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

Table of Contents



## 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

- A is a prefix,
- B is a more-specific prefix within A,
- C is a different prefix, and
- 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-

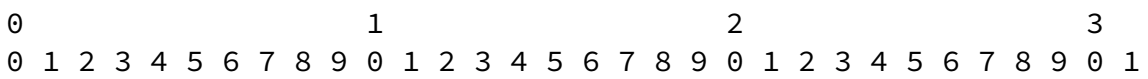
external-LSAs, and the intra-area-prefix LSA. This can be done specifically with the information we have thought about, or designed for extensibility. We choose extensibility.

## 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.

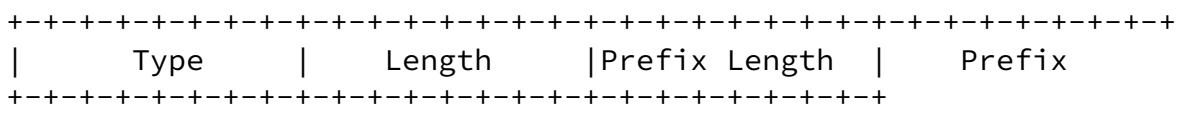
### 3.1.1. IPv6 Destination Prefix TLV

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.








Destination Prefix TLV

Destination Prefix Type: assigned by IANA

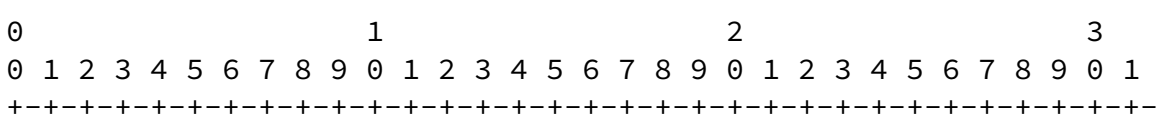
