Abstract

This document presents the concepts and the implementations of address-sharing dual-stateless IPv4/IPv6 translation (dIVI).

The dIVI is an extension of 1:1 stateless IPv4/IPv6 translation with features of IPv4 address sharing and dual translation. The major benefits of those extensions are using public IPv4 addresses more effectively and removing the requirements of DNS64 and ALG, without loosing the stateless, end-to-end address transparency and bidirectional-initiated communications.

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 11, 2012.

Copyright Notice

Copyright (c) 2011 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
Table of Contents

1. Introduction .................................................................................. 3
2. Applicability ............................................................................... 3
3. Terminologies .............................................................................. 4
4. Port-set Algorithm ........................................................................ 4
5. Suffix Coding ............................................................................... 6
   5.1. Extended Address Format ...................................................... 7
   5.2. Suffix Table ........................................................................... 8
6. Dual Stateless Translation .............................................................. 8
   6.1. Header Translation and MTU Handling .................................. 9
   6.2. Port Mapping Algorithm in CPE ........................................... 9
   6.3. Relations to Tunneling .......................................................... 9
7. Implementation Considerations ...................................................... 10
   7.1. Host implementation ............................................................ 10
   7.2. Port Mapping in the Core Translator ..................................... 10
8. Security Considerations ................................................................. 10
9. IANA Considerations .................................................................... 10
10. Acknowledgments ........................................................................ 11
11. References .................................................................................. 11
   11.1. Normative References ......................................................... 11
   11.2. Informative References ....................................................... 12
Appendix A. Testing environment and workflow examples .................. 12
   A.1. The address-sharing host on an IPv6 network initiates communication ............................................. 13
   A.2. A host in the IPv4 Internet initiates communication ................................................................. 13
Authors’ Addresses ........................................................................... 14
1. Introduction

The experiences for the IPv6 deployment in the past 10 years strongly indicate that for a successful transition, the communication between IPv4 and IPv6 address families should be supported.

Recently, the stateless and stateful IPv4/IPv6 translation methods are developed and became the IETF standards. The original stateless IPv4/IPv6 translation (stateless 1:1 IVI) is scalable, maintains the end-to-end address transparency and support both IPv6 initiated and IPv4 initiated communications [RFC6052], [RFC6144], [RFC6145], [RFC6147], [RFC6219]. But it can not use the IPv4 addresses effectively. The stateful IPv4/IPv6 translation can share the IPv4 addresses among IPv6 hosts, but it only supports IPv6 initiated communication [RFC6052], [RFC6144], [RFC6145], [RFC6146], [RFC6147]. In addition, both stateless and stateful IPv4/IPv6 translation technologies require the application layer gateway (ALG) for the applications which embed IP address literals.

In this document, we present concepts and the implementations of the dual-stateless IPv4/IPv6 translation (dIVI). The dIVI can solve the IPv4 address sharing and the ALG problems mentioned above, though still keeps the stateless, end-to-end address transparency and supporting of both IPv6 initiated and IPv4 initiated communications. The dIVI is in the family of stateless IPv4/IPv6 translation, along with IVI and dIVI. These techniques are used for the following scenarios defined in [RFC6144].

- Scenario 1: An IPv6 network to the IPv4 Internet.
- Scenario 2: The IPv4 Internet to an IPv6 network.
- Scenario 5: An IPv6 network to an IPv4 network.
- Scenario 6: An IPv4 network to an IPv6 network.

2. Applicability

The address sharing dual stateless IPv4/IPv6 translation (dIVI) is shown in the following figure.
Where the 1:N XLAT (first translator) is the IPv4/IPv6 translator perform stateless 1:N translation between the IPv4 Internet and an IPv6 network; The CPE.0 to CPE.K (second translators) are 1:1 IPv6/IPv4 translators/IPv6 routers which also perform port mapping function for Host.x when it is communicating with the IPv4 Internet with IPv4 address sharing; the Host.1 to Host.N are the dual-stack hosts.

The IPv6 network routes the IPv6 more specific routes of the IPv4-translatable addresses (longer than /64) defined in Section 5 of this document to corresponding CPEs. In the case of prefix delegation (shorter than /64), the dual stateless IPv4/IPv6 translation with prefix delegation (dIVI-pd) defined in [[I-D.xli-behave-divi-pd]] can be used.

3. Terminologies

This document uses the terminologies defined in [RFC6144].

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

4. Port-set Algorithm

The address-sharing scheme is shown in the following figure.
In the above figure, the IPv4-translatable.0, IPv4-translatable.1, IPv4-translatable.2, ..., IPv4-translatable.K are sharing the same IPv4 address IPv4-addr, but port number range for different IPv4-translatable addresses (i.e. port-range.0, port-range.1, port-range.2, port-range.K) are not overlapped. When an IPv6 node using IPv4-translatable addresses communicates with the IPv4 Internet via a translator, it looks like a single host using IPv4 address (IPv4-addr) communicating with the IPv4 Internet.

