Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 22 September 2026

W. Augustyn, Ed.
21 March 2026

IP Addressing with References (IPREF)
draft-augustyn-intarea-ipref-07

Abstract

IP addressing with references, or IPREF for short, is a method for end-to-end communication across different address spaces normally not reachable through native means. IPREF uses references to addresses instead of real addresses. It allows to reach across NAT/NAT6 and across protocols IPv4/IPv6. It is a pure layer 3 addressing feature that works with existing network protocols.

IPREF forms addresses (IPREF addresses) made of context addresses and references. These IPREF addresses are publishable in Domain Name System (DNS). Any host in any address space, including behind NAT/NAT6 or employing different protocol IPv4/IPv6, may publish IPREF addresses of its services in DNS. These services will be reachable from any address space, including those running different protocol IPv4/IPv6 or behind NAT/NAT6, provided both ends support IPREF.

IPREF provides much needed IPv4/IPv6 compatibility for the de facto mixed protocol Internet.

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 22 September 2026.

Table of Contents

1. Introduction

Some 25 years ago, it became clear that the 32 bit address space of the IP protocol was too small for the growing Internet. An effort to avert the impending 'shortage of IP addressees', as the problem was referred to at the time, led to development of a new IP protocol named version 6. That protocol, referred to as IPv6, had a larger address space thus instantaneously solving the problem. Unfortunately, it created another problem. It operated in a different, incompatible address space than the existing IP protocol, referred to since as IPv4. In this way, two different address spaces, inaccessible to each other, have been created.

Meanwhile, a grass roots effort led to development of another solution to the 'shortage of IP addresses' problem that was compatible with existing IPv4 protocol. IPv4 address space was extended by rewriting network addresses. That technique, since widely known as NAT, effectively created an address space hierarchy. At the top of the hierarchy, there was the original 32 bit address space. These addresses, at the edge of the hierarchy, were translated into and from another set of addresses belonging to so called 'private ranges'. These addresses formed new address spaces. They were 32 bit wide but only 24 bit subsets were used in practice. Together with the top hierarchy, they produced a 56 bit address space for IPv4. Another variant, known as 'carrier grade NAT', added another level of hierarchy which extended total address space to 72 bits. This approach averted the shortage of IP addresses, but it achieved it at a cost of creating another problem. Namely, it created millions of address spaces inaccessible from one another. These new addresses were not equal value, they were mostly useful for clients accessing servers within their own address spaces or within the space at the top of the hierarchy, commonly referred to as 'global IP addresses'.

As a result, the Internet evolved into a collection of a large number of generally inaccessible address spaces with two incompatible network protocols.

There were attempts at improving the situation which went in two directions. There were attempts aiming to solve access across address spaces within the same protocol, typically referred to as 'NAT traversal solutions', and there were attempts aiming to bridge IPv4/IPv6 divide. All of these attempts resulted in partial solutions with substantial limitations. We could call them nicely 'focused'. The ones dealing with IPv4/IPv6, like NAT64/SIIT, were asymmetrical, worked differently in one direction than the other, didn't scale well, required global IPv4 addresses, couldn't traverse NAT. The ones dealing with NAT traversal, like STUN or NAT-T

collection, didn’t work across protocols IPv4/IPv6, were very limited, dealing mostly with port manipulation, never truly solving NAT traversal.

The result of these attempts was a hodgepodge of limited, mostly lacking, poorly scaling, half measures which left these millions of different address spaces still generally inaccessible.

A closer analysis of these different solutions reveals they suffer great difficulties primarily because they try to render necessary address transformations too early. IPREF takes a different approach. It is based on the observation that the originating hosts do not have to know what the destination addresses are, or what protocol they belong to. Thus, IPREF uses references to addresses rather than real addresses. This approach works for both NAT traversal as well as for protocol traversal since both problems are substantially the same. The one difference between them being having to repackage packets on the IPv4/IPv6 boundary. Such repackaging requires sufficient compatibility between the protocols which fortunately exists.

With this approach, IPREF does not need to negotiate anything. It does not need any external devices, any shared configurations. It does not require allocating of any global IP addresses. It does not create any new address spaces, it always refers to existing ones. The result is a highly scalable solution that works over NAT, NAT6, filters, and across IPv4/IPv6. All of the millions of different address spaces, whether behind NAT/NAT6, or across IPv4/IPv6 may now communicate directly using IPREF.

In the background to all the above, there is an ongoing effort to convert every IPv4 network out there to IPv6. The idea is to run only one protocol everywhere which would simplify many issues including address space accessibility while complicating other issues perhaps of greater significance. For one, such a return to a single address space, single protocol environment runs against a broad, long running trend of diverging interest of the Internet providers versus the interests of their customers. The appearance of such ideas like NAT6 is symptomatic which should give the proponents of IPv6 everywhere a pause. Indeed, the customers of the Internet fiercely defend their independence which is the primary reason why the transition to IPv6 has been resisted for so long.

