

Network Working Group
Internet-Draft
Intended status: Informational
Expires: June 24, 2009

B. Carpenter
Univ. of Auckland
R. Atkinson
Extreme Networks
H. Flinck
Nokia Siemens Networks
December 21, 2008

Renumbering still needs work
draft-carpenter-renum-needs-work-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 June 24, 2009.

Copyright Notice

Copyright (c) 2008 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.

Abstract

This document reviews the existing mechanisms for site renumbering for both IPv4 and IPv6, and identifies operational issues with those mechanisms. It also summarises current technical proposals for additional mechanisms. Finally there is a gap analysis.

Table of Contents

1.	Introduction	4
2.	Existing Host-related Mechanisms	5
2.1.	DHCP	5
2.2.	IPv6 Stateless Address Auto-configuration	6
2.3.	IPv6 ND Router/Prefix advertisements	7
2.4.	PPP	7
2.5.	DNS configuration	8
3.	Existing Router-related Mechanisms	9
3.1.	Router renumbering	9
4.	Existing Multi-addressing Mechanism for IPv6	9
5.	Operational Issues with Renumbering Today	9
5.1.	Host-related issues	10
5.1.1.	Network layer issues	10
5.1.2.	Transport layer issues	12
5.1.3.	DNS issues	12
5.1.4.	Application layer issues	12
5.2.	Router-related issues	13
5.3.	Other issues	14
5.3.1.	NAT state issues	14
5.3.2.	Mobility issues	14
5.3.3.	Multicast issues	15
5.3.4.	Management issues	15
5.3.5.	Security issues	17
6.	Proposed Mechanisms	18
6.1.	SHIM6	18
6.2.	MANET proposals	18
6.3.	Other IETF work	19
6.4.	Other Proposals	19
7.	Gaps	19
7.1.	Host-related gaps	19
7.2.	Router-related gaps	20
7.3.	Operational gaps	20
7.4.	Other gaps	20
8.	Security Considerations	20
9.	IANA Considerations	21
10.	Acknowledgements	21
11.	Change log	21
12.	Informative References	21
Appendix A.	Embedded IP addresses	25
	Authors' Addresses	26

1. Introduction

[[This is an early draft; some sections are incomplete. The authors invite comments.]]

In early 1996, the IAB published a short RFC entitled "Renumbering Needs Work" [[RFC1900](#)], which the reader is urged to review before continuing. Almost ten years later, the IETF published "Procedures for Renumbering an IPv6 Network without a Flag Day" [[RFC4192](#)]. A few other RFCs have touched on router or host renumbering: [[RFC1916](#)], [[RFC2071](#)], [[RFC2072](#)], [[RFC2874](#)], [[RFC2894](#)], and [[RFC4076](#)].

In fact, since 1996, a number of atomic mechanisms have become available to simplify some aspects of renumbering. The Dynamic Host Configuration Protocol is available for IPv4 [[RFC2131](#)] and IPv6 [[RFC3315](#)]. IPv6 includes Stateless Address Autoconfiguration (SLAAC) [[RFC4862](#)], and this includes Router Advertisements that include options listing the set of active prefixes on a link. PPP [[RFC1661](#)] also allows for automated address assignment for both versions of IP.

Despite these efforts, renumbering, especially for medium to large sites and networks, is widely viewed as an expensive, painful and error-prone process, and is therefore avoided by network managers as much as possible. This has the highly unfortunate consequence that any mechanisms for managing the scaling problems of wide-area (BGP4) routing that require occasional or frequent site renumbering have been consistently dismissed as unacceptable. This document aims to explore the issues behind this problem statement, especially with a view to identifying the gaps and known operational issues.

It is worth noting that for a very large class of users, renumbering is not in fact a problem of any significance. A domestic or small office user whose device operates purely as a client or peer-to-peer node is in practice renumbered at every restart (even if the address assigned is often the same). A user who roams widely with a laptop or pocket device is also renumbered frequently. Such users are not concerned with the survival of very long term application sessions and are in practice indifferent to renumbering. Thus, this document is mainly concerned with issues affecting medium to large sites.

There are numerous reasons why such sites may need to renumber in a planned fashion, including:

- o Change of service provider, or addition of a new service provider, when provider-independent addressing is not an option.
- o A service provider itself has to renumber.
- o Change of site topology (i.e., subnet reorganization).

- o Merger of two site networks into one, or split of one network into two.
- o During IPv6 deployment, change of IPv6 access method (e.g., from tunneled to native).

The most demanding case would be unplanned automatic renumbering, presumably initiated by a site border router, for reasons connected with wide-area routing. There is already a degree of automatic renumbering for some hosts, e.g., IPv6 "privacy" addresses [[RFC4941](#)].

It is certainly to be expected that as the pressure on IPv4 address space intensifies in the next few years, there will be many attempts to consolidate usage of addresses so as to avoid wastage, as part of the "end game" for IPv4, which necessarily requires renumbering of the sites involved. However, strategically, it is more important to implement and deploy techniques for IPv6 renumbering, so that as IPv6 becomes universally deployed, renumbering becomes viewed as a relatively routine event. In particular, some mechanisms being considered to allow indefinite scaling of the wide-area routing system may assume site renumbering to be a straightforward matter.

IP addresses do not have a built-in lifetime. Even when an address is leased for a finite time by DHCP or SLAAC, or when it is derived from a DNS record with a finite time to live, this information is lost once the address has been passed to an upper layer by the socket interface. Thus, a renumbering event is almost certain to be an unpredictable surprise from the point of view of any software using the address. Many of the issues listed below derive from this fact.