TLV Length: Length of the TLV in octets

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.



```
|      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

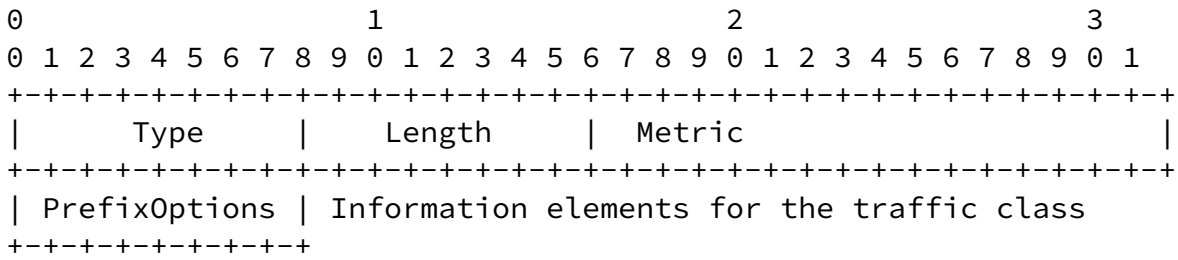
Flow Label Type: assigned by IANA

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




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.

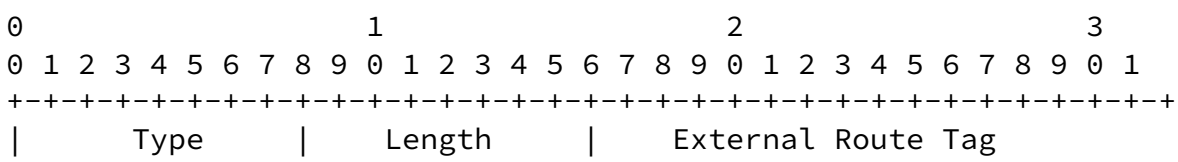
Information Elements This information element will be followed by zero or more information elements that describe the traffic class. the traffic class will have been fully described when parsing reaches the end of the LSA or finds a new Metric TLV.

### 3.1.5. External Route Tag TLV





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



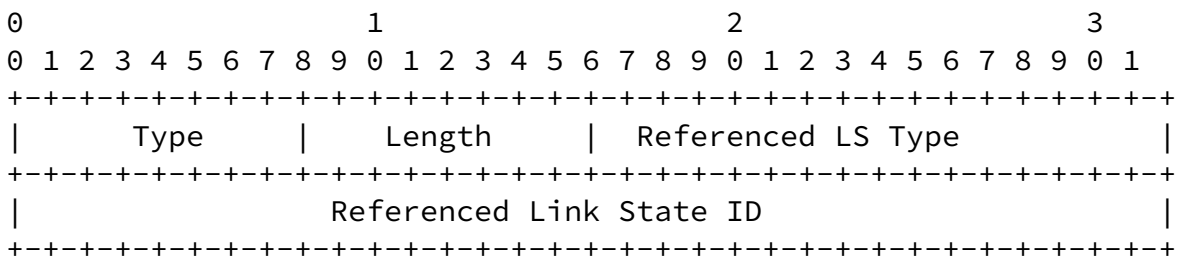
External Route Tag TLV

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