In this context, IPREF offers a compelling alternative. Its ability to make IPv4 and IPv6 compatible allows the ISPs to deploy IPv6 Internet without replacing customer networks. In this way ISPs get what they want and their customers also get what they want. Thus the Internet may evolve into a compatible Multiprotocol Internet. It will not only stop massive replacements, it will also provide a path for future upgrades to IPv6, where a new protocol could be introduced gradually while allowing the customers to keep their networks.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. IPREF Terminology

IPREF - short name referring to technology described in this document, pronounced I-P-REF

context address - a native address component of an IPREF address

reference - an unsigned integer component of an IPREF address

IPREF address - addressing unit used with IPREF, a combination of a context address and a reference

encoding network - network representing IPREF addresses in native form

IPREF gateway - a device, or software component, that performs IPREF address rewriting

2. IPREF Overview

IPREF is a method for communication across different address spaces, that is those that cannot be reached through native means of a network protocol. Examples of such address spaces are networks behind NAT, NAT6, or filters, as well as networks employing different network protocols such as IPv4/IPv6.

Key characteristics of IPREF:

* massively scalable

* cross protocol, cross address space (such as NAT/NAT6)

* strictly layer 3 (no port manipulation, all ports available)

* addresses publishable in DNS

* no need for external translators or shared configurations

* no need for global IPv4 addresses

* no dual stacks

2.1. References

A common problem with cross address space communication is to decide how to deal with addressing. Communicating peers cannot interpret each other's addresses therefore some other method of directing packets to proper destinations must be devised.

IPREF solves this problem by using references to addresses rather than actual addresses. The references are then evaluated within each address space to render proper native addresses suitable for forwarding. The evaluation is performed at each address space independently. As a result the same references render different addresses at different address spaces.

References are unsigned integers meaningful only in the context of the address space they pertain to. Local administrators allocate values to references and assign their meaning. They may divide a reference into fields for any purpose believed useful. For example, the fields may reflect different sources of allocations, or provide extra security bits, or provide load balancing hints, etc. Peers are unaware of these fields and treat references as opaque values.

2.2. IPREF Addresses

IPREF uses references to addresses in lieu of actual addresses. Such references need context to be meaningful, therefore IPREF uses addressing units that combine a context address, which MUST be a native address, and a reference. These addressing units are called IPREF addresses.

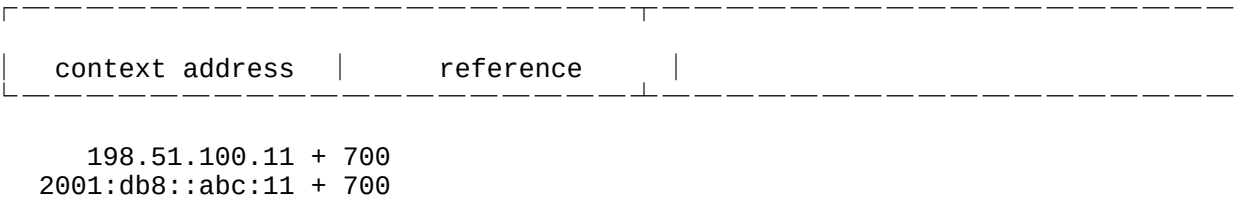


Figure 1: IPREF address

The context address portion of an IPREF address is an address within adjacent address space. Typically, this is the Internet, so this is usually an Internet address of an edge router of the source or destination network. Other addresses known to the network may also be used.

References are allocated by administrators of respective address spaces. The references may only refer to hosts from these address spaces. Thus, the destination references are allocated by the administrators of the destination networks while the source references are allocated by the administrators of the source networks. There are no conflicts and there is no need for negotiations. Each peer defines their own references and accepts peer references as opaque values.

Figure 1 shows two examples of IPREF addresses. In the notation, a plus sign '+' separates context address from the reference. A packet carries two IPREF addresses: a source IPREF address and a destination IPREF address.

IPREF addresses are publishable in DNS.

2.3. Gateways and Encoding Networks

IPREF addresses are placed in packets by gateways. The gateways must be installed at each address space participating in IPREF communication. Intermediate address spaces, such as the Internet, don't need gateways.

The gateways inject and remove IPREF addresses as packets leave and enter local networks. They also encode IPREF addresses for internal use as local network protocols only understand native addresses. The encoding networks are local private networks to which the gateways map IPREF addresses. For example, IPv4 networks could use a 10/8 network while IPv6 networks could use an fd::/64 network for the purpose. For local hosts, these encoded addresses represent peers they communicate with. The peers appear as if on the local network. This is common with cross protocol communication or more generally with cross address space communication. The gateways replace these encoded addresses with IPREF addresses when sending packets outside local networks.

An IPREF gateway is essentially a router. It is referred to as a gateway to emphasize its role in cross address space communication.

2.4. Traversing Address Spaces

Similarly to standard IP protocols, IPREF packets travel through address spaces based on source and destination IPREF addresses. After all, they utilize existing protocols for transport thus they must obey the same rules. The IPREF addresses are interpreted at each address space to render native addresses for forwarding.

