakrabarti-homenet-prefix-alloc-01 (work in progress), October 2011.
- [I-D.cheshire-dnsext-dns-sd]
Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-11 (work in progress), December 2011.
- [I-D.cheshire-dnsext-multicastdns]
Cheshire, S. and M. Krochmal, "Multicast DNS",

draft-cheshire-dnsexst-multicastdns-15 (work in progress),
December 2011.

[I-D.donley-dhc-cer-id-option]

Donley, C. and C. Grundemann, "Customer Edge Router
Identification Option", draft-donley-dhc-cer-id-option-01
(work in progress), September 2012.

[I-D.ietf-homenet-arch]

Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil,
"Home Networking Architecture for IPv6",
draft-ietf-homenet-arch-07 (work in progress),
February 2013.

[I-D.ietf-pcp-base]

Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P.
Selkirk, "Port Control Protocol (PCP)",
draft-ietf-pcp-base-29 (work in progress), November 2012.

[I-D.vyncke-advanced-ipv6-security]

Vyncke, E., Yourtchenko, A., and M. Townsley, "Advanced
Security for IPv6 CPE",
draft-vyncke-advanced-ipv6-security-03 (work in progress),
October 2011.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and
E. Lear, "Address Allocation for Private Internets",
BCP 5, RFC 1918, February 1996.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
RFC 2131, March 1997.

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

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

[SSDP] UPnP Forum, "Universal Plug and Play (UPnP) Device
Architecture 1.1", October 2008, <<http://www.upnp.org/>>.

[UPnP-IGD]

UPnP Forum, "Universal Plug and Play (UPnP) Internet
Gateway Device (IGD)", November 2001,
<<http://www.upnp.org/>>.

Authors' Addresses

Chris Grundemann
CableLabs
858 Coal Creek Circle
Louisville, CO 80027
USA

Phone: +1-303-351-1539
Email: c.grundemann@cablelabs.com

Chris Donley
CableLabs
858 Coal Creek Circle
Louisville, CO 80027
USA

Email: c.donley@cablelabs.com

John Jason Brzozowski
Comcast Cable Communications
1306 Goshen Parkway
Chester, PA 19380
USA

Email: john_brzozowski@cable.comcast.com

Lee Howard
Time Warner Cable
13241 Woodland Park Rd
Herndon, VA 20171
USA

Email: william.howard@twcable.com

Victor Kuarsingh
Rogers Communications
8200 Dixie Road
Brampton, ON L6T 0C1
Canada

Email: victor.kuarsingh@rci.rogers.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 8, 2013

A. Lindem
J. Arkko
Ericsson
October 5, 2012

OSPFv3 Auto-Configuration
draft-ietf-ospf-ospfv3-autoconfig-00.txt

Abstract

OSPFv3 is a candidate for deployments in environments where auto-configuration is a requirement. One such environment is the IPv6 home network where users expect to simply plug in a router and have it automatically use OSPFv3 for intra-domain routing. This document describes the necessary mechanisms for OSPFv3 to be self-configuring.

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 April 8, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
1.1. Requirements notation	3
1.2. Acknowledgments	3
2. OSPFv3 Default Configuration	4
2.1. Wait Timer Reduction	4
3. OSPFv3 Router ID Selection	6
4. OSPFv3 Adjacency Formation	7
5. OSPFv3 Duplicate Router-ID Detection and Resolution	8
5.1. Duplicate Router-ID Detection for Neighbors	8
5.2. Duplicate Router-ID Detection for OSPFv3 Routers that are not Neighbors	8
5.2.1. OSPFv3 Router Auto-Configuration LSA	8
5.2.2. Router-Hardware-Fingerprint TLV	10
5.3. Duplicate Router-ID Resolution	10
5.4. Change to Received Self-Originated LSA Processing	11
6. Security Considerations	12
7. Management Considerations	13
8. IANA Considerations	14
9. References	15
9.1. Normative References	15
9.2. Informative References	15
Authors' Addresses	16

1. Introduction

OSPFv3 [OSPFV3] is a candidate for deployments in environments where auto-configuration is a requirement. Its operation is largely unchanged from the base OSPFv3 protocol specification [OSPFV3].

