Internet Engineering Task Force
Internet Draft
Expires May 1995

Dimitry Haskin
Ross Callon
Bay Networks, Inc.
November 1994

Routing Aspects Of IPv6 Transition

(draft-haskin-ipv6-routing-aspects-00.ps)

Status of this memo

Abstract

This paper discusses routing aspects associated with the transition from IPv4 to IPv6. The approach outlined here is designed to be compatible with the Simple Internet Transition (SIT) mechanism.

The proposals contained in this document are the opinions of the authors, and have not yet been discussed in detail by the working group. This document is intended as input to the IPNG, Tacit, and Ngtrans working groups.

1 Terminology

This paper uses the following terminology:

Node
A protocol module that implements IPv6.

Router
A *node* that forwards packets not explicitly addressed to itself.

Host
Any *node* that is not a router.

Link
A communication facility or medium over which *nodes* can communicate at the link layer, i.e., the layer immediately below network layer.

Interface
A node's attachment to a *link*.

Address
An network layer identifier for an *interface* or a group of interfaces.

Neighbors
Nodes attached to the same *link*.

Routing Domain
A collection of routers which coordinate the routing knowledge using a single routing protocol.

*Routing Region (*or just *Region)*
A collection of routers interconnected by a single internet protocol (e.g. IPv6) and coordinating their routing knowledge using routing protocols from a single internet protocol stack. A routing region can be a subset or a superset of a *routing domain.*

Border Router
A *router* that forwards packets across *routing region* boundaries.

Tunneling
Encapsulation of protocol A within protocol B, such that A treats B as though it were a datalink layer.

Reachability Information
: Information describing the set of reachable destinations that can be used for packet forwarding decisions.

Routing Information
: Same as *reachability information.*

Address Prefix
: The high-order bits in an *address.*

Routing Prefix
: *Address prefix* that expresses destinations which have addresses with the matching address prefixes. It is used by *routers* to locate a *link* for delivery a datagram.

Route Leaking
: Advertisement of network layer *reachability information* across *routing region* boundaries.

Translating Router
: A dual (IPv4/IPv6) protocol *router* that is capable translate IPv4 packet headers into IPv6 headers and vise versa.

2 Issues And Outline

This internet draft gives an initial overview of the routing aspects of IPv4 to IPv6 transition. The approach outlined here is designed to be compatible with the Simple IPv6 Transition (SIT) [4][5].

During an extended IPv4-to-IPv6 migration period, IPv6-based systems must coexist with the installed base of IPv4 systems. In such a dual internetworking protocol environment, both IPv4 and IPv6 routing infrastructure will be present. Initially, deployed IPv6-capable domains might not be globally interconnected via an IPv6-capable internet infrastructure and therefore may need to communicate across IPv4-only routing regions.

In order to achieve dynamic routing in such a mixed environment, there need to be mechanisms to globally distribute IPv6 network layer reachability information between dispersed IPv6 routing regions. The same techniques can be used in later stages of IPv4-to-IPv6 transition to route IPv4 packets between isolated IPv4-only routing region over an IPv6 infrastructure.

Similarly, at some stages of transition, translation between IPv4 and IPv6 packet formats may be necessary in order to allow IPv6-only systems to talk with IPv4-only systems.

The SIT transition provides a dual-stack transition, augmented by use of encapsulation and translation where necessary and appropriate. Routing issues related to this transition include:

(1) Routing for IPv4 packets

(2) Routing for IPv6 packet

(2a) IPv6 packets with IPv4-incompatible addresses

(2b) IPv6 packets with IPv4-compatible addresses

(3) Operation of manually configured static tunnels

(4) Operation of automatic encapsulation and translation

(4a) Locating encapsulators and translators

(4b) Ensuring that routing is consist with encapsulation and translation

Basic mechanisms required to accomplish these goals include: (i) Dual Stack Route Computation; (ii) Manual configuration of tunnels; and (iii) Route Leaking to support translation and automatic encapsulation.

The basic mechanism for routing of IPv4 and IPv6 involves dual-stack routing. This implies that routes are separately calculated for IPv4 addresses and for IPv6 addresses. This may be done either by running a separate routing protocol for each, or for running one “integrated” routing protocol which calculates routes for both IPv4 and IPv6. For example, one instance of OSPF might be used, with a single topology (a single area structure, a single router ID for each router, a single DR per LAN, etc.) to calculate routes for both IPv4 and IPv6.

