IETF Operations Area
Internet-Draft
Intended status: Informational
Expires: July 23, 2010

B. Carpenter
Univ. of Auckland
R. Atkinson
Extreme Networks
H. Flinck
Nokia Siemens Networks

January 19, 2010

Renumbering still needs work
draft-carpenter-renum-needs-work-05

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 identifying possible areas for future work.

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 July 23, 2010.

Copyright Notice

Table of Contents

1. Introduction

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 individual mechanisms have become available to simplify some aspects of renumbering. The Dynamic Host Configuration Protocol (DHCP) is available for IPv4 [RFC2131] and IPv6 [RFC3315]. IPv6 includes Stateless Address Autoconfiguration (SLAAC) [RFC4862], and this includes Router Advertisements (RAs) that include options listing the set of active prefixes on a link. The Point-to-Point Protocol (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. Some would argue that the very design of IP addressing and routing makes automatic renumbering intrinsically impossible. In fact, managers have an economic incentive to avoid having to renumber, and many have resorted to private addressing and NAT as a result. 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. But none of this means that we can duck the problem, because as explained below, renumbering is sometimes unavoidable. 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 might need to renumber in a planned fashion, including:

- Change of service provider, or addition of a new service provider, when provider-independent addressing is not an option.
- A service provider itself has to renumber.
- Change of site topology (i.e., subnet reorganization).
- Merger of two site networks into one, or split of one network into two or more parts.
- 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 might assume site renumbering to be a straightforward matter.

There is work in progress that, if successful, would eliminate some of the motivations for renumbering. In particular, some types of solution to the problem of scalable routing for multihomed sites would likely eliminate both multihoming, and switching to another ISP, as reasons for site renumbering.

Several proposed identifier/locator split schemes provide good examples, including at least Identifier Locator Network Protocol (ILNP) [I-D.rja-ilnp-intro], Locator/ID Separation Protocol (LISP) [I-D.ietf-lisp], and Six/One [I-D.vogt-rrg-six-one] (in alphabetical order). The recent discussion about IPv6 Network Address Translation (IPv6 NAT) provides a separate example[I-D.mrw-behave-nat66]. While remaining highly contentious, this approach, coupled with unique local addresses or a provider-independent address prefix, would appear to eliminate some reasons for renumbering in IPv6. However, even if successful, such solutions will not eliminate all of the reasons for renumbering. This document does not take any position about development or deployment of protocols or technologies that would make long-term renumbering unnecessary, but rather deals with practical cases where partial or complete renumbering is necessary in today's Internet.

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 unavailable to applications 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 application 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 might 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. Some sites require pre-registration of MAC addresses as a security measure, while other sites permit any MAC address to obtain an IP address. Similarly, some sites use DHCP to provide the same IP address to a given MAC address each time (this is sometimes called "Static DHCP"), while other sites do not (this is sometimes called "Dynamic DHCP"), and yet other sites use a combination of these two modes where some devices (e.g. servers, VoIP handsets) have a relatively static IP address that is provisioned via DHCP while other devices (e.g. portable computers) have a different IP address each time they connect to the network. As an example, many US and UK universities require MAC address registration of faculty, staff, and student devices (including hand-held computers connected via wireless).

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 other administrative information of various kinds. Such a database can be used to generate configuration files for DHCP, DNS, and any access control mechanisms that might be in use. For example, only certain MAC addresses might 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.

The IEEE 802.1X standard [IEEE.802-1X], [IEEE.802-1X-REV] specifies a connection mechanism for wired/wireless Ethernet that is often combined with DHCP and other mechanisms to form, in effect, a network login. Using such a network login, the user of a device newly connecting to the network must provide both identity and authentication before being granted access to the network. As part of this process, the network control point will often configure the point of network connection for that specific user with a range of parameters -- such as Virtual LAN (VLAN), Access Control Lists (ACLs), and Quality-of-Service (QoS) profiles. Other forms of Network Login also exist, often using an initial web page for user identification and authentication. The latter approach is commonly used in hotels or cafes.

In principle, a site that uses DHCP can 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 IPv6 processes for Duplicate Address Detection (DAD), Neighbor Discovery (ND), and Neighbor Unreachability Discovery (NUD). 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, or when a provider wishes to force a subscriber to use a specific interface identifier. 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 DHCPv6. 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. An alternative is to use Secure Dynamic DNS 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. So do many home gateways. 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., Windows Server) often are deployed with Secure Dynamic DNS Update also enabled. In some cases, merely enabling both the DNS server and the DHCP server might enable Secure Dynamic DNS Update as an automatic side-effect ([dnsbook] p. 506). So in some cases, sites might have deployed Secure Dynamic DNS Update already, without realising it. An additional enhancement would be for DHCP clients to implement support for the "Client FQDN" option (Option 81).

Since address changes are usually communicated to other sites via the DNS, the latter's security is essential for secure renumbering. The Internet security community believes that the current DNS Security and Secure Dynamic DNS Update specifications are sufficiently secure and has been encouraging DNSsec deployment, [RFC3007], [RFC4033], [RFC4034], [RFC4035].