The following aspects of OSPFv3 auto-configuration are described:

1. Default OSPFv3 Configuration
2. Unique OSPFv3 Router-ID generation
3. OSPFv3 Adjacency Formation
4. Duplicate OSPFv3 Router-ID Resolution

1.1. Requirements notation

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-KEYWORDS].

1.2. Acknowledgments

This specification was inspired by the work presented in the Homenet working group meeting in October 2011 in Philadelphia, Pennsylvania. In particular, we would like to thank Fred Baker, Lorenzo Colitti, Ole Troan, Mark Townsley, and Michael Richardson.

Arthur Dimitrelis and Aidan Williams did prior work in OSPFv3 auto-configuration in the expired "Autoconfiguration of routers using a link state routing protocol" IETF Draft. There are many similarities between the concepts and techniques in this document.

Thanks for Abhay Roy and Manav Bhatia for comments regarding duplicate router-id processing.

Thanks for Alvaro Retana and Michael Barnes for comments regarding OSPFv3 Instance ID auto-configuration.

Thanks to Faraz Shamim for review and comments.

Thanks to Mark Smith for the requirement to reduce the adjacency formation delay in the back-to-back ethernet topologies that are prevalent in home networks.

The RFC text was produced using Marshall Rose's xml2rfc tool.

2. OSPFv3 Default Configuration

For complete auto-configuration, OSPFv3 will need to choose suitable configuration defaults. These include:

1. Area 0 Only - All auto-configured OSPFv3 interfaces MUST be in area 0.
2. OSPFv3 SHOULD be auto-configured on for IPv6 on all interfaces intended as general IPv6-capable routers. Optionally, an interface MAY be excluded if it is clear that running OSPFv3 on the interface is not required. For example, if manual configuration or an other condition indicates that an interface is connected to an Internet Service Provider (ISP) and there is no Border Gateway Protocol (BGP) [BGP] peering, there is typically no need to employ OSPFv3. However, note that in many environments it can be useful to test whether an OSPFv3 adjacency can be established. In home networking environments, an interface where no OSPFv3 neighbors are found but a DHCP prefix can be acquired may be considered as an ISP interface.
3. OSPFv3 interfaces will be auto-configured to an interface type corresponding to their layer-2 capability. For example, Ethernet interfaces will be auto-configured as broadcast networks and Point-to-Point Protocol (PPP) interfaces will be auto-configured as Point-to-Point interfaces. Most extant OSPFv3 implementations do this already.
4. OSPFv3 interfaces MUST use the default HelloInterval, 10 seconds, and RouterDeadInterval, 40 seconds, as suggested in Appendix C of [OSPFV3].
5. All OSPFv3 interfaces SHOULD be auto-configured to use an Interface Instance ID of 0 that corresponds to the base IPv6 unicast address family instance ID as defined in [OSPFV3-AF]. Similarly, if IPv4 unicast addresses are advertised in a separate auto-configured OSPFv3 instance, the base IPv4 unicast address family instance ID value, i.e., 64, SHOULD be auto-configured as the Interface Instance ID for all interfaces corresponding to the OSPFv3 instance [OSPFV3-AF].

2.1. Wait Timer Reduction

In many situations, auto-configured OSPFv3 routers will be deployed in environments where back-to-back ethernet connections are utilized. When this is the case, an OSPFv3 broadcast interface will not come up until the other OSPFv3 router is connected and the routers will wait RouterDeadInterval seconds before forming an adjacency [OSPFV2]. In

order to reduce this delay, an auto-configured OSPFv3 router MAY reduce the wait interval to a value no less than (HelloInterval + 1), i.e., 11 seconds. Reducing the setting will slightly increase the likelihood of the Designated Router (DR) flapping but is preferable to the long adjacency formation delay. Note that this value is not included in OSPFv3 Hello packets and does not impact interoperability.

3. OSPFv3 Router ID Selection

As OSPFv3 Router implementing this specification must select a unique Router-ID. A pseudo-random number SHOULD be used for the OSPFv3 Router-ID. The generation should be seeded with a variable that is likely to be unique in that environment. A good choice of seed would be some portion or hash of the Route-Hardware-Fingerprint as described in Section 5.2.2.

Since there is a possibility of a Router ID collision, duplicate Router ID detection and resolution are required as described in Section 5 and Section 5.3.

