Internet Protocol, Version 4 64-bit Address Extension
draft-tian-ipv4-64-bit-00

Abstract

This document describes a 64-bit address mechanism that serves as an extension to IPv4, namely, IPv4_64. As an add-on to IPv4, this extension is designed to solve the IPv4 address exhaustion issue in an innovative way, and slightly improve some IPv4 functions at the same time.

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/1id-abstracts.html

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

Copyright Notice

Copyright (c) 2019 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 Motivation ........................................... 4
1. Introduction

Although IPv4 address exhaustion has occurred, due to the risk, cost and complexity of applying new network technologies, it is clear that IPv4 is going to exist for a long time. But current methods used to prolong the lifespan of IPv4 can only alleviated IPv4 address exhaustion issue to some extent. Applying those technologies will inevitably face the continuously growing costs of running the IPv4 network and the deficiency of the network connectivity. This document describes a 64-bit address mechanism that serves as the extension to IPv4, namely, IPv4_64. As an add-on to IPv4, this extension is designed to solve the IPv4 address depleted issue in an innovative way, and slightly improve some IPv4 functions at the same time.

The choice of 64-bit is because that modern computer science has favoured 64-bit more than other number of bits as the upgrade from 32-bit. The 64-bit addresses will be more natural for modern computer and human to handle.

1.1 Motivation

To mitigate IPv4 address exhaustion issue, a 64-bit address mechanism is introduced. And a new header format is introduced to keep those 64-bit address information. When the originator or recipient is 64-bit address or need to bear any other 64-bit address information in the header, the IPv4_64 header is used instead of the IPv4 one.

The introduction of the IPv4_64 header also makes it possible to improve some functions of IPv4 while not breaking current IPv4 network functions. Most of those improvements are heavenly borrowed from IPv6, since IPv6 has done an excellent job in improving IPv4 functions.

1.2 IPv4_64 Header

In summary, the content of the IPv4_64 header falls primarily into the following categories:

- Expanded Addressing Capabilities

  IPv4_64 increases the IP address size from 32-bit to 64-bit, to support more levels of addressing hierarchy, a much greater number of addressable nodes, in a way that keeps compatible with IPv4. No different configuration method of address is
needed. Any new configuration method must support IPv4 networks as well. Also, IPv4_64 64-bit addresses and IPv4 32-bit addresses share the same address space.

- **Header Format Simplification**

  The IPv4_64 header format makes it possible to implement some simplification to IPv4 header. Like IPv6, some IPv4 header fields have been made optional and removed from the header, to reduce the common-case processing cost of packet handling and to limit the bandwidth cost of the IPv4_64 header.

- **Improved Support for Extensions and Options**

  Like IPv6, changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

- **Authentication and Privacy Capabilities**

  Like IPv6, extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv4_64 is also needed.

This document specifies the basic format of IPv4_64 header and the initially defined extension and options for those IPv4 fields that have been removed from the header. It also discusses the IPv4-compatible considerations.

### 2. Terminology

- **IPv4_64** IPv4 64-bit address extension.
- **node** a device that implements IPv4 or IPv4_64.
- **upper layer** see [RFC8200].
- **packet** an IPv4 or IPv4_64 header plus payload.

### 3. IPv4_64 Header Format

```
+-------------------------------+-------------------------------+
| Version | Res | Type of Service |       Reserved                |
|-------------------------------+-------------------------------+
| Payload Length |  Next Header |  Time to Live |
|-------------------------------+-------------------------------+
|                  64-bit Source Address |
|-------------------------------+-------------------------------+
|                      |               |
|                  64-bit Destination Address |
|-------------------------------+-------------------------------+
```

- **Version** 4-bit Internet Protocol number, be 4. See [RFC791]
- **Res** 4-bit reserved field, must be 0.
Type of Service 8-bit type of service field, same as IPv4.

Reserved 16-bit reserved field. The value initialized by the originator should be delivered to the recipient without being changed.

Payload Length 16-bit unsigned integer, same as IPv6. Length of the protocol payload, i.e., the rest of the packet following this header, in octets, including any extension headers.

Next Header 8-bit selector, same as IPv6. Identifies the type of header immediately following this header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].

Time to Live 8-bit unsigned integer, same as IPv4.

Source Address 64-bit address of the originator of the packet.

Destination Address 64-bit address of the intended recipient of the packet (possibly not the ultimate recipient, if a Routing header is present).

4. IPv4_64 Extension Headers

IPv6 has already devised the most sophisticated extension headers. Except some changes needed due to IPv4-compatible requirements and the 64-bit address space, IPv4_64 extension headers are identical to the IPv6 ones described in [RFC8200].