Tunnels (either IPv4 over IPv6, or IPv6 over IPv4) may be manually configured. For example, in the early stages of transition this may be used to allow two IPv6 regions to interact over an IPv4 infrastructure. Manually configured static tunnels are treated as if they were a normal data link. This is discussed in more detail in section 3.1.

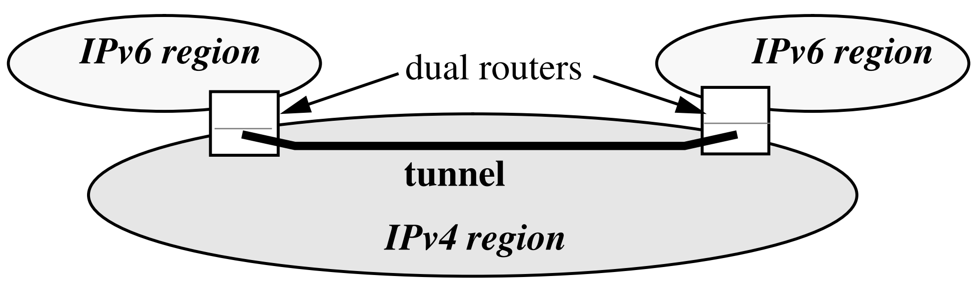
Use of automatic encapsulation and translation requires consistency of routes between IPv4 routes and IPv6 routes for destinations using IPv4-compatible addresses. For example, consider a packet which starts off as an IPv6 packet, but then is translated into an IPv4 packet in the middle of its path from source to destination. This packet must locate a translator at the correct part of its path. Also, this packet has to follow a consistent route for the entire path from source to destination. This is discussed in more detail in sections 3.2 and 4.

3 Route Dissemination Techniques

3.1 Manually Configured Static Tunnels

Tunneling techniques are already widely deployed for bridging non-IP network layer protocols (e.g. AppleTalk, CLNP, IPX) over IPv4 routed infrastructure. IP tunneling is an encapsulation of arbitrary packets inside IP datagrams that are forwarded over IP infrastructure between tunnel endpoints. For a tunneled protocol, a tunnel appears as a single-hop link (i.e. routers that establish a tunnel over a network layer infrastructure can inter-operate over the tunnel as if it were a one-hop, point-to-point link). Once a tunnel is established, routers at the tunnel endpoints can establish routing adjacencies and exchange routing information. Describing the protocols for performing encapsulation is outside the scope of this paper.

In order for a IPv6 router to be able to encapsulate IPv6 packets into IPv4 datagrams and to forward encapsulated packet over a IPv4 tunnel, such a router must also be able to speak IPv4 (i.e. be a dual protocol router). The route tunnelling techniques call for dual protocol routers to be deployed at demarcation points between adjacent IPv6 and IPv4-only routing regions:



Forwarding of IPv6 packets between IPv6 routing regions is straightforward -- when a packet reaches a border router, the border router examines it’s routing database to find an interface to the next-hop router. If the forwarding interface is connected to a tunneled link, the packet must be encapsulated into an IPv4 datagram according to the adopted encapsulation scheme, and the resulting datagram is forwarded over the tunnel. Intermediary IPv4 routers between the tunnel endpoints forward the datagram as they would any other IPv4 datagram, using information in the encapsulating header. The recipient border router, in an adjacent IPv6 region, strips off the encapsulating header and forwards the original IPv6 packets toward its ultimate destination. Note that the destination and source addresses in the encapsulating header are the IPv4 addresses of the tunnel endpoints, not derivatives of the destination and source addresses of the encapsulated IPv6 packet.

In the route tunneling scheme described here, there is a complete separation of the IPv6 network reachability information from the IPv4 routing information; i.e. this is a dual stack approach. Nevertheless, an IPv4 routing region and an IPv6 routing region can overlap. Each such region can share routers which support both IPv4 and IPv6 routing.

The downside of the dual stack approach is that it does not support a dynamic end-to-end routing of data sent between an IPv4-only node and an IPv6-only node with an IPv4-compatible address. [It is already accepted that IPv4-only nodes will not be able to interoperate with IPv6-only nodes with IPv4-incompatible addresses.] When sending packets between IPv4-only and IPv6-only nodes, due to the routing information separation, the route to the destination node is not available to the routing protocol at the source node. Therefore, packets first have to be sent to a default translating dual protocol router which then can dynamically route converted packets to their ultimate destinations.

There are the following advantages to employ encapsulation for exchanging routing information:

- The underline infrastructure which furnishes a tunnel link is transparent to protocols that are bridged with the tunnel (i.e. there is no changes to bridged protocols).