2. Existing Host-related Mechanisms

2.1. DHCP

At high level, DHCP [[RFC2131](#)] [[RFC3315](#)] offers similar support for renumbering for both versions of IP. A host requests an address when it starts up, the request may be delivered to a local DHCP server or via a relay to a central server, and if all local policy requirements are met, the server will provide an address with an associated lifetime, and various other network-layer parameters (in particular, the subnet mask and the default router address).

From an operational viewpoint, the interesting aspect is the local policy. Do MAC addresses have to be pre-registered, or can any MAC interface be given an IP address? Will the same IP address be assigned to the same MAC address every time, according to a predefined scheme? (In this case, DHCP is used to mimic manual fixed address assignments.) Alternatively, will the IP addresses in a

subnet be assigned on a first-come, first-served basis?

These policy choices interact strongly with whether the site has what might be called "strong" or "weak" asset management. At the strong extreme, a site has a complete database of all equipment allowed to be connected, certainly containing the MAC address(es) for each host as well as administrative information of various kinds. Such a database can be used to generate configuration files for DHCP, DNS and any access control mechanisms that may be in use. For example, only certain MAC addresses may be allowed to get an IP address on certain subnets. At the weak extreme, a site has no asset management, any MAC address may get a first-come first-served IP address on any subnet, and there is no network layer access control.

A site that uses DHCP can in principle renumber its hosts by reconfiguring DHCP for the new address range. The issues with this are discussed below.

2.2. IPv6 Stateless Address Auto-configuration

SLAAC, although updated recently [[RFC4862](#)], was designed prior to DHCPv6, intended for networks where unattended automatic configuration was preferred. Ignoring the case of an isolated network with no router, which will use link-local addresses indefinitely, SLAAC follows a bootstrap process. Each host first gives itself a link-local address, and then needs to receive a link-local multicast Router Advertisement (RA) [[RFC4861](#)] which tells it the routeable subnet prefix and the address(es) of the default router(s). A node may either wait for the next regular RA, or solicit one by sending a link-local multicast Router Solicitation. Knowing the link prefix from the RA, the node may now configure its own address. There are various methods for this, of which the basic one is to construct a unique 64 bit identifier from the interface's MAC address.

We will not describe here the processes of duplicate address detection, neighbor discovery, and neighbor unreachability discovery. Suffice it to say that they work, once the initial address assignment based on the RA has taken place.

The contents of the RA message are clearly critical to this process and its use during renumbering. An RA can indicate more than one prefix, and more than one router can send RAs on the same link. For each prefix, the RA indicates two lifetimes: "preferred" and "valid". Addresses derived from this prefix must inherit its lifetimes. When the valid lifetime expires, the prefix is dead and the derived address must not be used any more. When the preferred lifetime is expired (or set to zero) the prefix is deprecated, and must not be

used for any new sessions. Thus, setting a finite or zero preferred lifetime is SLAAC's warning that renumbering will occur. SLAAC assumes that the new prefix will be advertised in parallel with the deprecated one, so that new sessions will use addresses configured under the new prefix.

[2.3.](#) IPv6 ND Router/Prefix advertisements

With IPv6, a Router Advertisement not only advertises the availability of an upstream router, but also advertises routing prefix(es) valid on that link (subnetwork). Also, the IPv6 RA message contains a flag indicating whether the host should use DHCPv6 to configure or not. If that flag indicates the host should use DHCPv6, then the host is not supposed to auto-configure itself as outlined in [Section 2.2](#). However, there are some issues in this area, described in [Section 5.1.1](#).

In an environment where a site has more than one upstream link to the outside world, the site might have more than one valid routing prefix. In such cases, typically all valid routing prefixes within a site will have the same prefix length. Also in such cases, it might be desirable for hosts that obtain their addresses using DHCPv6 to learn about the availability of upstream links dynamically, by deducing from periodic IPv6 RA messages which routing prefixes are currently valid. This application seems possible within the IPv6 Neighbour Discovery architecture, but does not appear to be clearly specified anywhere. So at present this approach for hosts to learn about availability of new upstream links or loss of prior upstream links is unlikely to work with currently shipping hosts or routers.

[2.4.](#) PPP

The Point-to-Point Protocol [[RFC1661](#)] includes support for a Network Control Protocol (NCP) for both IPv4 and IPv6.

For IPv4, the NCP is known as IPCP [[RFC1332](#)] and allows explicit negotiation of an IP address for each end. PPP endpoints acquire (during IPCP negotiation) both their own address and the address of their peer, which may be assumed to be the default router if no routing protocol is operating. Renumbering events arise when IPCP negotiation is restarted on an existing link, when the PPP connection is terminated and restarted, or when the point-to-point medium is reconnected. Peers may propose either the local or remote address or require the other peer to do so. Negotiation is complete when both peers are in agreement. In practice, if no routing protocol is used, as in a subscriber/provider environment, then the provider proposes both addresses and requires the subscriber either to accept the connection or abort. Effectively, the subscriber device is

renumbered each time it connects for a new session.

For IPv6, the NCP is IP6CP [[RFC5072](#)] and is used to configure an interface identifier for each end, after which link-local addresses may be created in the normal way. In practice, each side can propose its own identifier and renegotiation is only necessary when there is a collision. Once link-local addresses are assigned and IP6CP is complete, automatic assignment of global scope addresses is performed by the same methods as with multipoint links, i.e., either SLAAC or DHCP6. Again, in a subscriber/provider environment, this allows renumbering per PPP session.