	Network A	Internet	Network B
src:	2001:db8::9 + 1579	2001:db8::9 + 1579	2001:db8::9 + 1579
dst:	2001:db8::8 + 2466	2001:db8::8 + 2466	2001:db8::8 + 2466

src:	172.17.1.75	2001:db8::9	fdee:eeee::5951
dst:	10.128.48.62	2001:db8::8	2001:db8:b::28

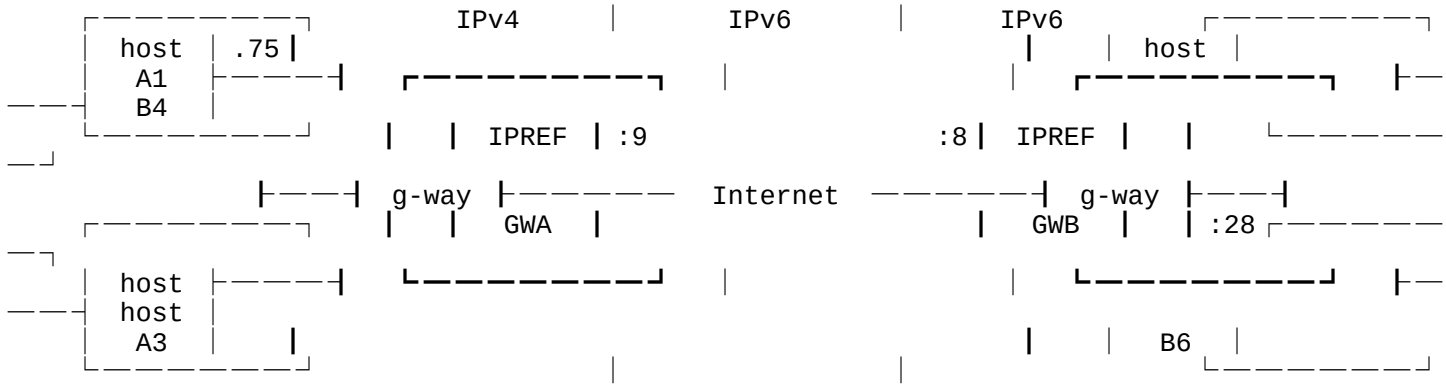


Figure 2: traversing address spaces

For example, in Figure 2, if host A1 wanted to send a packet outside of its own address space to host B6, it would need a source IPREF address for itself and a destination IPREF address for B6.

The destination IPREF address of host B6, presumably a server, would be allocated by the admin of network B where B6 resides. It would normally be advertised in DNS. The source IPREF address of A1, presumably a client, would normally be allocated dynamically by the local IPREF gateway.

As packets travel from source to destination, the following takes place:

* At network A, the IPREF source address is interpreted as a native address of host A1. The destination IPREF address is interpreted as some local private address on the encoding network. The encoding allows host A1 to place native addresses in the packet.

The IPREF gateway will replace both source and destination addresses with their IPREF equivalents. It will also repackage IPv4 packets into IPv6 since the next address space is IPv6.

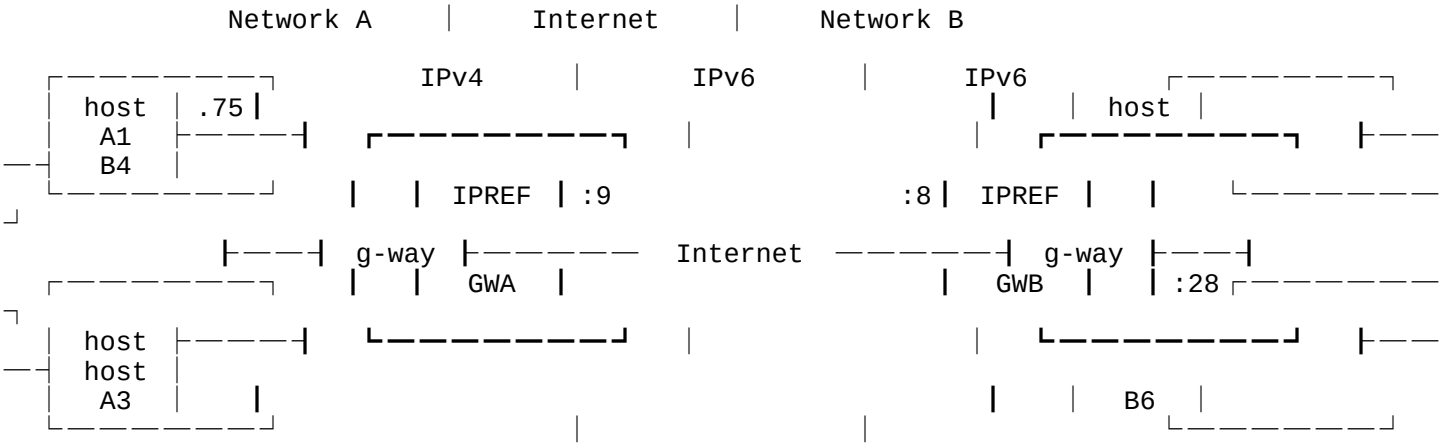
* In the transient address space, the Internet, IPREF addresses are interpreted as well even though the network is unaware of that. In the transient networks only context addresses are used while the references are ignored. This is accomplished by placing context addresses (which must be native) in the respective source and destination fields of the native network protocol. In this way, transient networks, such as the Internet, don't need to know anything about IPREF.