- All types of IPv6 route prefixes without exception can be advertised with routing protocols. Therefore, no restriction need to be imposed on formats of the addresses in IPv6 packets that can be routed with this scheme.

- If a connectivity between IPv6 nodes is all that needed, only border routers at boundaries with IPv4-only routing regions need to be dual protocol routers.

- Since IPv6 packets are encapsulated only when they travel over network segments that don't support IPv6, and are forwarded according to their native headers anywhere else, all kind of policy routing can be employed over the entire IPv6 portion of data path.

- Routers from major vendors already support the multiprotocol operation that is needed at tunnel endpoints.

The disadvantages of tunneling are that:

- They need to be manually configured.
- They may circumvent security firewalls of the encapsulating infrastructure -- only addresses of the tunnel endpoints are normally visible to such firewalls, not actual attributes of encapsulated packets. Note that the same stands true for "automatic" tunneling which is used in the route leaking scheme.
- Since a tunnel may appear as a one-hop link, some routing protocols may prefer a tunnel over a real multi-hop link. Therefore, IPv6 packets may be routed across an IPv4 routing region even when an alternative homogeneous IPv6 path is available. Note that this disadvantage may be eliminated by using a routing protocol such as OSPF which allows a considerable dynamic range of metric values to be assigned to links.

Though this section describes tunneling of IPv6 routing information and IPv6 packets over an IPv4 infrastructure, when a need arises for IPv4 routing regions to communicate via an IPv6 routing infrastructure, the similar tunneling technique can be used -- except IPv4 packets will be encapsulated within IPv6 datagrams.

3.2 Route Leaking

"Route leaking" is used to describe advertisement of network layer reachability information across routing domain boundaries. Normally, routing domains exchanging network layer reachability information run the same network layer protocol (i.e. use the same addressing scheme). However, we are particularly interested here in route leaking between IPv4-only and IPv6-only regions, between dual regions and IPv4-only regions, and between dual regions and IPv6-only regions.

Since the IPv6 addressing architecture is quite different from the IPv4 addressing architecture, it isn't plausible that route leaking can be used to leak all types of IPv6 address into an IPv4 routing region. Nevertheless, there is an important class of IPv6 addresses -- IPv6 addresses with embedded IPv4 addresses [3] -- that can be successfully injected into an IPv4 region. IPv6 addresses with embedded IPv4 addresses are designed to facilitate the transition of the Internet from IPv4 to IPv6. There are two forms of such addresses: an IPv4-compatible address that can be assigned to an IPv6 node and an IPv4-mapped address that is the IPv6 representation of an IPv4 address. Both address forms embed an IPv4 address in the low-order 32 bits.

When a dual protocol border router leaks an IPv6 routing prefix with an embedded IPv4 routing prefix into an adjacent IPv4 routing region, it strips off the high-order 96 bits and advertises the IPv4 routing prefix with a IPv4 routing protocol. The recipient border router in an adjacent IPv4 region treats IPv4 prefixes that are received from neighboring IPv6 regions as regular IPv4 routing prefixes and is free to disseminate them through its own routing region as well as propagate them to other adjacent IPv6 routing regions.

When an IPv4 routing prefix reaches a dual protocol border router in an IPv6 region, there is no efficient way to determine whether this prefix represents IPv6 or IPv4 destinations. Therefore, such a prefix needs to be installed in the IPv4 routing information base (RIB), and the IPv4-compatible IPv6 equivalent is to be installed into the IPv6 RIB. [It would appear that, for completeness, the matching IPv4-mapped address would need to be installed along with the IPv4-compatible IPv6 prefix into the IPv6 RIB, but, if the router is made capable of forwarding IPv6 packets with IP4-mapped IPv6 destination addresses as if they were IPv4-compatible IPv6 addresses (e.g. by substituting the address prefix 0:0:0:0:0:0 with the 0:0:0:0:0:FFFF prefix during forwarding table lookup), such a duplication doesn't seem to be necessary.]

It is important to observe that in order for this scheme to work, each participating IPv6 node must be a dual protocol (IPv4/IPv6) node with an IPv4-compatible IPv6 address as well as the matching IPv4 address assigned to it. This is the clear downside of the route leaking method since it limits types of IPv6 addresses that can be disseminated with this scheme.