2.5. DNS configuration

A site must provide DNS records for some or all of its hosts, and of course these DNS records must be updated when hosts are renumbered. Most sites will achieve this by maintaining a DNS zone file (or a database from which it can be generated) and loading this file into the site's DNS server(s) whenever it is updated. As a renumbering tool, this is clumsy but effective. Clearly perfect synchronisation between the renumbering of the host and the updating of its A or AAAA record is impossible. The alternative is to use DNS dynamic update [[RFC3007](#)], in which a host informs its own DNS server when it receives a new address.

There are widespread reports that the freely available BIND DNS software (which is what most UNIX hosts use), Microsoft Windows (XP and later), and MacOS X all include support for Secure Dynamic DNS Update. Further, there are credible reports that these implementations are interoperable when configured properly ([[dnsbook](#)] p. 228 and p. 506).

Commonly used commercial DNS and DHCP servers (e.g., MS Exchange) often are deployed with Dynamic DNS also enabled. In some cases, merely enabling both the DNS server and the DHCP server might enable Dynamic DNS also ([[dnsbook](#)] p. 506). So in some cases, sites might have deployed Dynamic DNS without realising it.

The network security community appears to believe that the current DNS Security and Secure Dynamic DNS Update specifications are reasonably secure for most deployment environments [[RFC3007](#)], [[RFC4033](#)], [[RFC4034](#)], [[RFC4035](#)].

The authors note that at the time of this writing there appears to be significantly more momentum towards rapid deployment of DNS Security standards in the global public Internet than previously. See for example <<http://www.dnssec-deployment.org/>> and <<http://www.ntia.doc.gov/DNS/DNSSEC.html>>.

3. Existing Router-related Mechanisms

3.1. Router renumbering

Although DHCP was originally conceived for host configuration, it can also be used for some aspects of router configuration. The DHCPv6 Prefix Delegation options [[RFC3633](#)] are intended for this. For example, DHCPv6 can be used by an ISP to delegate or withdraw a prefix for a customer's router, and this can be cascaded throughout a site to achieve router renumbering. [[Say more.]]

An ICMPv6 extension to allow router renumbering for IPv6 is specified in [[RFC2894](#)], but there appears to be little experience with it. It is not suggested as a useful mechanism by [[RFC4192](#)].

4. Existing Multi-addressing Mechanism for IPv6

IPv6 was designed to support multiple addresses per interface and multiple prefixes per subnet. As described in [[RFC4192](#)], this allows for a phased approach to renumbering (adding the new prefix and addresses before removing the old ones).

As an additional result of the multi-addressing mechanism, a site may choose to use Unique Local Addressing (ULA) [[RFC4193](#)] for all on-site communication, or at least for all communication with on-site servers, while using globally routeable IPv6 addresses for all off-site communications. It would also be possible to use ULAs for all on-site network management purposes, by assigning ULAs to all devices. This would make these on-site activities immune to renumbering of the prefix(es) used for off-site communication. Finally, ULAs can be safely shared with peer sites with which there is a VPN connection, which cannot be done with ambiguous IPv4 addresses [[RFC1918](#)]; such VPNs would not be affected by renumbering.

The IPv6 model also includes "privacy" addresses which are constructed with pseudo-random interface identifiers to conceal actual MAC addresses [[RFC4941](#)]. It is worth noting that IPv6 stacks and client applications need to be agile enough to handle frequent changes in the privacy address, since in a paranoid environment the address lifetime may be rather short.

5. Operational Issues with Renumbering Today

For IPv6, a useful description of practical aspects was drafted in [[I-D.chown-v6ops-renumber-thinkabout](#)], as a complement to [[RFC4192](#)]. As indicated there, a primary requirement is to minimize the

disruption caused by renumbering. This applies at two levels: disruption to site operations in general, and disruption to individual application sessions in progress at the moment of renumbering. In the IPv6 case, the intrinsic ability to overlap usage of the old and new prefixes greatly mitigates disruption to ongoing sessions, as explained in [[RFC4192](#)]. This approach is in practice excluded for IPv4.

[5.1.](#) Host-related issues

[5.1.1.](#) Network layer issues

For IPv4, the vast majority of client systems (PCs and workstations) today use DHCP to obtain their addresses and other network layer parameters. Since DHCP provides for lifetimes after which the address lease expires, it should be possible to devise an operational procedure in which lease expiry coincides with the moment of renumbering (within some margin of error). In this case it would be the DHCP server itself that automatically accomplishes client renumbering, although this would cause a peak of DHCP traffic and therefore would not be instantaneous. DHCPv6 could accomplish a similar result. It has a useful extra feature, a "reconfig-init" message that can be sent to all hosts to inform them to check their DHCPv6 server for an update.

Using such an approach with DHCP will be very different depending whether the site uses strong or weak asset management. With strong asset management, and careful operational planning, the subnet addresses and masks will be updated in the database, and a script will be run to regenerate the DHCP MAC-to-IP address tables and the DNS zone file. DHCP and DNS timers will be set temporarily to small values. The DHCP and DNS servers will be fed the new files, and as soon as the previous DHCP leases and DNS TTLs expire, everything will follow automatically, as far as the host IP layer is concerned. In contrast, with weak asset management, and a casual operational approach, the DHCP table will be reconfigured by hand, the DNS zone file will be edited by hand, and when these configurations are installed, there will be a period of confusion until the old leases and TTLs expire. The DHCPv6 "reconfig-init" message could shorten this confusion to some extent.