* At network B, the IPREF source address is interpreted as some local private address on the encoding network. The IPREF destination address is interpreted as the native address of host B6. In this way, a packet originating at an IPv4 host A1 reaches an IPv6 host B6.

On the way back, source and destination addresses are reversed and their interpretation follows the same pattern, thus a reply packet originating at an IPv6 host B6 reaches an IPv4 host A1.

In the example, network A may be IPv4 or IPv6, the Internet may be IPv4 or IPv6, and network B may be IPv4 or IPv6. In any combination, a packet traveling through these address spaces would be processed in the same manner and it would reach its destination in the same way.

2.5. Traversing Address Spaces in Detail



Internet ifc: 2001:db8::9	Internet ifc: 2001:db8::8
Local net A: 172.17.1.0/24	Local net B: 2001:db8:b::/64
Encoding net: 10.128.0.0/10	Encoding net: fdee:eeee::/64
Host A1: 172.17.1.75	Host B6: 2001:db8:b:28
DNS A1: (none)	DNS B6: 2001:db8::8 + 2466

Figure 3: traversing address spaces in detail

Let’s say an IPv4 host A1 wants to send a packet to an IPv6 host B6:

1. Host A1 finds out the address of host B6

   Host A1 issues a DNS query for host B6. The query goes through the local resolver. Local resolver receives a response:

   B6 has address: 2001:db8::8 + 2466

   Since local hosts don’t understand IPREF addresses, the resolver asks gateway GWA to allocate an encoded address for it. The gateway responds:

   map: 2001:db8::8 + 2466 = 10.128.48.62

   The resolver returns the encoded result of the DNS query to host A1:

   B6 has address: 10.128.48.62

2. Host A1 sends a packet out to B6

   Host A1 places its own address as source and the address of host B6 returned by the local resolver as destination.

   packet out: | 172.17.1.75 | 10.128.48.62 | payload |

3. Packet arrives at gateway GWA

   packet in: | 172.17.1.75 | 10.128.48.62 | payload |

   The gateway notices that it does not have mapping for the source address, so it creates one on the fly:

   map: 172.17.1.75 = 2001:db8::9 + 1579

   It replaces source address with the just created corresponding IPREF address. It replaces encoded destination address with the original IPREF address returned by the DNS query. It also repackages IPv4 packet into IPv6 since the Internet is IPv6.

   packet out: | 2001:db8::9 + 1579 | 2001:db8::8 + 2466 | payload |

4. Packet arrives at gateway GWB

   packet in: | 2001:db8::9 + 1579 | 2001:db8::8 + 2466 | payload |

The gateway notices that it does not have encoding for the source IPREF address, so it creates one on the fly:

map: 2001:db8::9 + 1579 = fdee:eeee::5951

It replaces source IPREF address with the just created local encoded address. It replaces destination IPREF address with the local address of host B6.

packet out: | fdee:eeee::5951 | 2001:db8:b::28 | payload |

5. Packet arrives at host B6

packet in: | fdee:eeee::5951 | 2001:db8:b::28 | payload |

Host B6 recognizes destination address as its own and consumes the packet. OS passes the packet to an application implied by layer 4.

6. Host B6 sends a reply back to A1

Host B6 gets a reply payload from the application, swaps source and destination addresses, and sends the packet back to A1.

packet out: | 2001:db8:b::28 | fdee:eeee::5951 | payload |

7. Packet arrives at gateway GWB

packet in: | 2001:db8:b::28 | fdee:eeee::5951 | payload |

The gateway replaces local source address with corresponding IPREF address previously allocated and advertised in DNS. It replaces destination local encoded address with the corresponding IPREF address from mapping created earlier.

packet out: | 2001:db8::8 + 2466 | 2001:db8::9 + 1579 | payload |

8. Packet arrives at gateway GWA

packet in: | 2001:db8::8 + 2466 | 2001:db8::9 + 1579 | payload |

The gateway replaces source IPREF address with the corresponding local encoded address from mapping created earlier. It replaces destination IPREF address with the local address of host A1. It also realizes that the local network is IPv4, so it repackages the packet into IPv4.

packet out: | 10.128.48.62 | 172.17.1.75 | payload |

9. Packet arrives at host A1

packet in: | 10.128.48.62 | 172.17.1.75 | payload |

Host A1 recognizes destination address as its own and consumes the packet. OS passes the packet to the application that sent the original packet.

At this point any missing mappings have been allocated. Subsequent packet exchange continues without additional allocations.

2.6. Embedding References in IP Packets

References are network layer entities therefore the most natural place for embedding them would be network layer headers, such as IPv4 options or IPv6 extension headers.

Unfortunately, many Internet Service Providers drop packets with uncommon options and extension headers for various reasons. Even if they don’t, routers tend to put them on a slow processing path resulting in poor performance. For that reason, the most reliable way to embed references is to place them in the payload. One common technique of accomplishing this is tunneling where references could be made a part of the payload.