There exist many port-range coding schemes and each one may have its advantages and disadvantages, as well as has its best application scenario. In this document, we will introduce a simple one, which we believe is suitable for the IPv4/IPv6 stateless translation. This encoding scheme uses the modulus operator to define the port number range for each host to use.

- For given multiplexing ratio N, the port-set numbers of a IPv4-translatable address with port-set-id K is composed of P=j*N + K + 1024, for all the values of j=0, 1, ..., (65536-N)/N.

- For a destination port number (P), the port-set-id of a given IPv4-translatable address with port-set-id K is determined by the modulo operation: K=((P-1024)%N) (% is the Modulus Operator).

For example, if N=128, then IPv6 node K=5 is only allowed to use port numbers 5, 133, 261, 389, 517, 645, 773, 901, ... 65,413 as the source port, while the packets with these port numbers as the destination port number will be send to IPv6 node K=5.
The modulus operator has several features:

1. The port ranges are not overlapped for different IPv6 nodes.

2. The individual ports for each IPv6 node are not continues and the whole 65536 port range is equally shared by IPv6 nodes.

3. The port number range can be uniquely determined by given sharing ratio (N) and the IPv6 node index (K).

Based on the modulus operator, We need to encode N and K in the suffix as discussed in [RFC6052].

5. Suffix Coding

In Section 2.2 of [RFC6052] IPv4-embedded IPv6 address format is defined which composed of a variable length prefix, the embedded IPv4 address, and a variable length suffix, as presented in the following diagram, in which PL designates the prefix length:

```
+-----+--------------------------+--+-+-+-|
|PL| 0----32--40--48--56--64--72--80--88--96--104-112-120--|
+-----+--------------------------+--+-+-+-|

|32| prefix       |v4(32)       | u | suffix |
+-----+--------------------------+--+-+-+-|

|40| prefix       |v4(24)       | u | (8) | suffix |
+-----+--------------------------+--+-+-+-|

|48| prefix       |v4(16)       | u | (16) | suffix |
+-----+--------------------------+--+-+-+-|

|56| prefix       |     (8)     | u | v4(24) | suffix |
+-----+--------------------------+--+-+-+-|

|64| prefix       | u | v4(32) | suffix |
+-----+--------------------------+--+-+-+-|

|96| prefix        | v4(32)    |
+--------------------------+--+-+-+-|
```

Figure 3: Address Format

In Section 3.5 of [RFC6052], it states:

"There have been proposals to complement stateless translation with a port-range feature. Instead of mapping an IPv4 address to exactly one IPv6 prefix, the options would allow several IPv6 nodes to share an IPv4 address, with each node managing a different range of ports. If a port range extension is needed, it could be defined later, using bits currently reserved as null in the suffix."
This document defines such a suffix encoding scheme with the corresponding port-range algorithm.

5.1. Extended Address Format

It is clear that only Prefix lengths (PL) with 32, 40, 48, 56 and 64 have suffix portions, with maximum suffix lengths of 56, 48, 40, 32, 24, respectively. In order to use different prefix length and unify the port range encoding method, the suffix should be 24 bits or less.

Since the transport port number is a 16 bit integer, the sharing ratio (N) and the port-set index (K) can both have the value from 0 to 65535, which requires 32 bits. In order to fit into 24 bit suffix range, we need to do compression.

First, we can use number of bits to represent the sharing ratio when the sharing ratio is bit-wise, hence 4 bits is enough for N.