DHCP, particularly for IPv4, has acquired a very large number of additional capabilities, with approximately 170 options defined at the time of this writing. Although most of these do not carry IP address information, some do (for example, options 68 through 76 all carry various IP addresses). Thus, renumbering mechanisms involving DHCP have to take into account more than the basic DHCP job of leasing an address to each host.

SLAAC is much less overloaded with options than DHCP; in fact its only extraneous capability is the ability to convey a DNS server address. Using SLAAC to force all hosts on a site to renumber is therefore less complex than DHCP, and the difference between strong and weak asset management is less marked. The principle of synchronising the SLAAC and DNS updates, and of reducing the lease time and TTL, does not change.

We should note a currently unresolved ambiguity in the interaction between DHCPv6 and SLAAC from the host's point of view. RA messages include a 'Managed Configuration' flag known as the M bit, which is supposed to indicate that DHCPv6 is in use. However, it is unspecified whether hosts must interpret this flag rigidly (i.e., only start DHCPv6 if it is set, or if no RAs are received) or whether hosts are allowed or are recommended to start DHCPv6 by default. An added complexity is that DHCPv6 has a 'stateless' mode [[RFC3736](#)] in which SLAAC is used to obtain an address but DHCPv6 is used to obtain other parameters. Another flag in RA messages, the 'Other configuration' or O bit, indicates this.

Until this ambiguous behaviour is clearly resolved by the IETF, operational problems are to be expected. Also, it should be noted that on an isolated LAN, neither RA nor DHCPv6 responses will be received, and the host will remain with only its self-assigned link-local address. One could also have a situation where a multihomed network uses SLAAC for one address prefix and DHCPv6 for another, which would clearly create a risk of inconsistent host behavior and operational confusion.

The SLAAC approach, or DHCP without pre-registered MAC addresses, do not work for servers, printers, or for any other systems that are assigned fixed IP addresses for practical reasons. Manual or script-driven procedures, likely to be site-specific and definitely prone to human error, are needed. If a site has even one host with a fixed, manually configured address, completely automatic host renumbering is very likely to be impossible.

The above assumes the use of typical off-the-shelf hardware and software. There are other environments, often referred to as embedded systems, where DHCP or SLAAC may not be used and even configuration scripts are not an option; for example, fixed IP addresses may be stored in read-only memory, or even set up using DIP switches. Such systems create special problems that no general-purpose solution is likely to address.

5.1.2. Transport layer issues

TCP connections and UDP flows are rigidly bound to a given pair of IP addresses. These are included in the checksum calculation and there is no provision for them to change. It is therefore fundamentally impossible for the flows to survive a renumbering event at either end. From an operational viewpoint, this means that a site that plans to renumber itself is obliged either to follow the overlapped procedure described in [\[RFC4192\]](#), or to announce a site-wide outage for the renumbering process, during which all user sessions will fail. In the case of IPv4, overlapping of the old and new addresses is unlikely to be an option, and in any case is not commonly supported by software. Therefore, absent enhancements to TCP and UDP to enable dynamic endpoint address changes (for example, [\[handley\]](#)), interruptions to TCP and UDP sessions seem inevitable if renumbering occurs at either session endpoint. The same appears to be true of DCCP [\[RFC4340\]](#).

In contrast, SCTP already supports dynamic multi-homing of session end-points, so SCTP sessions ought not be adversely impacted by renumbering the SCTP session end-points [\[RFC4960\]](#), [\[RFC5061\]](#).

5.1.3. DNS issues

The main issue is whether the site in question has a systematic procedure for updating its DNS configuration. If it does, updating the DNS for a renumbering event is essentially a clerical issue that must be coordinated as part of a complete plan, including both forward and reverse mapping. As mentioned in [\[RFC4192\]](#), the DNS TTL will be manipulated to ensure that stale addresses are not cached. However, if the site uses a weak asset management model in which DNS updates are made manually on demand, there will be a substantial period of confusion and errors will be made.

There is anecdotal evidence that many small user sites do not even maintain their own DNS configuration, despite running their own web and email servers. They point to their ISP's resolver, request the ISP to install DNS entries for their servers, but operate internally mainly by IP address. Thus, renumbering for such sites will require administrative coordination between the site and its ISP(s).

5.1.4. Application layer issues

Ideally, we would carry out a renumbering analysis for each application protocol. To some extent, this has been done, in [\[RFC3795\]](#). This found that 34 out of 257 standards-track or experimental application layer RFCs had explicit address dependencies. Although this study was made in the context of IPv4 to

IPv6 transition, it is clear that all these protocols might be sensitive to renumbering. However, the situation is worse, in that there is no way to discover by analysing specifications whether an actual implementation is sensitive to renumbering. Indeed, such analysis may be quite impossible in the case of proprietary applications.

The sensitivity depends on whether the implementation stores IP addresses in such a way that it may refer back to them later, without allowing for the fact that they may no longer be valid. In general, we can assert that any implementation that does not check that an address is valid (e.g., by resolving relevant FQDNs again) each time it opens a new communications session is at risk from renumbering. There are quite egregious breaches of this principle, for example software license systems that depend on the licensed host computer having a particular IP address. Other examples are the use of literal IP addresses in URLs, HTTP cookies, or application proxy configurations. (Also see [Appendix A](#).)