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.

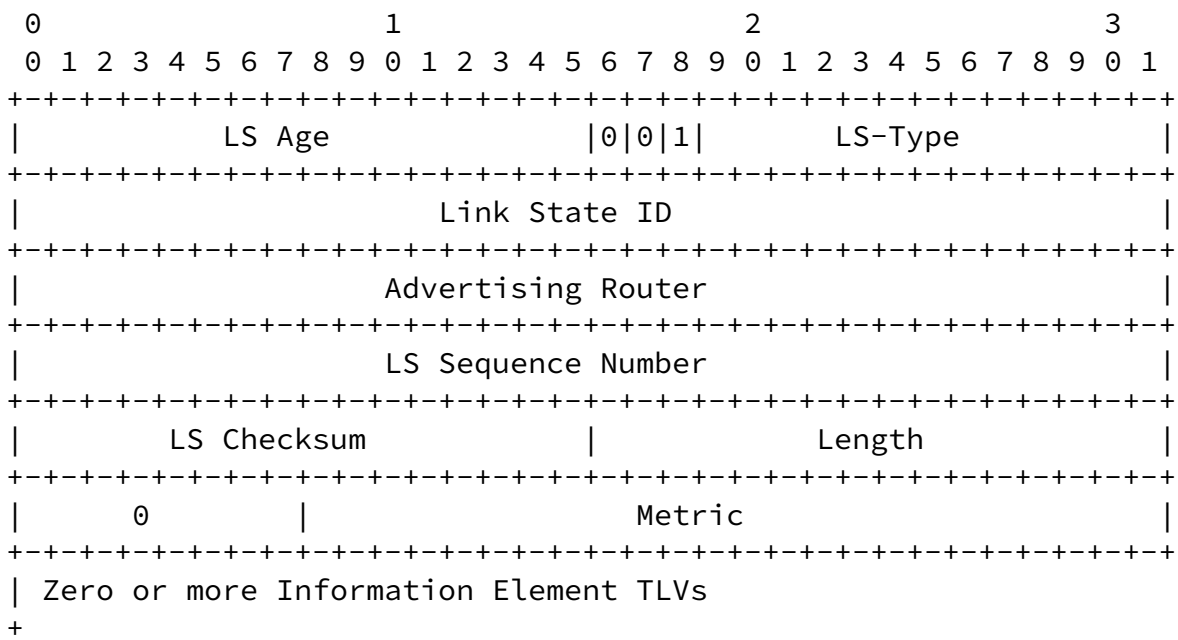



Figure 1: Extensible-Inter-area-prefix-LSA

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.

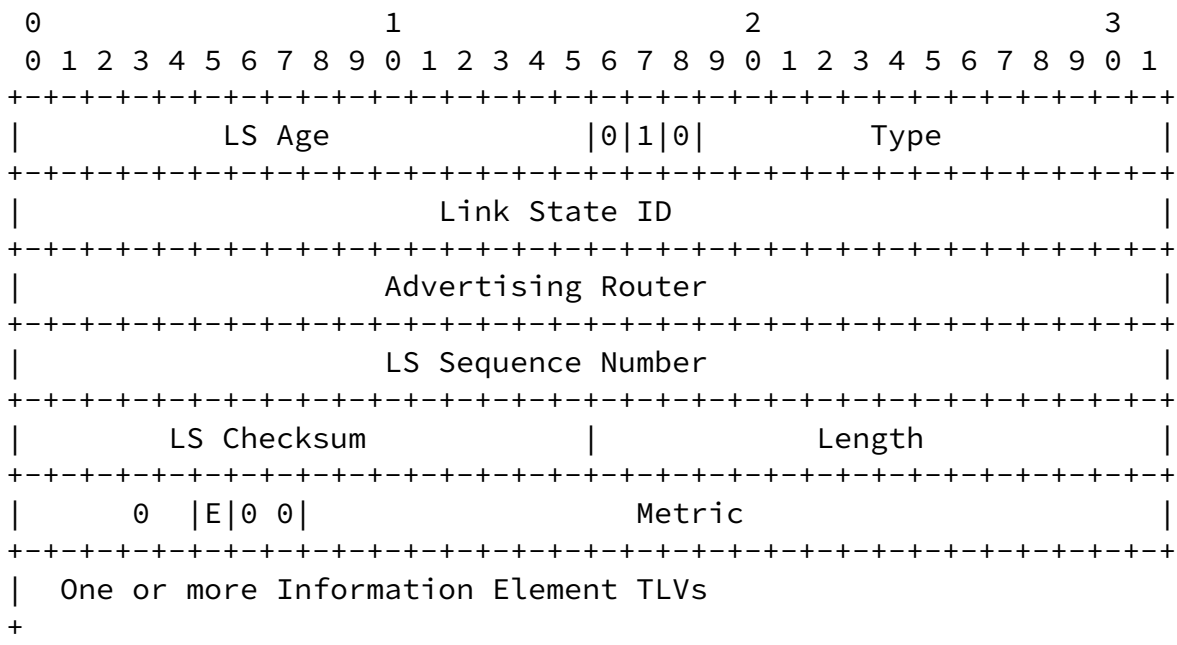



Figure 2: Extensible-AS-external-LSA

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.

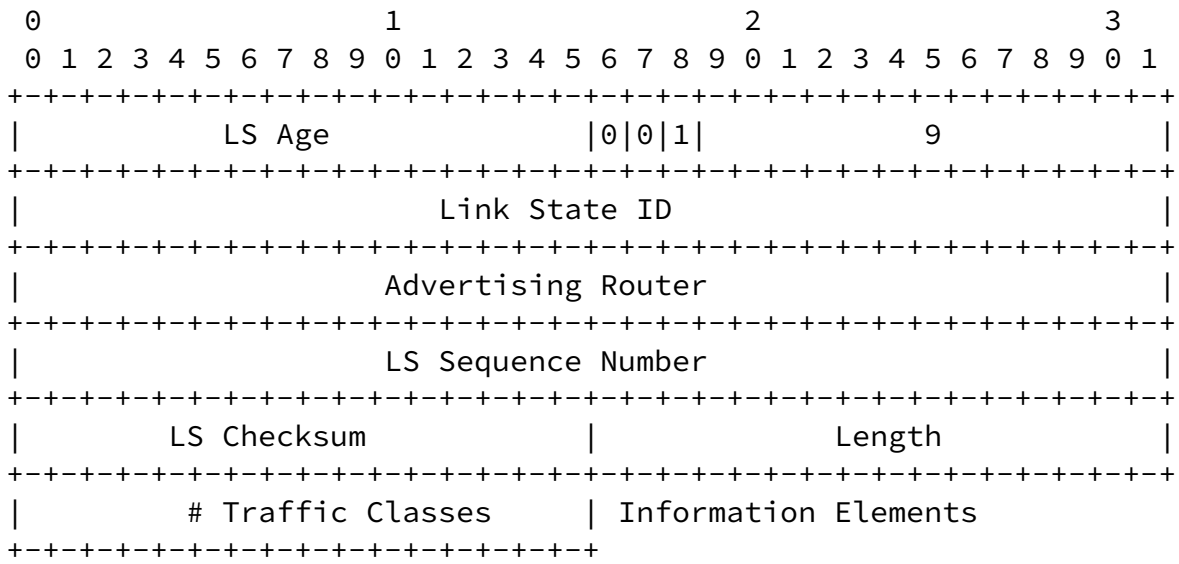



Figure 3: Extensible-Intra-Area-Prefix-LSA

# 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 discontiguous lookup key.

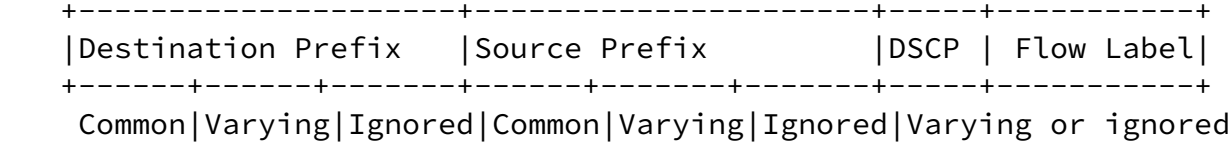


Figure 4: Treating a traffic class as a virtual bit string

### 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
```





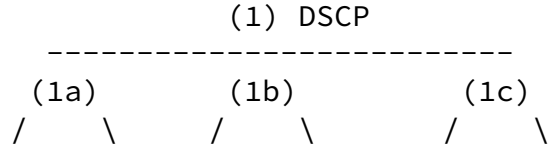



Figure 5: Example PATRICIA Tree

### 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