2.6.1. IPv4 Option

With IPv4, references may be embedded in an IPv4 header option [RFC791]. It would be a new option type. It would contain an octet with option type and option number, an octet with length, and two octets reserved for possible future use while padding to four octet boundary. The source and destination references would then follow.

A new option number would be registered with IANA. Until then, experimental option, 30, could be used.

2.6.2. IPv6 Extension Header

With IPv6, references may be embedded in an IPv6 extension header [RFC8200]. It would be a new header type. It would contain an octet with next header type value, an octet with length, and two octets reserved for possible future use. This would be followed by four octets of padding to 8 octet boundary. Padding octets would not be intended for any future use. The source and destination references would then follow.

A new protocol type would be registered with IANA. Until then, experimental protocol, 254, could be used.

2.6.3. UDP Tunnel

Placing references in a UDP tunnel works for both IPv4 and IPv6. The references would be embedded between the UDP header and the packet payload. In addition, a tunnel information record would be added. The order of items would be as follows:

- UDP header
- Tunnel information record
- Source and destination references
- Packet payload

The tunnel information record would contain the value of TTL and the protocol number plus any bit fields necessary for the tunnel operation.

The tunnel would operate at port 1045. This value would be registered with IANA.

2.7. Name Resolution Support

IPREF gateways provide mapping support for name resolution services such as DNS. When resolving queries on behalf of local hosts, all returned IPREF addresses must be mapped to native addresses of the local network protocol. Typically, a subnet on a private network is dedicated for the purpose. That subnet is used just for encoding, there are no real hosts with these addresses. There may be a need to standardize on how resolvers communicate with gateways to obtain such mappings.

3. DNS with IPREF

IPREF takes full advantage of DNS [RFC1035]. Local networks may publish IPREF addresses of any services hosted on local networks. Standard unmodified DNS servers may be used for the purpose.

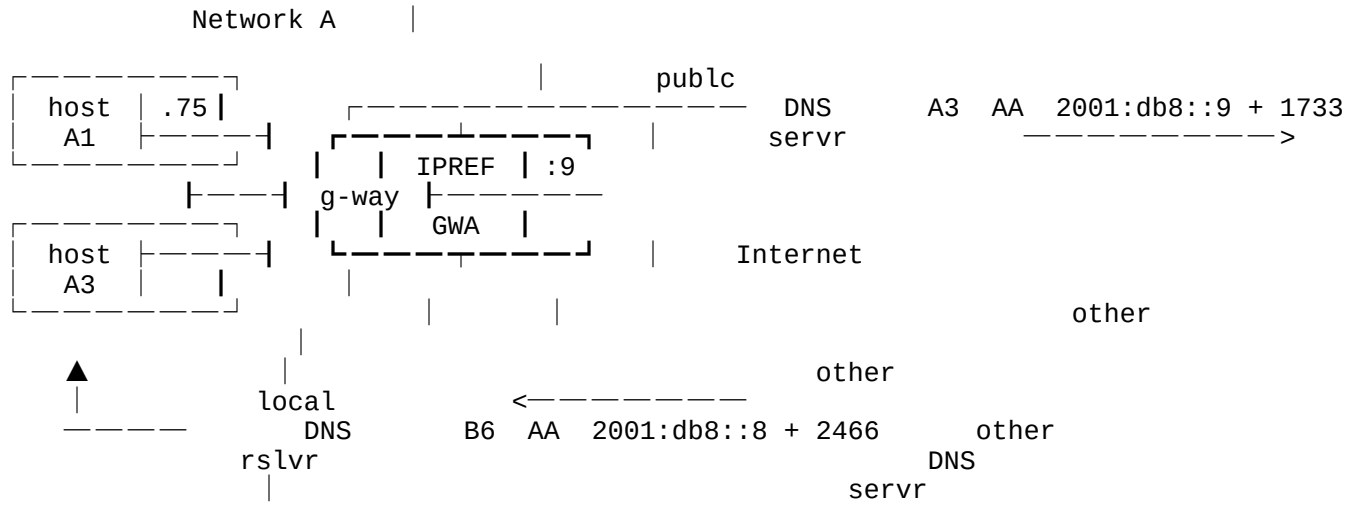


Figure 4: DNS with IPREF

For the resolution of DNS names, a modified recursive resolver is required because IPREF addresses are different from IPv4 and IPv6 addresses. The resolver must recognize them in addition to native IP addresses.

3.1. DNS Records

IPREF addresses are unlike well known IPv4 addresses or IPv6 addresses. These addresses consist of two components which must be considered as a single address entity. It is not possible to separate references from their companion context addresses. For that reason, there is a need for a new resource record type. The new record type is tentatively called AA. This record type has not been registered yet. Until such time, a TXT record may be used instead. The TXT record simply holds a textual representation of an AA record.

Here are some examples of what the records might look like:

```
Host-B4     AA      198.51.100.22 + 400
Host-A5     AA      net-A + 500
Host-A5     AA      2001:db8::abc:11 + 700
Host-A5     TXT     "AA net-A + 500"
```

The IP portion may be provided numerically or as a domain name. The format for the reference would allow unsigned decimal integers as well as unsigned hex integers. Other formats may be defined as well.