It should be noted that applications are in effect encouraged to be aware of and to store IP addresses by the very nature of the socket API calls `gethostbyname()` and `getaddrinfo()`. It is highly unfortunate that network layer addresses are ever exposed to application sessions, although it may have seemed like the obvious solution when the API was designed 25 years ago. This is made worse by the fact that these functions do not return an address lifetime, so that applications have no way to know when an address is no longer valid. The extension of the same model to cover IPv6 has complicated this problem somewhat. If a model was adopted in which only FQDNs were exposed to applications, and addresses were cached with appropriate lifetimes within the API, most of these problems would disappear. It should be noted that at least the first part of this is already available for some programming languages, notably Java, where only FQDNs need to be handled by application code.

Server applications will likely need to be restarted when the host they contain is renumbered, to ensure that they are listening on a port and socket bound to the new address. In an IPv6 multi-addressed host, server applications need to be able to listen on more than one address simultaneously, in order to cover an overlap during renumbering. Not all server applications are written to do this, and a name-based API as just mentioned would have to provide for this case invisibly to the server code.

[5.2](#). Router-related issues

[RFC2072] gives a detailed review of the operational realities in 1997. A number of the issues discussed in that document were the

result of the relatively recent adoption of classless addressing; those issues can be assumed to have vanished by now. Also, DHCP was a relative newcomer at that time, and can now be assumed to be generally available. Above all, the document underlines that systematic planning and administrative preparation is needed, and that all forms of configuration file and script must be reviewed and updated. Clearly this includes filtering and routing rules (e.g., when peering with BGP, but also with intradomain routing as well). Two particular issues mentioned in [[RFC2072](#)] are:

- o Addresses are cached in routers - routers may need to be restarted.
- o Addresses used by configured tunnels [and today, VPNs] may be overlooked.

In IPv6, if a site wanted to be multi-homed using multiple provider-aggregated (PA) routing prefixes with one prefix per upstream provider, then the interior routers would need a mechanism to learn which upstream providers and prefixes were currently reachable (and valid). In this case their Router Advertisement messages could be updated dynamically to only advertise currently valid routing prefixes to hosts. This would be significantly more complicated if the various provider prefixes were of different lengths or if the site had non-uniform subnet prefix lengths.

[5.3.](#) Other issues

[5.3.1.](#) NAT state issues

When a renumbering event takes place, entries in the state table of any Network Address Translator that happen to contain the affected addresses will become invalid and will eventually time out. Since TCP and UDP sessions are unlikely to survive renumbering anyway, the hosts involved will not be additionally affected. The situation is more complex for multihomed SCTP [[I-D.xie-behave-sctp-nat-cons](#)], depending whether a single or multiple NATs are involved.

A NAT itself may be renumbered and may need a configuration change during a renumbering event.

[5.3.2.](#) Mobility issues

A mobile node using Mobile IP that is not currently in its home network will be affected if either its current care-of address or its home address is renumbered. For IPv6, if the care-of address changes, this will be exactly like moving from one foreign network to another, and Mobile IP will re-bind with its home agent in the normal way. If its home address changes unexpectedly, it can be informed of the new global routing prefix used at the home site through the

Mobile Prefix Solicitation and Mobile Prefix Advertisement ICMPv6 messages [[RFC3775](#)]. The situation is more tricky if the mobile node is detached at the time of the renumbering event, since it will no longer know a valid subnet anycast address for its home agent, leaving it to deduce a valid address on the basis of DNS information.

By contrast to Mobile IPv6, mobile IP for IPv4 does not support prefix solicitation and prefix advertisement messages, limiting its renumbering capability to well scheduled renumbering events when the mobile node is connected to its home agent and managed by the home network administration. Unexpected home network renumbering events when the mobile node is away from its home network and not connected to the home agent are supported only if a relevant AAA system is able to allocate dynamically a home address and home agent for the mobile node.

[5.3.3.](#) Multicast issues

As discussed in [[I-D.chown-v6ops-renumber-thinkabout](#)], IPv6 multicast can be used to help rather than hinder renumbering, for example by using multicast as a discovery protocol (as in IPv6 Neighbor Discovery). On the other hand, the embedding of IPv6 unicast addresses into multicast addresses specified in [[RFC3306](#)] and the embedded-RP (Rendezvous Point) in [[RFC3956](#)] will cause issues when renumbering. Changing the unicast source address of a multicast sender may also be an issue for receivers, especially for source specific multicast (SSM).

[[Need text for the IPv4 case.]]

[5.3.4.](#) Management issues

Today, static IP addresses are routinely embedded in numerous configuration files and network management databases, including MIB modules. Ideally, all these would be generated from a single central asset management database for a given site, but this is far from being universal practice. It should be noted that for IPv6, where multiple prefixes and multiple addresses per host are standard practice, the database and the configuration files will need to allow for this (rather than for a single IPv4 address per host).

Furthermore, because of routing policies and VPNs, a site or network may well embed addresses from other sites or networks in its own configuration data. Thus renumbering will cause a ripple effect of updates for a site and for its neighbours. To the extent that these updates are manual, they will be costly and prone to error. Note that [Section 4](#) suggests that IPv6 ULAs can mitigate this problem, but of course only for VPNs and routes which are suitable for ULAs rather

than globally routeable addresses. The majority of external addresses to be configured will not be ULAs.

