IPv6 Edge Domain Implicit Tunnel (EDIT)
draft-hain-ipv6-edit-01

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 26, 2010.

Copyright Notice

Copyright (c) 2009 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 in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

There will be IPv4-only applications that remain in use as network managers begin deploying IPv6-only networks, or turning off IPv4 in the exiting dual-stack networks. These applications require a dual-
stack capable host operating system, but that OS may find that the local network has not provided on-link IPv4 provisioning or routing support for the resulting packets. Skepticism that this situation will exist is widespread; because it is easy to ignore the costs someone else will incur and focus on the local situation. Taking the system level viewpoint, that skepticism makes no sense when it is widely acknowledged that people will resist replacing something that is working, particularly when the reason it might artificially stop is due to ‘someone else’s problem’. The reality of IPv4-only applications persisting in an IPv6-only provisioned network is specifically ignored by RFC 4038. This document deals with the situation by defining a new tunneling type for use within the edge or leaf network to the border where it interconnects with an IPv4 routing domain.

The draft is being discussed on the ipv6@ietf.org list.

Legal

This documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
Table of Contents

1. Introduction ........................................ ancho
   1.1. Acknowledgements ............................... acks
2. Terminology ....................................... ancho
3. Mechanism .......................................... ancho
   3.1. Format ........................................ ancho
   3.2. Logical Perspective .............................. ancho
   3.3. DHCP ........................................ ancho
   3.4. DNS Issues ....................................... ancho
   3.5. Fragmentation ................................... ancho
4. Domain of applicability and network diagram ............ ancho
5. Sequence ........................................... ancho
6. Open Issues ........................................ ancho
7. IANA Considerations ................................ ancho
8. Security Considerations .............................. ancho
9. References .......................................... ancho
   9.1. Normative References ............................ ancho
   9.2. Informative References ........................... ancho
Author’s Address ....................................... 0
1. Introduction

As network managers begin considering deployment of IPv6-only networks, they stumble across the problem of IPv4-only applications that would continue to work as long as there is a way to transport the IPv4 across their IPv6-only network. The proposed approach known as DSTM [DSTM] was an earlier attempt at solving this problem, which failed to gain traction, likely due to its complexity, but also simply being too far ahead of the timeframe when many people were giving serious consideration to the existence of IPv6-only networks.

As the reality of the IPv4 free pool exhaustion[potaroo][tndh] becomes accepted by network managers, the desire to move straight to IPv6-only is becoming a common theme. While this might seem counter intuitive to some, but is a natural follow-on to the long running industry hype about ‘convergence’. At a minimum, adding a second protocol to run dual-stack would appear to contradict every justification used to get rid of other protocols in favor of IPv4, so there is a strong, possibly unconscious, self-preservation based bias to do a hard cutover. While the goal of running an IPv6-only network is within reach from an infrastructure perspective, the number of IPv4-only applications, that will continue to exist for some time, makes that difficult to achieve in practice. Most people immediately look for an IPv6 to IPv4 address family translator such as IVI [IVI], but the translation approach is designed to deal with the case where one end of the application has switched while the other one has not. There are proposals to translate from IPv4 to IPv6 then back to IPv4 before delivery, yet these are generally dismissed due to complexity. To support an IPv4 api on the endpoint, tunneling of IPv4 over IPv6 to a point that has native IPv4 routing makes more sense than translation, and is less prone to application failure than any 464 translation sequence with the complimentary set of application gateway services.

DS-lite [DS-lite] is a technology which does tunnel IPv4 over IPv6, and is being targeted for environments where the local area network continues to support IPv4 while a portion of the transit network is IPv6-only. It is also targeted at deployment in wired environments where the overhead bits of encapsulation are effectively free.

The radio spectrum limitations of Wireless networks make the bits-per-packet more expensive than wired environments, so removing any unnecessary overhead is a high priority task. Another characteristic of the Wireless network environment is typically the devices are battery powered, so minimizing the number of instructions to prepare a packet for transmission becomes important. Taken together, these important issues lead one to conclude that the simple IPv4-in-IPv6 encapsulation of DS-lite (or DSTM), and header compression requiring
extra battery consumption are not acceptable engineering approaches to meet those specific cost constraints of the wireless environment.