4. OSPFv3 Adjacency Formation

Since OSPFv3 uses IPv6 link-local addresses for all protocol messages other than message sent on virtual links (which are not applicable to auto-configuration), OSPFv3 adjacency formation can proceed as soon as a Router-ID has been selected and the IPv6 link-local address has completed Duplicate Address Detection (DAD) as specified in IPv6 Stateless Address Autoconfiguration [SLAAC]. Otherwise, there is no change to the OSPFv3 base specification except with respect to duplicate Router-ID detection and resolution as described in Section 5 and Section 5.3.

5. OSPFv3 Duplicate Router-ID Detection and Resolution

There are two cases of duplicate OSPFv3 Router-ID detection. One where the OSPFv3 router with the duplicate Router-ID is directly connected and one where it is not. In both cases, the resolution is for one of the routers with the duplicate OSPFv3 Router-ID to select a new one.

5.1. Duplicate Router-ID Detection for Neighbors

In this case, a duplicate Router-ID is detected if any valid OSPFv3 packet is received with the same OSPFv3 Router-ID but a different IPv6 link-local source address. Once that occurs, the OSPFv3 router with the numerically smaller IPv6 link-local address will need to select a new Router-ID as described in Section 5.3. Note that the fact that the OSPFv3 router is a neighbor on a non-virtual interface implies that the router is directly connected. An OSPFv3 router implementing this specification should assure that the inadvertent connection of multiple router interfaces to the same physical link in not misconstrued as detection of a different OSPFv3 router with a duplicate Router-ID.

5.2. Duplicate Router-ID Detection for OSPFv3 Routers that are not Neighbors

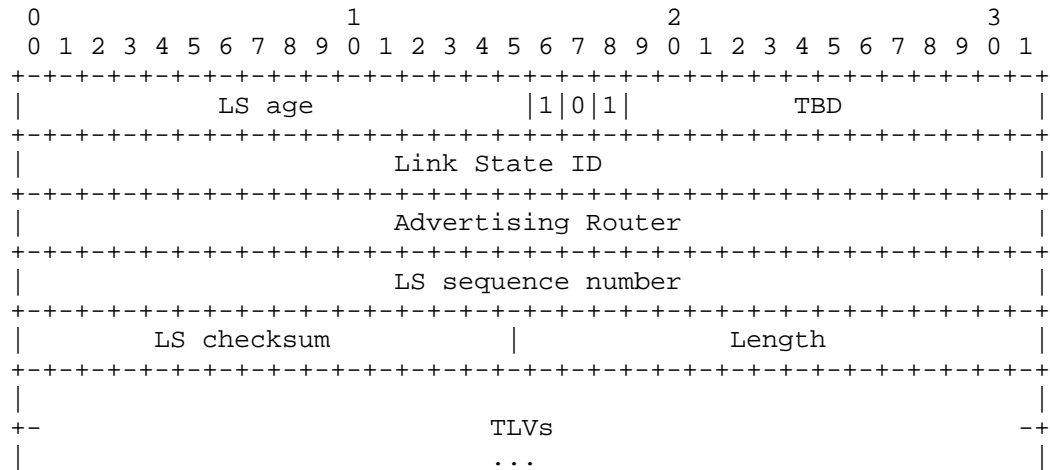
OSPFv3 Routers implementing auto-configuration, as specified herein, MUST originate an Auto-Config (AC) Link State Advertisement (LSA) including the Router-Hardware-Fingerprint Type-Length-Value (TLV). The Router-Hardware-Fingerprint TLV contains a variable length value that has a very high probability of uniquely identifying the advertising OSPFv3 router. An OSPFv3 router implementing this specification MUST compare a received self-originated Auto-Config LSA's Router-Hardware-Fingerprint TLV against its own router hardware fingerprint. If the fingerprints are not equal, there is a Router-ID conflict and the OSPFv3 Router with the numerically smaller router hardware fingerprint MUST select a new Router-ID as described in Section 5.3.

This new LSA is designated for information related to OSPFv3 Auto-configuration and, in the future, could be used other auto-configuration information, e.g., global IPv6 prefixes. However, this is beyond the scope of this document.