See [Appendix A](#) for an extended list of possible static or embedded addresses.

Some address configuration data are relatively easy to find (for example, site firewall rules, ACLs in site border routers, and DNS). Others may be widely dispersed and much harder to find (for example, configurations for building routers, printer addresses configured by individual users, and personal firewall configurations). Some of these will inevitably be found only after the event, when the users concerned encounter a problem.

The overlapped model for IPv6 renumbering, with old and new addresses valid simultaneously, means that planned database and configuration file updates will proceed in two stages - add the new information some time before the renumbering event, and remove the old information some time after. All policy rules must be configured to behave correctly during this process (e.g., preferring the new address as soon as possible). Similarly, monitoring tools must be set up to monitor both old and new during the overlap.

If both IPv4 and IPv6 are renumbered simultaneously in a dual-stack network, additional complications could result, especially with configured IP-in-IP tunnels. This scenario should probably be avoided.

Use of FQDNs rather than raw IP addresses wherever possible in configuration files and databases might reduce/mitigate the potential issues arising from such configuration files or management databases when renumbering is required or otherwise occurs. However, by definition there is then at least one place (i.e., the DNS zone file or the database that it is derived from) where address information is nevertheless inevitable.

It should be noted that the management and administration issues (i.e., tracking down, recording, and updating all instances where addresses are stored rather than looked up dynamically) is the dominant concern of managers considering the renumbering problem. A good example in a corporate context is VPN configuration data held in every employee laptop, for use while on travel and connecting securely from remote locations. How is the IT department able to rapidly update all these devices at exactly the right moment? This has led to a strong managerial preference for continuing the pre-CIDR approach of a provider-independent (PI) prefix, or even for using private addressing and NAT as a matter of choice rather than obligation. The direct cost of renumbering is perceived to exceed

the indirect costs of these alternatives. Additionally, there is a risk element stemming from the complex dependencies of renumbering: it is hard to be fully certain that the renumbering will not cause unforeseen service disruptions, leading to unknown additional costs.

It should be noted that site and network operations managers are often very conservative, and reluctant to change operational procedures that are working reasonably well and are perceived as reasonably secure. They quite logically argue that any change brings with it an intrinsic risk of perturbation and insecurity. Thus, even if procedural changes are recommended that will ultimately reduce the risks and difficulties of renumbering (such as using FQDNs protected by DNSSEC where addresses are used today), these changes may be resisted.

5.3.5. Security issues

For IPv6, addresses are intended to be protected against forgery during neighbor discovery by SEcure Neighbor Discovery (SEND) [[RFC3971](#)]. This appears to be a very useful precaution during dynamic renumbering, to prevent hijacking of the process by an attacker. However, SEND appears to be very difficult to actually deploy and operate. At present it is unclear whether or when SEND might be widely implemented or widely deployed.

Firewall rules will certainly need to be updated, and any other cases where addresses or address prefixes are embedded in security components (access control lists, AAA systems, intrusion detection systems, etc.). If this is not done in advance, legitimate access to resources may be blocked after the renumbering event. If the old rules are not removed promptly, illegitimate access may be possible after the renumbering event. Thus, the security updates will need to be made in two stages (immediately before and immediately after the event).

There will be operational and security issues if an X.509v3 PKI Certificate includes a subjectAltName extension that contains an iPAddress [[RFC5280](#)], and if the corresponding node then undergoes an IP address change without a concurrent update to the node's PKI Certificate. For these reasons, use of the dNSName rather than the iPAddress is recommended for the subjectAltName extension. Any other use of IP addresses in cryptographic material is likely to be similarly troublesome.

If a site is for some reason listed by IP address in a white list (such as a spam white list) this will need to be updated. Conversely, a site which is listed in a black list can escape that list by renumbering itself.

The use of IP addresses instead of FQDNs in configurations is sometimes driven by a perceived security need. Since the name resolution process is typically quite insecure, administrators prefer to use raw IP addresses when the application is security-sensitive (firewalls and VPN are two typical examples). It may be possible to solve this issue over a number of years with DNSSEC (see [Section 2.5](#)).

6. Proposed Mechanisms

6.1. SHIM6

SHIM6, proposed as a host-based multihoming mechanism for IPv6, has the property of switching addresses dynamically in the actual packet stream while presenting a constant upper layer identifier to the transport layer [[I-D.ietf-shim6-proto](#)]. At least in principle, this property could be used during renumbering to alleviate the problem described in [Section 5.1.2](#).

6.2. MANET proposals

The IETF working groups dealing with mobile ad-hoc networks have been working on a number of mechanisms for mobile routers to discover available border routers dynamically, and for those mobile routers to be able to communicate that information to hosts connected to those mobile routers.

Recently, some MANET work has appeared on a Border Router Discovery Protocol that might be useful work towards a more dynamic mechanism for site interior router renumbering [[I-D.boot-autoconf-brdp](#)].

At present, the IETF AutoConf WG [<http://www.ietf.org/html.charters/autoconf-charter.html>] is working on address auto-configuration mechanisms for MANET networks that seem likely to be useful for ordinary non-mobile non-MANET networks also [[I-D.ietf-autoconf-manetarch](#)]. This work is extensively surveyed in [[I-D.bernardos-manet-autoconf-survey](#)] and [[I-D.bernardos-autoconf-solution-space](#)]. Other work in the same area, e.g., [[I-D.templin-autoconf-dhcp](#)], may also be relevant.