This document details an approach which effectively completely compresses out the inner IPv4 header, without the extra computation of something like VJ header compression [6]. As there are concerns about the system impact of integrating IPv4 routing with IPv6 routing, this approach is limited to the edge or leaf network, where the degree of integration is under local control, and the total amount of detailed routing knowledge will be limited.

This is not a means to route entire IPv4 networks over IPv6 (as DS-lite would be when replacing an IPv4 nat), rather it is a way for an operating system to present a working set of IPv4 api’s to the applications, despite the lack of IPv4 routing on the local network segment. There is no relationship between this mechanism and any IPv6 address that might be used for communicating with other IPv6 endpoints, as the IPv6 address and functionality here is limited to the tunnel interface for transiting IPv4 packets. This adds a function specific IPv6 address; it does not replace the one used by IPv6 applications.

1.1. Acknowledgements

Comments provided by Dan Wing, Fred Baker, and Wojciech Dec

2. Terminology

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

3. Mechanism

3.1. Format

The mechanism defined here is a means to provision and transport IPv4-only applications over IPv6-only network as a tunnel. To minimize transport overhead, the IPv4 header is implicit and completely compressed out when transiting the IPv6-only network, then the IPv6 header is stripped off and replaced by the implied IPv4 one when transiting the IPv4 enabled part of the network. At the boundary between the IPv6-only leaf network, and the IPv4 transit domain, the packet header is replaced by the opposing version before forwarding onward. As this is not a complete version translation, and the end-to-end perceives this is a functional transparent IPv4
path between the api’s, no application layer gateways (ALGs) are required (beyond whatever might have been required when the endpoints actually had IPv4 routing available).

```
| 80 bits | 16 | 32 bits |
+----------------------------------------+--------------------------+
0000..............................0000| FFFF| IPv4 address |
+----------------------------------------+--------------------------+
```

The IPv4-mapped address discussed in section 2.5.5.2 of the IP Version 6 Addressing Architecture [ADDARCH] is used to provision the tunnel interface for IPv6, then the IPv4 address is extracted to provision the IPv4 side of the tunnel interface. Routing of the ::ffff:0/96 prefix is limited to the edge network, and that prefix or anything longer than /96 that might be used for local traffic engineering, does not impact any other IPv6 routing domain. While the simple thing to do would be to just route the /96 to the border router, that would cause all IPv4 traffic within the edge network to bounce off the border which may be sub-optimal. The more logical thing to do would be to route the /120 (if IPv4 would have been a /24), or /96+ what would have been the case if IPv4 routing were provided within the edge network.

3.2. Logical Perspective

```
| IPv6 | IPv4 |
+------|------+
|       |      |
| Physical interface | EDIT tunnel interface |
|                  | (DHCPv4 client) |
```

3.3. DHCP

The goal of the approach is to maintain maximum compatibility for IPv4-only applications, so the existing IPv4 DHCP service will be used. This will mean all IPv4 DHCP options continue to be available. Reaching the DHCP service will require a relay function to exist at the remote border router EDIT tunnel interface, while the local EDIT tunnel interface will need to intercept all IPv4 DHCP messages and forward them to the border router relay function. The IPv6 address
of the DHCP relay function to the IPv4 DHCP service will be the IPv6 anycast address ::ffff:0:3; (ff02::1:2 is the link-scope multicast for all DHCPv6 agents & ff05::1:3 is the site-scope multicast for all DHCPv6 servers ff05::1:4 is the site-scope multicast for all DHCPv6 relays; an RFC 3306 interpretation of those could be ff02::ffff:1:2, ff05::ffff:1:3, & ff05::ffff:1:4, but assuming that the border router is not on the same link as the EDIT enabled device, the link scope one would not work; multicast makes no sense in this context, as the goal is to reach the nearest instance of the relay, not to reach all the members of the group; ::ffff:1:4 would be in the HP/DEC 16/8 block). The DHCP provided lifetime for public IPv4 addresses will be 5 minutes by default in this environment, and the EDIT interface will be tasked with renewing as long as there are active connections, then releasing it when the timer expires after all connections are closed.