3.2. Local Network Resolvers

Local network resolvers must recognize IPREF addresses returned by DNS queries. They must also be capable of encoding them into native IP addresses in consultation with IPREF gateways. This is shown in Figure 4 as a line connecting local DNS resolver with IPREF gateway GWA. Encoding is needed because local hosts do not understand IPREF addresses. Instead, local hosts use encoded equivalents which are then replaced with the actual IPREF addresses by the gateways. There may be a need to standardize on how resolvers communicate with gateways.

3.3. Detecting Published IPREF Addresses

All published IPREF addresses must be communicated to local IPREF gateways so that they can properly map destination addresses of incoming packets. This is reflected in Figure 4 by a line connecting public DNS server with IPREF gateway GWA. There is a number of ways to accomplish this. There may be a need to standardize on how this should be done to allow interoperability between different gateways and different DNS servers.

3.4. DNSSEC compatibility

IPREF address records may be protected by DNSSEC. Packets leaving local networks contain the exact addresses returned from DNS. Internally, local hosts use encoded versions of these addresses which complicates things a little but the gateways replace them with the original IPREF addresses listed in DNS before sending packets out of the local network.

3.5. IPREF Address Literals

DNS is immensely useful and it is considered a necessary component of IP networking. But IP networking does not relay on DNS. It works just fine with IP address literals entered and distributed manually. Similarly, IPREF does not rely on DNS. It is possible to use IPREF address literals in a manner similar to IP. The addresses would still need to be encoded by the gateways. They would be entered manually at gateways first, then their encoded equivalents would be distributed to hosts, possibly as entries in /etc/hosts files.

4. Multiprotocol Internet

4.1. Bridging compatibility divide

IPREF is a general purpose technology that can bridge the divide between incompatible protocols and between unreachable address spaces. This is the network world we live in. There are two public Internets and millions of address spaces. It will stay like this for decades to come. IPREF addresses this situation. In addition to IPv4 and IPv6, new protocols are being developed, such as those presented in various IRTF groups. Some are specialized, some are intended for wider use. Each of them create their own address spaces with a need to exist within wider networking system. IPREF can bridge these new technologies with existing IPv4 and IPv6 Internets allowing them to strive.

4.2. Using IPREF for adoption of IPv6 Internet

Using IPREF for adoption of IPv6 Internet offers significant benefits to both the enterprises and individual users as well as to the Internet service providers and operators.

* only pure IPv6 Internet (no need for IPv4 services or translators)
* only pure IPv4 or pure IPv6 local networks (no dual stacks)
* no need for global IPv4 addresses
* elimination of NAT/NAT6/CGNAT (all hosts reachable, all ports available)
* plays nicely with DNSSEC
* adoption at own pace
* massively scalable
* dramatic speed up of IPv6 Internet adoption
* huge cost savings
* support for emerging new network technologies

The adoption may involve merely a switch to the IPv6 Internet in preparation for the take down of the IPv4 Internet. Or, it may involve deployment of IPv6 local networks. These two tasks are independent. It is possible to keep local network unchanged and only switch to IPv6 Internet. In this case local IPv4 communication will go over IPv6 Internet reaching remote IPv4 or IPv6 destinations. It is also possible to deploy local IP networks while keeping IPv4

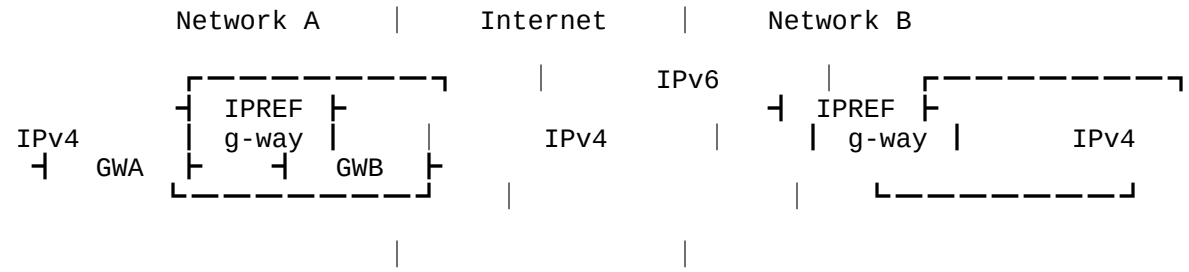
Internet which may be useful in mixed environment. In either case, very little to no coordination is needed with the peers. This alone is a huge advantage of the IPREF technology.

