

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 3 October 2025

L. Donnerhacke  
1 April 2025

IPv5 -- The missing transition step  
draft-april-ipv5-02

Abstract

Despite IPv6 was developed to replace IPv4, the transition process did not and will not ramp up as expected. Major reason for stalled deployment is non-technical, but historical knowledge as well as established processes. The missing part between IPv4 and IPv6 is designed to help the migration efforts by addressing the non-technical needs for a smooth transition.

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 <https://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 3 October 2025.

Copyright Notice

Copyright (c) 2025 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 (<https://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.

## Table of Contents

1. Introduction . . . . .	2
2. IPv5 Header Format . . . . .	2
3. Address resolution . . . . .	3
4. IPv5 Address Mapping . . . . .	3
4.1. Mapping IPv4 into IPv5 . . . . .	4
4.2. Mapping IPv5 into IPv6 . . . . .	4
5. IPv5 Netmasks . . . . .	4
5.1. Mapping IPv4 into IPv5 . . . . .	4
5.2. Mapping IPv5 into IPv6 . . . . .	5
6. Allocating resources . . . . .	6
6.1. Automatic reuse of existing resources . . . . .	6
6.2. Extended resources . . . . .	6
7. Host address assignment . . . . .	7
8. Reserved space . . . . .	7
9. Security considerations . . . . .	7
10. IANA considerations . . . . .	7
11. References . . . . .	7
11.1. Normative References . . . . .	7
11.2. Informative References . . . . .	8
Appendix A. Change history . . . . .	8
Author's Address . . . . .	8

## 1. Introduction

IPv4 [RFC791] comes with the visual appearance of `_dotted-quad-numbers_`, while IPv6 [RFC8220] looks like a `_memory dump_` during a kernel panic. Senior administrators as well as teachers, documentation writers, or film makers are familiar with the IPv4 style of addresses, so they can and will not change their habits, slides or books. IPv6 will not get traction until this generation of people left the companies and schools.

In order to introduce a new protocol, the visual and operation experience must not change. Coming from IPv4, there should be no difference in using IPv5. Evolving to IPv6, there should be only a minimal difference to IPv5. IPv5 is designed in a way to keep the older generation of people on board while introducing a protocol, which might be used like the newer IPv6.

Furthermore, the organizational impact should be minimized. People already using IPv4 or IPv6 should be able to switch to IPv5 without additional bureaucratic obstacles.

## 2. IPv5 Header Format

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|Version| Traffic Class |           Flow Label           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Payload Length           | Next Header | Hop Limit |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |
+                                     +
|                               Source Address                               |
+                                     +
|                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |
+                                     +
|                               Destination Address                               |
+                                     +
|                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv5 header is using the same fields as the IPv6 header. The major differences are:

- \* the version number is 5
- \* the address fields are 64bit each

For information about the other fields, please refer to IPv6 [RFC8220].

### 3. Address resolution

For address resolution an adapted version of neighbor discovery [RFC4861] is used.

The last 24bit of IPv5 addresses are mapped to the IANA-EUI [RFC9542] space 33:33:ff: for generating multicast addresses.

### 4. IPv5 Address Mapping

IPv5 addresses consist of four hextets a 16bit each. For the visual representation, those grouping are used. The hextets might be written in decimal, separated by dot '.' characters, or as hexadecimal numbers, separated by colon ':'. Intermixing both methods is not allowed. Multiple hextets of 0 might be concatenated to "::", so ":::1" is the same as "0:0:0:1". For decimal-dotted notation, all four decimals must be retrained, shortening is not allowed.

Decimal-dotted notation is used for all addresses which has lower than 4096 in the first hextet, otherwise hexadecimal-colon notation is used. So 987.543.210.1 is correct, but 987:543:210:1 is not. OTOH 1234:5678:9:0 is correct, but 12345.6789.0.1 is not.

#### 4.1. Mapping IPv4 into IPv5

Each octet of an IPv4 address is mapped into the corresponding hextet of the IPv5 address. In order to fill the double space, each component of the address is separately doubled in size.

The address IPv4 "198.51.100.24" will become a IPv5 address consisting of those eight bytes: 0 198 0 51 0 100 0 24. This IPv5 address is written "198.51.100.24" in decimal-dotted form.

#### 4.2. Mapping IPv5 into IPv6

IPv6 addresses are twice as large as IPv5 addresses. The first two hextets of the IPv5 address maps directly into the first hextets of the IPv6 address. The third hextet of the IPv5 address maps into the fourth hextet in IPv6, and the last hextet maps into the last hextet respectively.

So the IPv5 address "2001:db8:1234:5678" map to the IPv6 address "2001:db8:0:1234::5678".

### 5. IPv5 Netmasks

In order to keep the operational differences minimal, IPv4 as well as IPv6 addresses with their typical netmasks should be representable by IPv5 without change.

For routing purposes the *\*prefix length\** is used, which is different from the *\*netmask\**. Routing and networking people should ignore the netmask, and stick only to the prefix length for all practical purposes. If necessary, the prefix length should be written with a double slash "/", instead of a single slash "/" used by the netmask.