5.2.1. OSPFv3 Router Auto-Configuration LSA

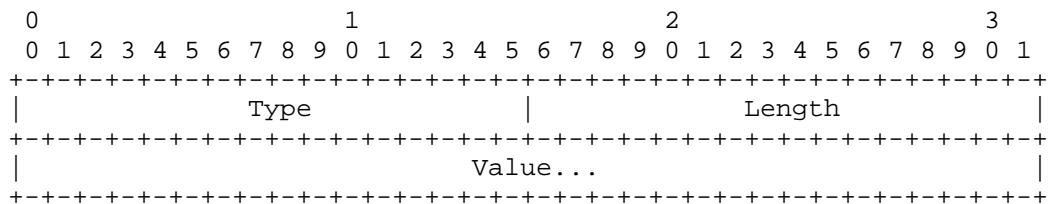
The OSPFv3 Auto-Configuration (AC) LSA has a function code of TBD and the S2/S1 bits set to 01 indicating Area Flooding Scope. The U bit will be set indicating that the OSPFv3 AC LSA should be flooded even

if it is not understood. The Link State ID (LSID) value will be a integer index used to discriminate between multiple AC LSAs originated by the same OSPF Router. This specification only describes the contents of an AC LSA with a Link State ID (LSID) of 0.



OSPFv3 Auto-Configuration (AC) LSA

The format of the TLVs within the body of an AC LSA is the same as the format used by the Traffic Engineering Extensions to OSPF [TE]. The LSA payload consists of one or more nested Type/Length/Value (TLV) triplets. The format of each TLV is:



TLV Format

The Length field defines the length of the value portion in octets (thus a TLV with no value portion would have a length of 0). The TLV is padded to 4-octet alignment; padding is not included in the length

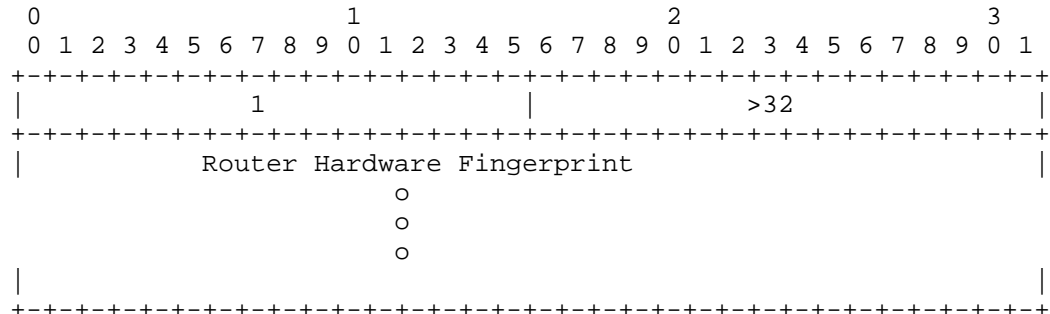
field (so a 3-octet value would have a length of 3, but the total size of the TLV would be 8 octets). Nested TLVs are also 32-bit aligned. For example, a 1-byte value would have the length field set to 1, and 3 octets of padding would be added to the end of the value portion of the TLV. Unrecognized types are ignored.

The new LSA is designated for information related to OSPFv3 Auto-configuration and, in the future, can be used other auto-configuration information, e.g., global IPv6 prefixes.

5.2.2. Router-Hardware-Fingerprint TLV

The Router-Hardware-Fingerprint TLV is the first TLV defined for the OSPFv3 Auto-Configuration (AC) LSA. It will have type 1 and MUST be advertised in the LSID OSPFv3 AC LSA with an LSID of 0. It SHOULD occur, at most, once and the first instance of the TLV will take precedence over preceding TLV instances. The length of the Router-Hardware-Fingerprint is variable but must be 32 bytes or greater.

The contents of the hardware fingerprint should be some combination of MAC addresses, CPU ID, or serial number(s) that provides an extremely high probability of uniqueness. It MUST be based on hardware attributes that will not change across hard and soft restarts.



Router-Hardware-Fingerprint TLV Format

5.3. Duplicate Router-ID Resolution

The OSPFv3 Router selected to resolve the duplicate OSPFv3 Router-ID condition must select a new OSPFv3 Router-ID. After selecting a new Router-ID, the Router-LSA with the prior duplicate Router-ID MUST be purged. all self-originated LSAs MUST be reoriginated, and any OSPFv3

neighbor adjacencies MUST be reestablished.