The adoption process takes advantage of IPREF's flexibility to operate in all possible combinations of address spaces:

	network A	Internet	Network B	
1	IPv4	IPv4	IPv4	-- common starting point
2	IPv4	IPv4	IPv6	
3	IPv6	IPv4	IPv4	(same as 2)
4	IPv6	IPv4	IPv6	
5	IPv4	IPv6	IPv4	-- common with IPv6 Internet
6	IPv4	IPv6	IPv6	
7	IPv6	IPv6	IPv4	(same as 6)
8	IPv6	IPv6	IPv6	

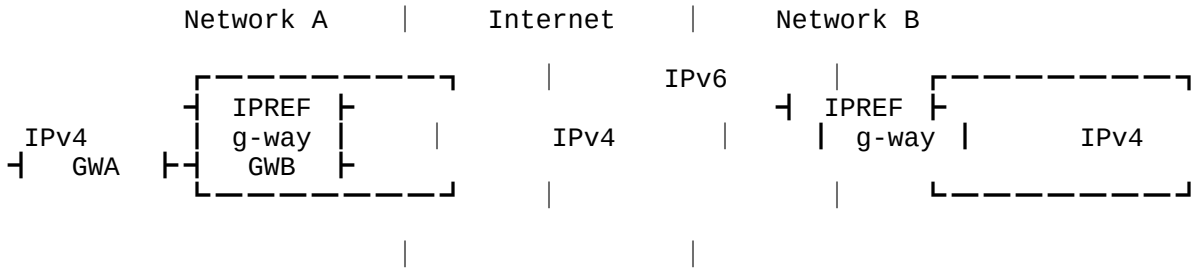
Because the three address spaces - network A, Internet, and network B - are processed independently by IPREF, the adoption process at local networks is decoupled from adoption efforts at peer networks. It is also decoupled from introducing any new protocols at the Internet itself.

1. Common starting point



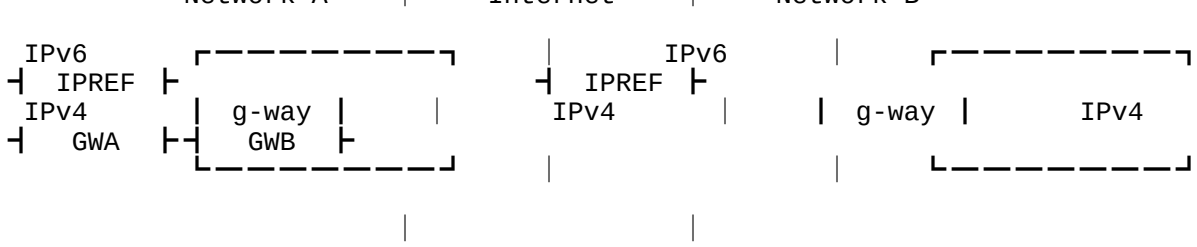
IPREF gateways are installed but not used. All connectivity is pure IPv4.

2. Switching traffic to IPREF



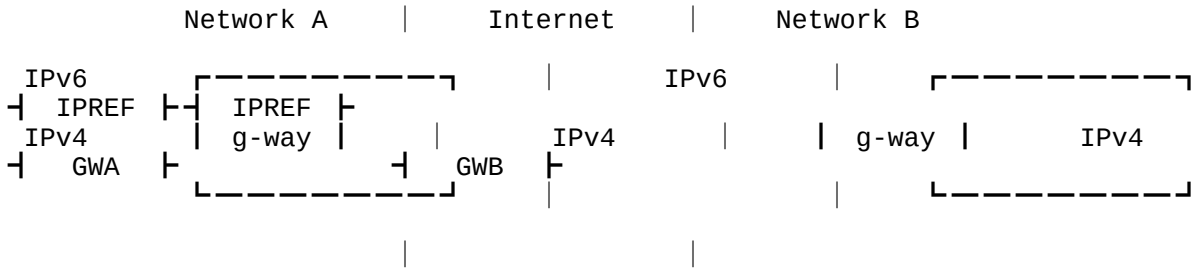
IPREF gateways are configured. References are assigned. Traffic goes through the gateways. All services subject to adoption of the IPv6 Internet are now accessed via IPREF. Traffic between gateways remains over IPv4 Internet until both ends connect to IPv6 Internet. This does not affect deployment of local IPv6 networks.

3. Deploying local IPv6 networks



Local IPv6 networks may be set up without waiting for IPv6 Internet. These are pure IPv6 networks, no dual stacks. Connectivity between local IPv4/IPv6 networks is provided by the same IPREF gateways that connect to remote peers. Clients located on local IPv6 networks can reach servers located on local IPv4 networks or on remote IPv4 or IPv6 networks. Similarly, servers on local IPv6 network can be reached from clients located on local IPv4 networks as well as from remote IPv4 or IPv6 clients over the Internet.

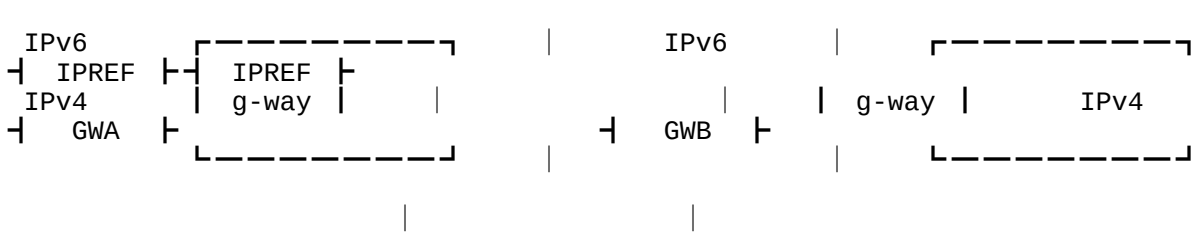
4. Adopting IPv6 Internet



After both ends connect to IPv6 Internet, IPREF addresses may be changed to switch traffic to go over the IPv6 Internet. Local networks are not affected by this change. Peers communicate the same way as if nothing happened. IPv4 Internet remains available but is not used. Most organizations would hold on to it until completely comfortable with the IPv6 Internet.