#### 5.1. Mapping IPv4 into IPv5

IPv4	netmask	prefix length	IPv4	netmask	prefix length
/0	/0	//0	/16	/16	//32
/1	/1	//9	/17	/17	//41
/2	/2	//10	/18	/18	//42
/3	/3	//11	/19	/19	//43
/4	/4	//12	/20	/20	//44

/5	/5	//13	/21	/21	//45	
/6	/6	//14	/22	/22	//46	
/7	/7	//15	/23	/23	//47	

Table 1

IPv4	netmask	prefix length	IPv4	netmask	prefix length	
/8	/8	//16	/24	/24	//48	
/9	/9	//25	/25	/25	//57	
/10	/10	//26	/26	/26	//58	
/11	/11	//27	/27	/27	//59	
/12	/12	//28	/28	/28	//60	
/13	/13	//29	/29	/29	//61	
/14	/14	//30	/30	/30	//62	
/15	/15	//31	/31	/31	//63	
			/32	/32	//64	

Table 2

This mapping ensures, that the existing netmask in IPv4 can be reused in IPv5 without change. Furthermore the typical network classes can be retrained: A class C is still a /24, and suitable for typical LAN setups.

## 5.2. Mapping IPv5 into IPv6

The first 32 bit prefix lengths are mapped 1:1 to IPv6 prefix lengths, and are identical to the netmasks.

Prefix length between //32 and //48 are mapped uniformly to the range 32 up to 64, so each step is doubled.

Prefix length between /48 and /64 are mapped uniformly to the range 64 up to 128, so each step is quadrupled.

If "n" denotes the netmask and "p" the prefix length, then the following formula holds:

$$n = \begin{cases} p & , \text{ if } 0 \leq p \leq 32 \\ 32 + 2*(p-32) & , \text{ if } 32 < p \leq 48 \\ 64 + 4*(p-48) & , \text{ if } 48 < p \leq 64 \end{cases}$$

This mapping ensures, that the existing netmask in IPv6 can be reused in IPv5 without change. Furthermore the typical network classes can be retrained: A class C is still a /64, and suitable for typical LAN setups.

## 6. Allocating resources

### 6.1. Automatic reuse of existing resources

Any existing network, which is using registered IPv4 or IPv6 space, can automatically use the corresponding IPv5 space. So if a network uses 2001:db8::/32 in IPv6, it can use 2001:db8::/32 in IPv5, too. If a network uses 192.0.2.0/24 in IPv4, it can use 192.0.2.0/24 in IPv5, too.

This simplified resource allocation process is possible for any IPv6 resource up to prefix length of /32, and for any IPv4 resource up to prefix length of /24.

### 6.2. Extended resources

By applying the previous allocation scheme, not all addresses are exhausted.

The obvious extension is to go "full decimal", as proposed by various film makers: Simply allow up to 999 in each hextet. So you can - without asking for extra permission - extend your networks in your prefix 128.66.0.0/16 by assigning class C networks up to 128.66.999.0/24. RIRs might apply for additional allocations from IANA up to 999.0.0.0/8.

Using more than three digits may be used, as long as the first hextet is excluded. More than 65535 per hextet must not be used.

Please check your applications, if they are able to handle more than three digits per hextet.

## 7. Host address assignment

Host addresses might be assigned - as usual - manually or by DHCP. The DHCP [RFC8415] protocol needs to be adapted for this purpose. Automatic assignment technologies like SLAAC [RFC4862] are not applicable.

For hosts in the class C network /24, you can assign any host value up to 999, and - with caution - up to 65534, in the last hextet.

For hosts in the class C network /64, host assignments can use the full space from 1 up to fffe in the last hextet.

## 8. Reserved space

The space 1000.0.0.0 - 1000:: is reserved for future extensions.

The space 4000:: - ffff:: is reserved for future extensions.

## 9. Security considerations

There are no related security considerations, besides those, which are already known from IPv4 and IPv6.

## 10. IANA considerations

IANA should open an new registry for the IPv5 address space.

## 11. References

### 11.1. Normative References

- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/rfc/rfc4861>>.
- [RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/rfc/rfc791>>.
- [RFC8220] Dornon, O., Kotalwar, J., Hemige, V., Qiu, R., and Z. Zhang, "Protocol Independent Multicast (PIM) over Virtual Private LAN Service (VPLS)", RFC 8220, DOI 10.17487/RFC8220, September 2017, <<https://www.rfc-editor.org/rfc/rfc8220>>.

- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/rfc/rfc8415>>.
- [RFC9542] Eastlake 3rd, D., Abley, J., and Y. Li, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 9542, DOI 10.17487/RFC9542, April 2024, <<https://www.rfc-editor.org/rfc/rfc9542>>.

## 11.2. Informative References

- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/rfc/rfc4862>>.

## Appendix A. Change history

### version 02

- \* fixed incorrect backmatter mark

### version 01

- \* fixed spelling errors (lldp)
- \* fixed maximum hex value for hosts (lldp)

### version 00

- \* initial draft

## Author's Address

Lutz Donnerhacke  
Jena  
Germany  
Email: [lutz@donnerhacke.de](mailto:lutz@donnerhacke.de)