MANETs are of course unusual in that they must be able to reconfigure themselves at all times and without notice. Hence the type of hidden static configurations discussed above [Section 5.3.4](#) are simply intolerable in MANETs. Thus, it does not follow that when a consensus is reached on autoconfiguration for MANETs, the solution will also solve the general renumbering problem. However, applying techniques that work for MANETs to conventional networks should

certainly be considered.

6.3. Other IETF work

In the area of management tools, NETCONF [[RFC4741](#)] is suitable for the configuration of any network element or server, so could in principle be used to support secure remote address renumbering.

The DNSOPS WG is working on a nameserver control protocol (NSCP) based on NETCONF that provides means for consistent DNS management including potential host renumbering events [[I-D.dickinson-dnsop-nameserver-control](#)].

6.4. Other Proposals

A proposal has been made to include an address lifetime as an embedded field in IPv6 addresses, with the idea that all prefixes would automatically expire after a certain period and become unrouteable [[scrocker](#)]. While this might be viewed as provocative, it would force the issue by making renumbering compulsory.

7. Gaps

[[This section is very sketchy - ideas wanted.]]

7.1. Host-related gaps

FQDN based network API, and/or FQDN-based transport layer.

Single registry per host for all address-based configuration (/etc/hosts, anyone?), with secure access for site network management.

Do we really need more than DHCP or SLAAC for regular hosts? Do we need an IPv4 equivalent of "reconfig-init"?

We need the IPV6 ND M/O flag debate to be resolved once for all, with default, mandatory and optional behaviors of hosts being fully specified.

The host behavior for upstream link learning suggested in [Section 2.3](#) should be documented.

Multipath survivable transport protocol (or institutionalise some aspects of SHIM6).

The various current discussions of a name-based transport layer or a name-based network API also have potential to alleviate the

application layer issues.

7.2. Router-related gaps

A non-proprietary secure mechanism to allow all address-based configuration to be driven by a central repository for site configuration data. NETCONF might be a suitable basis.

A MANET solution that's solid enough to apply to fully operational small to medium fixed sites for voluntary or involuntary renumbering.

A MANET-style solution that can be applied convincingly to large or very large sites for voluntary renumbering.

Short-term, make [[RFC2894](#)] and [[RFC3633](#)] operable.

7.3. Operational gaps

Deploy DNSSEC.

Deploy multi-prefix usage of IPv6.

Document and encourage systematic site databases and secure configuration protocols for network elements and servers (e.g., NETCONF).

Document functional requirements for site renumbering tools or toolkits.

In general, document renumbering instructions as part of every product manual.

7.4. Other gaps

Secure mechanism for announcing changes of site prefix to peer sites and in public.

For Mobile IP, better mechanism to handle change of home agent address while mobile is disconnected.

8. Security Considerations

Known current issues are discussed in [Section 5.3.5](#). Security issues related to SLAAC are discussed in [[RFC3756](#)].

For future mechanisms to assist and simplify renumbering, care must be taken to ensure that prefix or address changes (especially changes

coming from another site or via public sources such as the DNS) are adequately authenticated at all points. Otherwise, misuse of renumbering mechanisms would become an attractive target for those wishing to divert traffic or to cause major disruption. Whatever authentication method(s) are adopted, key distribution will be an important aspect. Most likely, public key cryptography will be needed to authenticate renumbering announcements passing from one site to another, since one cannot assume a pre-existing trust relationship between such sites.

9. IANA Considerations

This document requires no action by the IANA.

10. Acknowledgements

Significant amounts of text have been adapted from [[I-D.chown-v6ops-renumber-thinkabout](#)]. The authors of that draft have agreed to their text being submitted under the IETF's current copyright provisions.

Useful comments and contributions were made by Stephane Bortzmeyer, Teco Boot, James Woodyatt, Gert Doering, William Herrin, Iljitsch van Beijnum, Darrel Lewis, Fred Baker, Stig Venaas, and others.

This document was produced using the xml2rfc tool [[RFC2629](#)].

11. Change log

[draft-carpenter-renum-needs-work-00](#): original version, 2008-10-23

[draft-carpenter-renum-needs-work-01](#): additional text in many places, started gap analysis, additional author, 2008-12-21

12. Informative References

[I-D.bernardos-autoconf-solution-space]
Bernardos, C., Calderon, M., and H. Moustafa, "Ad-Hoc IP Autoconfiguration Solution Space Analysis", [draft-bernardos-autoconf-solution-space-02](#) (work in progress), November 2008.

[I-D.bernardos-manet-autoconf-survey]
Bernardos, C., Calderon, M., and H. Moustafa, "Survey of

IP address autoconfiguration mechanisms for MANETs", [draft-bernardos-manet-autoconf-survey-04](#) (work in progress), November 2008.

[I-D.boot-autoconf-brdp]

Boot, T. and A. Holtzer, "Border Router Discovery Protocol (BRDP) based Address Autoconfiguration", [draft-boot-autoconf-brdp-01](#) (work in progress), November 2008.

[I-D.chown-v6ops-renumber-thinkabout]

Chown, T., "Things to think about when Renumbering an IPv6 network", [draft-chown-v6ops-renumber-thinkabout-05](#) (work in progress), September 2006.

[I-D.dickinson-dnsop-nameserver-control]

Dickinson, J., Morris, S., and R. Arends, "Design for a Nameserver Control Protocol", [draft-dickinson-dnsop-nameserver-control-00](#) (work in progress), October 2008.