Secondly, if sharing ratio N is very high, each IPv6 node can only use a small number of concurrent sessions. For example, if N=4096, each IPv6 node will have 16 concurrent sessions, which may be too small for most of the applications. That may limit the use of some applications, therefore, we set the maximum sharing ratio N=4096, then 12 bits are enough for the IPv6 node index. In this case, we can design suffix which consists of 16 bits for encoding the port range as in the following figures. Note that for different PL, zero padding is needed.

```
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|PL| 0---------32--40--48--56--64--72--80--88--96--104-112-120--|
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|32| prefix  |v4(32)         | u | suffix|      zero         |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|40| prefix        |v4(24)     | u |(8)| suffix|     zero      |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|48| prefix            |v4(16) | u | (16)  | suffix|    zero   |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|56|     prefix                |(8)| u |  v4(24)   | suffix| zero  |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|64|     prefix                    | u |   v4(32)      | suffix|zer|
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 4: Extended Address Format

The port range is therefore embedded in the extended IPv4-translatable IPv6 addresses and bound to the IPv6 node index (K), the packets containing extended IPv4-translatable IPv6 addresses as the destination can be routed to different IPv6 nodes.
5.2. Suffix Table

The suffix table is shown in the following figures, where the most significant 4 bits define the multiplexing ratio and the least significant 12 bits define the port-set index.

<table>
<thead>
<tr>
<th>ratio</th>
<th>suffix range</th>
<th># of Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0000 - 0000</td>
<td>65,536</td>
</tr>
<tr>
<td>2</td>
<td>1000 - 1001</td>
<td>32,768</td>
</tr>
<tr>
<td>4</td>
<td>2000 - 2003</td>
<td>16,384</td>
</tr>
<tr>
<td>8</td>
<td>3000 - 3007</td>
<td>8,192</td>
</tr>
<tr>
<td>16</td>
<td>4000 - 400f</td>
<td>4,096</td>
</tr>
<tr>
<td>32</td>
<td>5000 - 501f</td>
<td>2,048</td>
</tr>
<tr>
<td>64</td>
<td>6000 - 603f</td>
<td>1,024</td>
</tr>
<tr>
<td>128</td>
<td>7000 - 707f</td>
<td>512</td>
</tr>
<tr>
<td>256</td>
<td>8000 - 80ff</td>
<td>256</td>
</tr>
<tr>
<td>512</td>
<td>9000 - 91ff</td>
<td>128</td>
</tr>
<tr>
<td>1,024</td>
<td>a000 - a3ff</td>
<td>64</td>
</tr>
<tr>
<td>2,048</td>
<td>b000 - b7ff</td>
<td>32</td>
</tr>
<tr>
<td>4,096</td>
<td>c000 - cfff</td>
<td>16</td>
</tr>
</tbody>
</table>

Figure 5: Suffix for Port Range Encoding

6. Dual Stateless Translation

In general, dual stateful IPv4/IPv6 translations (IPv4-IPv6-IPv4) are considered harmful, since the two translators cannot synchronize the parameter to translate the IPv4 address to its original format. However, the dual stateless IPv4/IPv6 translation (IPv4-IPv6-IPv4) can translate the IPv4 address to its original format based on an algorithm. There are several reasons for using dual stateless IPv4/IPv6 translation.

- The second translator (CPE) performs the port-set mapping algorithm defined in Section 4, therefore, there is no need to modify the host with IPv4 address sharing.
- ALG cannot be developed for some applications, or has not been developed during the early transition stage.
- DNS64 is not allowed (due to DNSSec, etc.).
6.1. Header Translation and MTU Handling

The general header and ICMP translation specifications are defined in [RFC6145].

Special MTU and fragmentation actions must be taken in the case of dual translation.

6.2. Port Mapping Algorithm in CPE

The port mapping algorithm is straightforward. The port mapping device maintains a database of allowed port numbers defined by the extended IPv4-translatable address format. If the packets from the host contains the source port number which do not match the port number range defined by the extended IPv4-translatable address format, the CPE will map the source port number to an allowed one and keep the record in the database for mapping back the returning packets and all the packets in the same session.

The port number database can be refreshed via the corresponding transport layer flags for TCP or via timeout for UDP sessions.

6.3. Relations to Tunneling

When dual stateless IPv4/IPv6 translation is deployed, its behavior is similar to tunneling. Tunneling do not require DNS64 and ALG., because the communication occurs in same address family. Dual translation don’t need DNS64 and ALG as well, even in each translator the communication occurs between different address families. However, there are following differences:

- Scalability. Dual stateless translation is based on routing, there is nothing needed to maintain in the translator, operator’s management loads are minimum compared with tunneling scheme, which has to maintain tunnel states.

- Low OPEX. Dual stateless translation can do traffic engineering and flow analysis without decapsulation which is a must in tunnel case.

- Header Compression. The dual stateless IPv4/IPv6 translation does not need to do encapsulation and at least 20 octets header overhead are reduced.

- No performance bottlenecks. The dual stateless translation handles the fragmented packets by hosts, while the tunneling handles the fragmented packets in the tunnel end points. Therefore, there is no performance bottlenecks.