When forwarding IPv6 packets, when a dual protocol router determines that the next hop for an IPv6 packet is an IPv4 router, the router sends the packet using “automatic” IPv6-over-IPv4 tunneling (i.e. encapsulates the packet with an IPv4 header with the destination address taken from the IPv4 address embedded in the destination address of the original IPv6 packet) if the packet’s destination address is an IPv6-compatible address of an IPv6 node, or the router needs to perform IPv6-to-IPv4 packet translation if the packet’s destination address is an IPv6-mapped address of an IPv4-only node. When the encapsulated packet enters an IPv6 region, it has to be decapsulated by a decapsulating router before it can proceed to the destination node. If the encapsulated packet manages to reach the destination IPv4/IPv6 host without being decapsulated by an intermediary router, the destination host must be able decapsulate the packet. Note that, in contrast to the decapsulating function at a tunnel endpoint, the decapsulating router must be able of decapsulation of packets that are not explicitly addressed to itself.

Another disadvantage of the scheme is that it may suffer from the pitfalls which can be encountered when reachability information from one routing protocol is injected into a different routing protocol -- one needs to be careful that a possible loss of information during the route leaking as well as a lack of the inter-protocol coordination, wouldn’t result in the formation of routing loops.

4 Translation and Automatic Encapsulation

The use of automatic tunneling and of translation introduces two issues: (i) How to ensure that packets which need to be translated (or encapsulated using automatic capabilities) arrive at a translator; (ii) How to ensure that the routing used before translation/encapsulation is compatible with the route used after translation/encapsulation.

In principle there are a wide variety of complex ways to locate translators and encapsulators. Where translators are located needs to be closely tied into routing, in the sense that in some cases the route that a packet takes has to traverse translators at the proper locations within the network. In order to keep things manageable, it is highly desirable to keep this simple. We therefore recommend that one or both of two approaches be used:

- All boundary routers between dissimilar routing regions must be dual, and be capable of translation and encapsulation.
- For stub (non-transit) dual-capable routing regions (supporting both IPv4 and IPv6) which are interconnected with only one backbone, where the backbone is a single-protocol backbone, an alternate solution is acceptable: Here the border routers between the stub and the backbone may support only the one protocol used in the backbone, and translators may be located internally within the region. The translators may attract traffic from inside the stub destined outside of the stub by advertising a default route.

So long as the routers at boundaries between dissimilar routing regions are capable of translating between IPv4 and IPv6 formats, routing of IPv6 packets with IPv4-compatible or IPv4-mapped addresses is essentially equivalent, from the perspective of calculated routes, to routing of IPv4 packets.

There are several options regarding how routes for IPv6 with IPv4-compatible or IPv4-mapped addresses may be calculated. This may be done by simply running routing for IPv4 separate from IPv6 (even in pure IPv6 regions), and using the IPv4 routes for those IPv6 packets which use IPv6 addresses with embedded IPv4 addresses. Alternatively, the IPv4 routes may be fed into IPv6 routing protocols. If the latter approach is used, then the routes may be fed in using only one prefix (0:0:0:0:0:0), or using both IPv4-compatible prefixes (0:0:0:0:0:0 and 0:0:0:0:0:FFFF). Note that using two prefixes is the most straightforward, but implies some duplication of function (since in most cases the same route should be used for a given IPv4 address, regardless of whether it is used for an IPv4 packet, an IPv6 packet with an IPv4-mapped address of the form "0:0:0:0:0:0<IPv4-address>", or with an IPv4-compatible address of the form "0:0:0:0:0:FFFF<IPv4-address>". The one case where this apparent duplication of function is required is when the translators and encapsulators are not located in the boundary routers between dissimilar routing regions. In this case, the path followed by a IPv6 packet with an IPv4-compatible address may be different from the path followed an IPv4 packet with the same address.

5 More Hard Problems

5.1 IPv4 --> IPv6, Translate or Encapsulate?

Consider an IPv4 packet, which is leaving an IPv4-only routing region and has arrived at a border router which needs to forward the packet into an IPv6-only routing region. Furthermore, lets assume that the packet is entering a backbone service provider, and the final destination for the packet is “relatively remote”, such as in another domain which served via a different service provider.

In this case, the packet needs to be either translated or encapsulated in order to transit the IPv6 region. However, note that the boundary router entering the first IPv6 public service provider can not reasonably be expected to know whether the final destination is an IPv4-only or IPv6-only or dual system. Thus, there is no way to know for sure whether the packet will ever again reach any system which is capable of understanding IPv4. This implies that the packet will need to be translated in this case. If the packet later has to exit the IPv6-only region and enter an IPv4-only region, then the packet may need to be translated again. This implies that it is possible that the packet may need to be translated multiple times before it is delivered to its final destination (with implicit performance consequences).