IPv4_64 extension headers have the following modifications:

* All the address fields should be changed to 64-bit.

* References to the IPv6 header are replaced by references to the IPv4_64 header.

* ICMP errors sent in the course of processing extension headers use ICMPv4 instead of ICMPv6.

* Extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv4_64 is also needed.

5. IPv4-Compatible Considerations

IPv4_64 headers are designed to work an add-on to IPv4. No different configuration method needed, and any new configuration method for IPv4_64 must support IPv4 network as well.

5.1 Res Field

The value of the IHL field of IPv4 will always be equal to or greater than 5 and the value 0, 1, 2, 3 and 4 are not used. IPv4_64 headers use that field as Res field and set it to 0 to distinguish with IPv4 ones.

5.2 IPv4-Compatible IPv4_64 Address

The IPv4-Compatible IPv4_64 Address is used to address IPv4 nodes.
The format of that address is as follows:

```
+--------------------------------+--------------------------------+
<table>
<thead>
<tr>
<th>32 bits</th>
<th>32 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 address</td>
<td>--------------------------------</td>
</tr>
</tbody>
</table>
```

5.3 The Loose-in and Strict-out Rule

To work as an add-on to IPv4 and not breaking current IPv4 functions, IPv4_64 takes the loose-in and strict-out rule. When originating a packet, if possible, IPv4 headers should always be used. While at the recipient, either the IPv4 header or the IPv4_64 header are equally accepted and processed.

6. Packet Size Issues

Even IPv4_64 headers slightly increase the header size by 4-byte, the MTU issues will arise. IPv4_64 follows the design of IPv6, requires that every link in the Internet have a MTU of 1280 octets or greater. Furthermore IPv4_64 also requires that every tunneling link in the Internet (for example, PPP links [RFC1661]) have a MTU of the sum of 1280 octets + the tunneling overhead or greater.

7. IANA Considerations

IANA is requested to update the descriptions of IPv6 extension headers to reflect that they are not IPv6 specific (such as, can be applied to IPv4 and its extensions). In the Assigned Internet Protocol Numbers Registry, the modified protocols descriptions are:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Keyword</th>
<th>Protocol</th>
<th>IPv6 Extension header</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HOPOPT</td>
<td>Hop-by-Hop</td>
<td></td>
<td>[RFC8200]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Option</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>Route</td>
<td>Routing</td>
<td></td>
<td>[Steve_Deering]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Header</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Frag</td>
<td>Fragment</td>
<td></td>
<td>[Steve_Deering]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Header</td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>NoNxt</td>
<td>No Next</td>
<td></td>
<td>[RFC8200]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Header</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>Optns</td>
<td>Destination</td>
<td></td>
<td>[RFC8200]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Security Considerations

IPv4_64, from the viewpoint of the basic format and transmission of packets, has security properties that are similar to IPv4 and IPv6. Same to IPv6 and IPv4, these security issues include:
- Eavesdropping, where on-path elements can observe the whole packet (including both contents and metadata) of each IPv4_64 datagram.
- Replay, where the attacker records a sequence of packets off of the wire and plays them back to the party that originally received them.
- Packet insertion, where the attacker forges a packet with some chosen set of properties and injects it into the network.
- Packet deletion, where the attacker removes a packet from the wire.
- Packet modification, where the attacker removes a packet from the wire, modifies it, and re-injects it into the network.
- Man-in-the-middle (MITM) attacks, where the attacker subverts the communication stream in order to pose as the sender to receiver and the receiver to the sender.
- Denial-of-service (DoS) attacks, where the attacker sends large amounts of legitimate traffic to a destination to overwhelm it.

In modern network, the upper-layer protocols such as Transport Layer Security (TLS) or Secure Shell (SSH) can be used to protect the application-layer traffic running on top of IPv4_64. IPv4_64 packets can also be protected from eavesdropping, replay, packet insertion, packet modification, and MITM attacks by updating the "Security Architecture for the Internet Protocol" [RFC4301] to support IPv4_64 headers.

There is not any mechanism to protect against DoS attacks. Defending against these type of attacks is outside the scope of this specification.

Similar to IPv6, IPv4_64 addresses of nodes are expected to be more visible on the Internet. This creates some additional privacy issues such as making it easier to distinguish endpoints.

Using the same extension header architecture with IPv6 also makes IPv4_64 face the similar security challenges with IPv6. See [RFC8200] for more information.

8. REFERENCES

8.1. Normative References


8.2. Informative References


Acknowledgments

The authors gratefully acknowledge the many helpful suggestions of the members of the Internet community at large.

Authors’ Addresses

Bo Tian
Calix

Email: bo.tian@calix.com