The source IPv4 address used during initial conversation with the IPv4 DHCP server would normally be the unspecified address, which using this mechanism would result in an IPv6 source address of ::ffff:0:0. As there is no way for the DHCP relay to route the responses to that address, a functional equivalent of the IPv6 solicited-node multicast group with site-scope will be used ... 224.0.0.1 is the IPv4 group for ‘all hosts’, so ff05::ffff:e0:1 would be the site-scope multicast for all IPv4 endpoints. Therefore every endpoint EDIT interface needs to join that group.

3.4. DNS Issues

The IPv4 side of the stack uses IPv4 to communicate with DNS and retrieve A records (or other RRs) as if there were no IPv6 in the system. There is no translation or synthesis of record types between versions, so dnssec will not be impacted.

3.5. Fragmentation

The MTU on the IPv4 side of the EDIT tunnel interface is the MTU provided to the IPv6 stack minus 20. (Could be set to 1280 just to make sure there is no outbound fragmentation for the local EDIT interface to deal with, as this is intended to be support for legacy apps as the ability to route IPv4 is being withdrawn.)

At the Border Router, fragmentation may be required in either direction, but this is no different than any other mid-network fragmentation that occurs in IPv4.
4. Domain of applicability and network diagram

There can be many border routers, and since the mapping operation is a stateless transform the packets don’t have to follow a symmetric path.

5. Sequence

Step 1) the application on the originating node calls the IPv4 stack API to initiate communication.

Step 2) the OS stack recognizes that the EDIT tunnel interface is not provisioned with an address, and that it is operating in an IPv6-only provisioning environment, so it requests an address from the DHCPv4 service. (This will require a DHCPv4 relay defined to exist at a well-known IPv6 address.)

Step 3) the DHCPv4 service acquires an IPv4 address from the IPv4 pool that this endpoint is assigned to, and the EDIT interface will use the IPv4 address to construct an IPv4-mapped format IPv6 address.

Step 4) the EDIT interface provisions the IPv6 side tunnel interface with the IPv4-mapped address, then the IPv4 side of its stack, and the OS responds to the api call.

Step 5) the application continues the connection attempt, resulting in a packet to the remote endpoint, which is routed internally via the IPv4 side of the stack, until the EDIT tunnel device driver picks
it up and continues processing on the IPv6 side of the stack. The actual packet transmitted by the physical interface is in IPv6 format, using IPv4-mapped addresses.

Step 6) the leaf network routing system forwards the IPv6 packet to the border, where the EDIT tunnel endpoint uses a stateless transform to extract the contents and construct an IPv4 packet based on the embedded IPv4 addresses and forwards that into the IPv4 routing domain.

Step 7) the return packet from the other end system is routed over the IPv4 routing domain to the border based on standard routing of the IPv4 prefix used, where the EDIT tunnel endpoint uses a stateless transform to extract the contents and construct an IPv6 packet based using IPv4-mapped addresses based on the IPv4 addresses in the received packet, then forwards that into the IPv6-only leaf network routing system.

Step 8) the OS stack IPv6 side receives the packet and passes it to the EDIT tunnel device driver which extracts the embedded IPv4 address from the IPv4-mapped format IPv6 address, as well as the contents, then constructs an IPv4 packet and passes the packet to the IPv4 side of the OS stack.

Step 9) once the connection or application is closed, the OS stack sends a message to the DHCPv4 service to explicitly release the reservation on the IPv4 address. (need to change this because ajax would cause constant churn)

6. Open Issues

Detecting when a border router is available on one side but not the other :: blackhole. Routing announcements on each side need to be keyed off the fact that both sides are up.

Tcp options mapping between headers... tunneling -- not translation : is this an IPv6 option that just carries the IPv4 option set intact?

7. IANA Considerations

There are no IANA issues at this time.

8. Security Considerations

The use of an implicit tunnel as described here has the same security
concerns that any tunneling mechanism would have, in that the agents of packet transformation must be trusted.

9. References

9.1. Normative References


9.2. Informative References


Author’s Address

Tony Hain
Cisco Systems
500 108th Ave NE
Bellevue, WA 98004
USA

Phone: +1 425 468-1061
Email: alh-ietf@tndh.net