Also, note that when the packet it being translated from IPv4 to IPv6, it is also impractical for the border router to know whether the addresses should be translated using the 0:0:0:0:0:0 or 0:0:0:0:0:FFFF prefix. Therefore, a single prefix will need to be used (presumably 0:0:0:0:0:0), and then the prefixes may need to be fixed by IPv6-only routers or border routers closer to the destination routing region.

5.2 Finer Scale for Translation

Note that translation is required because of the expectation that there will be IPv6-only systems in the near future (i.e., well before IPv4 ceases to be commercially important). However, if companies ship IPv6-only systems, then these are likely to be mixed in the same environment with IPv4-only systems.

This implies that it will not be sufficient to <u>just</u> translate on major region boundaries. Rather, it may be necessary for routers to translate between two devices in the same routing region, or even on the same LAN.

To make this work in practice, it is important to simplify the operation of routing in the dual environment which contains both IPv4-only and IPv6-only systems. In particular, we should require that all routers in the region be able to calculate routes to both IPv4 and IPv6 addresses, to translate between IPv4 and IPv6 formats, and to know which locally attached hosts are IPv4-only, which are dual, and which are IPv6-only. The last router along the path of the packet therefore translates when necessary. This approach has been called "*Ships In the Night, Translate Only When Necessary*", abbreviated SINTOWN).

6 Conclusion

Of two route dissemination techniques presented in this paper, the route tunneling approach seems to be the more straightforward solution which has the following advantages over the route leaking scheme:

- All types of IPv6 routing prefixes can be disseminated as opposed to the IPv4-compatible IPv6 addresses only.
- Only dual protocol border routers at boundaries between IPv6 and IPv4 routing regions need to implement the tunneling mechanism. All packet encapsulation/ decapsulation is performed by these border routers only. Contrary, since automatic tunneling is utilized in the route leaking scheme, with this scheme packet destination nodes may need to do decapsulation also.
- No need to deploy decapsulating routers for decapsulation of tunneled traffic for which such routers are not tunnel endpoints (termed "early decapsulation" in [4]).
- Since tunnels between IPv6 routing regions are manually configured, there is a better control over data path of IPv6 packet. This facilitates use of IPv6 routing extensions.
- Even though tunnels between isolated routing regions need to be manually configured (as opposed to the automatic tunneling of other scheme), the management of the routing infrastructure can be significantly simpler due to the clear separation of routing information between two different protocol stacks.
- No need to tweak IPv6 routing protocols to process IPv4 routes and vise versa -- there is no cross-pollination.
- It scales better since there is no need to duplicate routing information on dual protocol routers.

The advantages of the route leaking approach are:

- No need to manually configure tunnels between isolated routing regions.
- Packets sent between IPv4-only and IPv6-only nodes can be dynamically routed end-to-end.

Where translation is necessary, it is desirable that all boundary routers are dual-stack and capable of translation. This will allow routing of IPv6 packets with IPv4-compatible addresses and the routing of IPv4 packets to use the same routes.

Since route leaking allows end-to-end routing of packets sent between IPv4-only and IPv6-only nodes, it solves routing problem of directing traffic which needs translation to an appropriate translating router. Nevertheless, authors feel that it may be possible to come with some routing mechanism (e.g. use of default routes) to dynamically route packets between IPv4-only and IPv6-only nodes without resorting to route leaking technique or, at least, use route leaking in some limited way. Further study of routing in the presence of translation is necessary.

7 Security Considerations

It is indicated that use of tunnelling may violate firewalls of underlining routing infrastructure.

No other security issues are discussed in this paper.

8 References

[1] IP Next Generation Overview, R. Hinden, Internet Draft

[2] Internet Protocol, Version 6 (IPv6) Specification, R.Hinden (Editor), Internet Draft.

[3] IP Next Generation Addressing Architecture, R.Hinden (Editor), Internet Draft

[4] Simple IPv6 Transition (SIT) Overview, R. Gilligan, Internet Draft

[5] Transition Mechanism for IPv6 Hosts and Routers, R. Gilligan, E. Nordmark, Internet Draft

9 Authors' Addresses

Dimitry Haskin
Bay Networks, Inc.
2 Federal Street
Billerica, MA 01821
email: dhaskin@baynetworks.com

Ross Callon
Bay Networks, Inc.
3 Federal Street
Billerica, MA 01821
email: rcallon@baynetworks.com