TRANSPARENT transition to IPv6. The dual stateless translation can be treated as a special case of single stateless translation, the first XLAT performs exactly the same function, no matter there is a XLAT.x or not. Hence it is a unified approach, rather than special setup for the coexistence and transition. This is to say that the ISP can deploy IPv6-only network with XLAT, so the IPv6-only hosts can communicate with both the IPv6 Internet and the IPv4 Internet. However, if for some reason a specific ALG cannot be supported, and for users, who need that specific application, can deploy XLAT.x. When the application is updated, the XLAT.x can be removed. There is nothing to change to XLAT. With more and more contents and users move to IPv6, the working load of XLAT will be less and less, and eventually can be removed. The whole process is transparent, smooth and incremental.

7. Implementation Considerations

7.1. Host implementation

For the wireless mobile Internet environment, it is not difficult to modify the operating system of the mobile device, therefore it possible to integrate the port mapping algorithm and the second IPv4/IPv6 translation function in the mobile device, which is an IPv6-only host to the network and has a dual-stack socket API for the applications running on this host.

7.2. Port Mapping in the Core Translator

The port mapping can be performed in the core translator, in this case, in this case, there is no need to have the second translator (CPE.x) and no need to modify the IPv6 host for the port restriction requirements.

8. Security Considerations

There are no security considerations in this document.

9. IANA Considerations

This memo adds no new IANA considerations.

Note to RFC Editor: This section will have served its purpose if it correctly tells IANA that no new assignments or registries are required, or if those assignments or registries are created during the RFC publication process. From the author’s perspective, it may
therefore be removed upon publication as an RFC at the RFC Editor’s discretion.

10. Acknowledgments

The authors would like to acknowledge the following contributors in the different phases of the address-sharing IVI and dIVI development: Maoke Chen, Hong Zhang, Weifeng Jiang, Yuncehng Zhu and Guoliang Han.

The authors would like to acknowledge the following contributors who provided helpful inputs: Dan Wing, Fred Baker, Dave Thaler, Randy Bush, Kevin Yin, Wojciech Dec and Rajiv Asati.

11. References

11.1. Normative References


11.2. Informative References

[I-D.xli-behave-divi-pd]
draft-xli-behave-divi-pd-00 (work in progress), July 2011.

Appendix A. Testing environment and workflow examples

The prefix we selected is 2001:da8:a4a6::/48. The IPv4-converted address and the IPv4-translatable address are:

IPv4-converted address
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|PL| 0-------------32--40--48--56--64--72--80--88--96--104-112-120-|
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|48|     prefix            |v4(16) | u | (16)  | zero              |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

IPv4-translatable address
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|PL| 0-------------32--40--48--56--64--72--80--88--96--104-112-120-|
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|48|     prefix            |v4(16) | u | (16)  | suffix|    zero   |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

Figure 6: Address format

The IPv4 address sharing ratio is 16. So the port-set index is

<table>
<thead>
<tr>
<th>ratio</th>
<th>suffix range</th>
<th># of Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>4000 - 400f</td>
<td>4,096</td>
</tr>
</tbody>
</table>

Figure 7: Address format

The public IPv4 address is 202.38.117.0/24.

The host in the IPv4 Internet is 202.112.35.254 and the corresponding IPv4-converted address is 2001:da8:b4b6:ca:7023:fe00::.

The shared IPv4 address is 202.38.117.1 with sharing ratio 16, the corresponding IPv4-translatable addresses are 2001:da8:b4b6:ca:2675:

A.1. The address-sharing host on an IPv6 network initiates communication

An address-sharing Host.0 (202.38.117.1) in an IPv6 network behind CPE initiates communication with Host 202.112.35.254:80 in the IPv4 Internet

On the Host.0
Src#p= 202.38.117.1#1881 (random port)
Dst#p= 202.112.35.254#80 (server port)

On an IPv6 network
Src#p= [2001:da8:b4b6:ca:2675:0140:100::]#8192 (CPE mapped port)
Dst#p= [2001:da8:b4b6:ca:7023:fe00::]#80 (server port)

On the IPv4 Internet
Src#p= 202.38.117.1#8192 (CPE mapped port)
Dst#p= 202.112.35.254#80 (server port)

Figure 8: Example 1

The returning packets reverse the Src and Dst, the CPE maps the "CPE mapped port (8192)" back to the original "random port (1881)".

A.2. A host in the IPv4 Internet initiates communication

A host 202.112.35.254 in the IPv4 Internet initiates communication to the address-sharing Host.0 (202.38.117.1#2000)

On the IPv4 Internet
Src#p= 202.112.35.254#36567 (random port)
Dst#p= 202.38.117.1#2000 (server port)

On an IPv6 network
Src#p= [2001:da8:b4b6:ca:7023:fe00::]#36567 (random port)

On the Host.0
Src#p= 202.112.35.254#36567 (random port)
Dst#p= 202.38.117.1#2000 (server port)

Figure 9: Example 2

The returning packets reverse the Src and Dst.
Authors’ Addresses

Congxiao Bao
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN
Phone: +86 10-62785983
Email: congxiao@cernet.edu.cn

Xing Li
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN
Phone: +86 10-62785983
Email: xing@cernet.edu.cn

Yu Zhai
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN
Phone: +86 10-62785983
Email: jacky.zhai@gmail.com

Wentao Shang
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN
Phone: +86 10-62785983
Email: wentaoshang@gmail.com