5.4. Change to Received Self-Originated LSA Processing

RFC 2328 [OSPFV2], Section 13.4, describes the processing of received self-originated LSAs. If the received LSA doesn't exist, the receiving router will purge it from the OSPF routing domain. If the LSA is newer than the version in the Link State Database (LSDB), the receiving router will originate a newer version by advancing the LSA sequence number and reflooding. Since it is possible for an auto-configured OSPFv3 router to choose a duplicate OSPFv3 Router-ID, OSPFv3 routers implementing this specification should detect when multiple instances of the same self-originated LSA are purged or reoriginated since this is indicative of an OSPFv3 router with a duplicate Router-ID in the OSPFv3 routing domain. When this condition is detected, the OSPFv3 Router SHOULD delay self-originated LSA processing for LSAs that have recently been purged or reflooded. This specification recommends 10 seconds as the interval defining recent self-originated LSA processing and an exponential back off of 1 to 8 seconds for the processing delay.

6. Security Considerations

A unique OSPFv3 Interface Instance ID is used for auto-configuration to prevent inadvertent OSPFv3 adjacency formation, see Section 2

The goals of security and complete OSPFv3 auto-configuration are somewhat contradictory. When no explicit security configuration takes place, auto-configuration implies that additional devices placed in the network are automatically adopted as a part of the network. However, auto-configuration can also be combined with password configuration (see below) or future extensions for automatic pairing between devices. These mechanisms can help provide an automatically configured, securely routed network.

It is RECOMMENDED that OSPFv3 routers supporting this specification also offer an option to explicitly configure a password for HMAC- SHA authentication as described in [OSPFV3-AUTH-TRAILER]. When configured, the password will be used on all auto-configured interfaces with the Security Association Identifier (SA ID) set to 1 and HMAC-SHA-256 will be used as the authentication algorithm.

7. Management Considerations

It is RECOMMENDED that OSPFv3 routers supporting this specification also allow explicit configuration of OSPFv3 parameters as specified in Appendix C of [OSPFV3]. This is in addition to the authentication key configuration recommended in Section 6. However, it is acknowledged that there may be some deployment scenarios where manual configuration is not required.

8. IANA Considerations

This specification allocates a new OSPFv3 LSA, OSPFv3 Auto-Configuration (AC) LSA, TBD, as described in Section 5.2.1.

This specification also creates a registry for OSPFv3 Auto-Configuration (AC) LSA TLVs. This registry should be placed in the existing OSPFv3 IANA registry, and new values can be allocated via IETF Consensus or IESG Approval.

Three initial values are allocated:

- o 0 is marked as reserved.
- o 1 is Router-Hardware-Fingerprint TLV (Section 5.2.2).
- o 65535 is an Auto-configuration-Experiment-TLV, a common value that can be used for experimental purposes.

9. References

9.1. Normative References

- [OSPFV2] Moy, J., "OSPF Version 2", RFC 2328, April 1998.
- [OSPFV3] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
- [OSPFV3-AF] Lindem, A., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, April 2010.
- [OSPFV3-AUTH-TRAILER] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 6506, February 2012.
- [RFC-KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [SLAAC] Thomson, S., Narten, T., and J. Tatuya, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [TE] Katz, D., Yeung, D., and K. Kompella, "Traffic Engineering Extensions to OSPF", RFC 3630, September 2003.

9.2. Informative References

- [BGP] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

Authors' Addresses

Acee Lindem
Ericsson
102 Carric Bend Court
Cary, NC 27519
USA

Email: acee.lindem@ericsson.com

Jari Arkko
Ericsson
Jorvas, 02420
Finland

Email: jari.arkko@piuha.net

Homenet working Group
Internet-Draft
Intended status: Informational
Expires: August 22, 2013

O. Troan
Cisco Systems
L. Colitti
Google
February 18, 2013

IPv6 Multihoming with Source Address Dependent Routing (SADR)
draft-troan-homenet-sadr-00

Abstract

A multihomed network using provider aggregatable addresses must send the packet out the right path to avoid violating the provider's ingress filtering. This memo suggests a mechanism called Source Address Dependent Routing to solve that problem.

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, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
---------------------------	---

2. Conventions	2
3. Terminology	2
4. Using SADR for multihoming	3
5. A Conceptual Forwarding Algorithm	3
6. Routing considerations	4
6.1. Routing Protocol extensions	5
6.2. Simplified SADR in home networks	5
7. Interaction between routers and hosts	5
8. IANA Considerations	6
9. Security Considerations	6
10. Acknowledgements	6
11. References	6
11.1. Normative References	6
11.2. Informative References	7
Authors' Addresses	7

1. Introduction

IPv6 is designed to support multiple addresses on an interface, and the intention was to use this feature to support multihoming with provider aggregatable addresses.