[I-D.ietf-autoconf-manetarch]

Chakeres, I., Macker, J., and T. Clausen, "Mobile Ad hoc Network Architecture", [draft-ietf-autoconf-manetarch-07](#) (work in progress), November 2007.

[I-D.ietf-shim6-proto]

Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", [draft-ietf-shim6-proto-11](#) (work in progress), December 2008.

[I-D.templin-autoconf-dhcp]

Templin, F., "Virtual Enterprise Traversal (VET)", [draft-templin-autoconf-dhcp-24](#) (work in progress), December 2008.

[I-D.xie-behave-sctp-nat-cons]

Xie, Q., Stewart, R., Holdrege, M., and M. Tuexen, "SCTP NAT Traversal Considerations", [draft-xie-behave-sctp-nat-cons-03](#) (work in progress), November 2007.

[RFC1332] McGregor, G., "The PPP Internet Protocol Control Protocol (IPCP)", [RFC 1332](#), May 1992.

[RFC1661] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, [RFC 1661](#), July 1994.

- [RFC1900] Carpenter, B. and Y. Rekhter, "Renumbering Needs Work", [RFC 1900](#), February 1996.
- [RFC1916] Berkowitz, H., Ferguson, P., Leland, W., and P. Nesser, "Enterprise Renumbering: Experience and Information Solicitation", [RFC 1916](#), February 1996.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", [BCP 5](#), [RFC 1918](#), February 1996.
- [RFC2071] Ferguson, P. and H. Berkowitz, "Network Renumbering Overview: Why would I want it and what is it anyway?", [RFC 2071](#), January 1997.
- [RFC2072] Berkowitz, H., "Router Renumbering Guide", [RFC 2072](#), January 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", [RFC 2629](#), June 1999.
- [RFC2874] Crawford, M. and C. Huitema, "DNS Extensions to Support IPv6 Address Aggregation and Renumbering", [RFC 2874](#), July 2000.
- [RFC2894] Crawford, M., "Router Renumbering for IPv6", [RFC 2894](#), August 2000.
- [RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", [RFC 3007](#), November 2000.
- [RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", [RFC 3306](#), August 2002.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", [RFC 3633](#), December 2003.
- [RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", [RFC 3736](#), April 2004.

- [RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", [RFC 3756](#), May 2004.
- [RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", [RFC 3775](#), June 2004.
- [RFC3795] Sofia, R. and P. Nesser, "Survey of IPv4 Addresses in Currently Deployed IETF Application Area Standards Track and Experimental Documents", [RFC 3795](#), June 2004.
- [RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", [RFC 3956](#), November 2004.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", [RFC 3971](#), March 2005.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", [RFC 4033](#), March 2005.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", [RFC 4034](#), March 2005.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", [RFC 4035](#), March 2005.
- [RFC4076] Chown, T., Venaas, S., and A. Vijayabhaskar, "Renumbering Requirements for Stateless Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 4076](#), May 2005.
- [RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPV6 Network without a Flag Day", [RFC 4192](#), September 2005.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", [RFC 4193](#), October 2005.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", [RFC 4340](#), March 2006.
- [RFC4741] Enns, R., "NETCONF Configuration Protocol", [RFC 4741](#), December 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman,

- "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", [RFC 4862](#), September 2007.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", [RFC 4941](#), September 2007.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", [RFC 4960](#), September 2007.
- [RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", [RFC 5061](#), September 2007.
- [RFC5072] S.Varada, Haskin, D., and E. Allen, "IP Version 6 over PPP", [RFC 5072](#), September 2007.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 5280](#), May 2008.
- [dnsbook] Albitz, P. and C. Liu, "DNS and BIND (5th edition)", 2006.
- [handley] Handley, M., Wischik, D., and M. Bagnulo, "Multipath Transport, Resource Pooling, and implications for Routing", 2008, <<http://www.ietf.org/proceedings/08jul/slides/RRG-2.pdf>>.
- [scrocker] Crocker, S., "Renumbering Considered Normal", 2006, <http://www.arin.net/meetings/minutes/ARIN_XVIII/PDF/wednesday/Renumbering_Crocker.pdf>.

[Appendix A](#). Embedded IP addresses

This Appendix lists common places where IP addresses may be embedded. The list was adapted from [[I-D.chown-v6ops-renumber-thinkabout](#)].

Provider based prefix(es)

Names resolved to IP addresses in firewall at startup time

IP addresses in remote firewalls allowing access to remote services

IP-based authentication in remote systems allowing access to
online bibliographic resources
IP address of both tunnel end points for IPv6 in IPv4 tunnel
Hard-coded IP subnet configuration information
IP addresses for static route targets
Blocked SMTP server IP list (spam sources)
Web .htaccess and remote access controls
Apache .Listen. directive on given IP address
Configured multicast rendezvous point
TCP wrapper files
Samba configuration files
DNS resolv.conf on Unix
Any network traffic monitoring tool
NIS/ypbind via the hosts file
Some interface configurations
Unix portmap security masks
NIS security masks
PIM-SM Rendezvous Point address on multicast routers

Authors' Addresses

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland, 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Randall Atkinson
Extreme Networks
PO Box 14129
3306 East NC Highway 54, Suite 100
Research Triangle Park, NC 27709
USA

Email: rja@extremenetworks.com

Hannu Flinck
Nokia Siemens Networks
Linnoitustie 6
Espoo, 02600
Finland

Email: hannu.flinck@nsn.com