This concludes adoption of the IPv6 Internet.

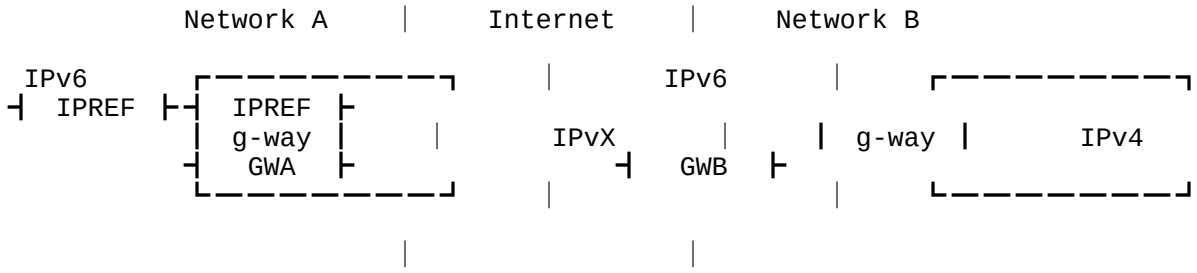
5. Dropping IPv4 Internet



IPv4 Internet may be dropped if all peers connect to IPv6. For a certain period of time, organizations would keep their IPv4 Internet in case some peers are not connected to IPv6 Internet. This process only requires that organizations install IPREF gateways, their local networks remain unchanged.

In the diagram, servers on Network B do not need any global IPv4 addresses, they are reachable from both IPv4 clients and IPv6 clients via IPREF without them. There may be thousands of IPv4 servers on network B and they will all be reachable. Indeed, thanks to NAT traversal capabilities, IPREF extends IPv4 address space to billions of new IPv4 servers, not just clients.

6. Transitioning of the Internet beyond IPv6



IPREF greatly simplifies evolving the Internet past IPv6. In case a new protocol is developed, let's call it IPvX, it may be deployed with relatively little disruption by asking the Internet users to upgrade their edge routers and IPREF gateways. Such an upgrade would typically come down to a routine software download. With most sites already equipped with IPREF gateways, such a massive global upgrade could be performed in a relatively short time and without affecting existing network operations.

5. Related Technologies

IPREF works well with related technologies. It does not create conflicts and it does not attempt to step on their functionality.

* IPv4 - IPREF does not replace IPv4, it is an optional add-on to enhance IPv4 capabilities in the area of address rewriting. It relies on IPv4 in all other functions of a network protocol.

* IPv6 - IPREF does not replace IPv6, it is an optional add-on to enhance IPv6 capabilities in the area of address rewriting. It relies on IPv6 in all other functions of a network protocol.

* NAT - IPREF is an address rewriting technology but operates differently than NAT. It operates exclusively at the network layer. It does not reach to upper layer protocols or lower layer protocols. For example, there is no manipulation of TCP/UDP ports. All layer 4 ports are carried transparently as-is. IPREF does not conflict with NAT. In a common configuration, a NAT gateway connects to the Internet without any changes. IPREF may operate in parallel to NAT on the same gateway, or it may operate on another gateway within the local network 'behind NAT'.

* VPN - IPREF deals mostly with addressing. It does not perform typical functions of a VPN and it does not replace it. It behaves differently. A typical VPN makes hosts of a local network appear as if members of some other local network. Hosts subjected to a VPN are managed by that other private networks. This involves a substantial level of trust between the two private networks.

IPREF does not do any of that. Hosts reachable through IPREF are firmly in their respective private networks and remain managed by their respective administrators. There may be cases where IPREF may be deemed sufficient for a particular purpose but in general IPREF is not a substitute for a VPN. IPREF does not conflict with VPNs. A VPN might use IPREF to establish a VPN connection.

* Firewall - IPREF is not a firewall. While allocation or lack of allocation of references has the effect of blocking or allowing access, blocking policy is best vested with firewalls. IPREF mappers deal with address rewriting, they are not packet filters. There is no conflict between firewalls and IPREF. Firewalls might benefit from IPREF specific features to simplify management and configuration.

6. IANA Considerations

This memo includes no requests to IANA.

7. Security Considerations

This document should not affect the security of the Internet. Documents describing particular pieces of IPREF might. Proper security consideration statements would be included in those documents.

8. References

8.1. Normative References

[RFC791] Postel, J., "Internet Protocol", RFC 791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

8.2. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

Author's Address

Waldemar Augustyn (editor)
Email: waldemar@wdmsys.com