One difficulty of multihoming with provider-aggregatable space is that providers typically employ BCP38 [RFC2827] filtering. If a network sends traffic to its upstream provider using a source address that was not assigned by that provider, the traffic will be dropped. Thus, if a network is multihomed to multiple providers, it must ensure that traffic is sent out the correct exit for the packet's source address.

As long as upstream traffic is sent to the correct provider, hosts inside the network are free to use source addresses assigned by any of the network's upstream providers. In such a scenario, each host has multiple addresses, one or more from each provider the network is connected to. The network ensures that packets are sent to the correct upstream by forwarding packets based on the destination address and the source address. This we call source address dependent routing (SADR). This memo shows how SADR can be used to implement multihoming.

2. Conventions

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

3. Terminology

Service Provider	An entity that provides the network with external connectivity, e.g. to the Internet.
------------------	---

WAN Interface	An interface connected to a Service Providers. WAN interfaces may either be physical links or virtual interfaces such as tunnels. WAN interfaces are used to send ingress traffic from the Internet to the End-User, and egress traffic from the End-User network to the Internet. Ingress traffic may be received on any active interface at any time. Egress traffic follows a set of rules within the router in order to choose the proper WAN interface.
Border Router	A border router has one or more external interfaces connecting it to one or more Service Providers. The border router receives one or more delegated prefixes, each associated with one or more WAN interfaces.
External Route	A route that is learned from a Service Provider. Each External Route has an Acceptable Source Prefix which determines which source addresses may use that route.
Internal Router	A router that is not a Border Router.
Internal Route	A route to a destination inside the network.

4. Using SADR for multihoming

SADR is similar to policy based routing. This memo proposes a simple extension to the destination based longest match algorithm to constrain it to source address.

In order to support ingress filtering by upstream networks, the network MUST treat external routes specially. Ingress filtering MAY also be used internally, by installing (S,D) routes for locally assigned prefixes, where the source prefix would be the aggregatable prefix. If no ingress filtering is performed inside the network, then normal non-source constrained forwarding is used.

5. A Conceptual Forwarding Algorithm

This section describes a conceptual forwarding algorithm. An implementation might implement this differently, e.g. with multiple tables, as long as the external behaviour is as described.

First a longest match lookup is done in the routing table for the destination address, then for the resulting set a longest matching lookup is done for the source address.

In a destination based routing table, an entry in the routing table can be shown as "D -> NH". That is, to get to a destination D, use next-hop NH. For a source constrained routing table we propose the following notation. (Source Network, Destination network) -> Next-hop. (S, D) -> NH. A route that is not source constrained can be represented as (*, D) -> NH.

For convenience this document shows the routing table as a single destination based routing table, with source address constrained paths. This does not preclude other implementations, as long as the external behaviour is the same.

A router forwarding a packet does a longest match look-up on the destination address. If this is a (*, D) entry, it forwards the packet out the best next-hop as before (doing equal cost multi path load balancing etc). If the look-up results in a (S, D) entry, the look-up function does a longest match on the source address among the set of (S, D) paths. If there is a match the packet is forwarded out the given next-hop, if not an ICMP destination address unreachable message, code 5 is returned [RFC4443]. A routing entry may have both (S, D) paths and (*, D) paths. The longest match wins.

The following example show the routing table of a network connected to two ISPs, ISP A and ISP B. Both ISPs offer default connectivity and ISP B also offers a more specific route to a walled garden service.

```
(2001:db8::/56, ::/0) -> ISP_A      # Default route to ISP A
(2001:db9::/56, ::/0) -> ISP_B      # Default route to ISP B
(*, 2001:db8::/64) -> R1             # Internal network, prefix from A
(*, 2001:db8:1::/64) -> R2           # Internal network, prefix from A
(*, 2001:db9::/64) -> R1             # Internal network, prefix from B
(*, 2001:db9:1::/64) -> R2           # Internal network, prefix from B
(*, fd00::/64) -> R3                 # Internal network ULA
(2001:db9::/56, 2001:420::/32) -> ISP_B # Walled garden route from ISP B
```

Figure 1: Example Routing Table

A packet with the SA, DA of 2001:db8::1, 2001:dead:beef::1 would be forwarded to ISP A, likewise a packet with SA, DA 2001:db9::1, 2001:dead:beef::1 would be forward to ISP B. An packet with SA,DA 2001:db8::1, 2001:db9::1 would be forwarded using normal destination based routing. A packet to the walled garden SA,DA 2001:db9::1, 2001:420::1 would be sent to ISP B. A packet with SA,DA 2001:db8::1,2001:420::1 would be dropped with an ICMP unreachable message being sent back.

6. Routing considerations

Now that we have described the function of the source constrained routing table. How does the table get populated?

6.1. Routing Protocol extensions

The generic answer is that the routing protocol used in the network has to be extended to support (S, D) routes. Specifically, the routing protocol should distribute, for each External Route, the Acceptable Source Prefix(es) for that route. This may be done, for example, using [I-D.baker-ipv6-ospf-dst-src-routing] or [I-D.baker-ipv6-isis-dst-src-routing]. In the case of OSPFv3, for example, external routes are advertised in an AS-External-prefix LSA, [RFC5340]

6.2. Simplified SADR in home networks

In a home network using a dynamic prefix assignment mechanism such as [I-D.arkko-homenet-prefix-assignment] it may be known that a particular Border Router is announcing both an External Route and a Usable Prefix (for example, if the same router ID is announcing both). In this case, interior routers may assume that the Acceptable Source Prefix of the External Route announced by that Border Router is in fact the Usable Prefix announced by that Border Router.

An internal router when receiving a AS-External LSA route will install that in the routing table as normal. When the internal router receives a usable prefix as part of prefix assignment, the router shall add source constrained entries to all the AS-External routes received from the same border router (matching router-ID).

Routes that are not associated with a border router or are not AS-External do not have source constrained paths.

The routing protocol requirements for simplified SADR in the home network are:

1. Routing protocol must flood all information to all routers in the home network. (Single area).
2. Prefix assignment and unicast routing must be done in the same protocol.
3. A router must be uniquely identified (router-id) so that router advertisements and prefix assignment can be tied together

7. Interaction between routers and hosts

Generally, hosts need not be aware that SADR is in use in the network. Hosts simply choose source addresses and the network will deliver the traffic to the appropriate upstream. One exception is when an Acceptable Source Prefix becomes invalid (e.g., if the Border Router which announced it crashes, or its WAN link goes down). In this case, current hosts will continue to use source addresses in that Acceptable Source Prefix without knowing that all communication outside the network is likely to fail. In this case, interior routers can improve responsiveness by deprecating the addresses in that Acceptable Source Prefix.

ICMP [RFC4443] includes a Destination unreachable code 5 - "Source address failed ingress/egress policy". Hosts MUST adhere to this message, and based on the unreachable message try another source address.

8. IANA Considerations

This specification does not require any IANA actions.

9. Security Considerations

10. Acknowledgements

The authors would like to thank Jari Arkko and Andrew Yourtchenko for their ideas and review.

11. References

11.1. Normative References

- [I-D.arkko-homenet-prefix-assignment]
Arkko, J., Lindem, A., and B. Paterson, "Prefix Assignment in a Home Network", draft-arkko-homenet-prefix-assignment-03 (work in progress), October 2012.
- [I-D.ietf-ospf-ospfv3-autoconfig]
Lindem, A. and J. Arkko, "OSPFv3 Auto-Configuration", draft-ietf-ospf-ospfv3-autoconfig-00 (work in progress), October 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

11.2. Informative References

- [I-D.baker-ipv6-isis-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-00 (work in progress), February 2013.
- [I-D.baker-ipv6-ospf-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using OSPFv3", draft-baker-ipv6-ospf-dst-src-routing-00 (work in progress), February 2013.
- [I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "Home Networking Architecture for IPv6", draft-ietf-homenet-arch-06 (work in progress), October 2012.

Authors' Addresses

Ole Troan
Cisco Systems
Philip Pedersens vei 1
Lysaker 1366
Norway

Email: ot@cisco.com

Lorenzo Colitti
Google
Roppongi Hills Mori Tower PO box 22
Minato, Tokyo 106-6126
Japan

Email: lorenzo@google.com