As of this writing there appears to be significantly more momentum towards rapid deployment of DNS Security standards in the global public Internet than previously. Several country-code Top-Level-Domains (ccTLDs) have already deployed signed TLD root zones (e.g. Sweden's .SE). Several other TLDs are working to deploy signed TLD root zones by published near-term deadlines (e.g. .GOV, .MIL). In fact it is reported that .GOV has been signed operationally since early February 2009. It appears likely that the DNS-wide root zone will be signed in the very near future. See, for example, <http://www.dnssec-deployment.org/> and <http://www.ntia.doc.gov/DNS/DNSSEC.html>.

2.6. Dynamic Service Discovery

The need for hosts to contain pre-configured addresses for servers can be reduced by deploying the Service Location Protocol (SLP). For some common services, such as network printing, SLP can therefore be an important tool for facilitating site renumbering. See [RFC2608], [RFC2610], [RFC3059], [RFC3224], [RFC3421] and [RFC3832].

Multicast DNS (mDNS) and DNS Service Discovery are already widely deployed in BSD, Linux, MacOS X, UNIX, and Windows systems, and are also widely used for both link-local name resolution and for DNS-based dynamic service discovery [I-D.cheshire-dnsext-multicastdns], [I-D.cheshire-dnsext-dns-sd]. In many environments, the combination of mDNS and DNS Service Discovery (e.g. using SRV records [RFC3958]) can be important tools for reducing dependency on configured addresses.

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.

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 mentioned as a useful mechanism by [RFC4192].

[RFC4191] extends IPv6 router advertisements to convey default router preferences and more-specific routes from routers to hosts. This could be used as an additional tool to convey information during

renumbering, but does not appear to be used in practice.

[I-D.ietf-v6ops-ipv6-cpe-router] requires that a customer premises router use DHCPv6 to obtain an address prefix from its upstream ISP, as well as using IPv6 RA messages to establish a default IPv6 route (when IPv6 is in use).

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 might 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]. This means that IPv6 stacks and client applications already need to be agile enough to handle frequent IP address changes (e.g. in the privacy address), since in a privacy-sensitive environment the address lifetime likely will 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, largely because IPv4 lacks a State-Less

Address Auto-Configuration (SLAAC) mechanism.

5.1. Host-related issues

5.1.1. Network layer issues

For IPv4, the vast majority of client systems (PCs, workstations, and hand-held computers) today use DHCP to obtain their addresses and other network layer parameters. DHCP provides for lifetimes after which the address lease expires. So 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 the simplest case, the network administrator just lowers all DHCP address lease lifetimes to a very short value (e.g. a few minutes) far enough before a site-wide change that each node will automatically pick up its new IP address within a few minutes of the renumbering event. 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.

The FORCERENEW extension is defined for DHCP for IPv4 [RFC3203]. This is specifically unicast-only; a DHCP client must discard a multicast FORCERENEW. This could nevertheless be used to trigger the renumbering process, with the DHCP server cycling through all its clients issuing a FORCERENEW to each one. DHCPv6 has a similar feature, i.e., a unicast RECONFIGURE message, that can be sent to each host to inform it to check its DHCPv6 server for an update. These two features do not appear to be widely used for bulk renumbering purposes.

Procedures for using a DHCP approach to site renumbering 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 DHCP FORCERENEW or RECONFIGURE messages 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 SLAAC lease time and DNS 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., may or must 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, since different host operating systems have taken different approaches. This makes it difficult for a site network manager to configure systems in such a way that all hosts boot in a consistent way. Hosts will start SLAAC if so directed by appropriately configured RA messages. However, if one operating system also starts a DHCPv6 client by default, and another one starts it only when it receives the M bit, systematic address management is impeded.

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.

Neither the SLAAC approach, nor DHCP without pre-registered MAC addresses, will work reliably in all cases of systems that are assigned fixed IP addresses for practical reasons. Of course, even systems with static addressing can be configured to use DHCP to

obtain their IP address(es). Such use of "Static DHCP" usually will ease site renumbering when it does become necessary. However, in other cases, 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 might not be used and even configuration scripts might not be an option; for example, fixed IP addresses might 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 at present for the endpoint IP addresses 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 are anecdotal reports 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), unless the DNS servers in use have Secure Dynamic DNS Update enabled. Some commercial DNS service firms include Secure Dynamic DNS Update as part of their DNS service offering.

It should be noted that DNS entries commonly have matching Reverse DNS entries. When a site renumbers, these reverse entries will also need to be updated. Depending on a site's operational arrangements for DNS support, it might or might not be possible to combine forward and reverse DNS updates in a single procedure.

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 might 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 might refer back to them later, without allowing for the fact that they might no longer be valid. In general, we can assert that any implementation is at risk from renumbering if it does not check that an address is valid each time it opens a new communications session. This could be done, for example, by knowing and respecting the relevant DNS time-to-live, or by resolving relevant Fully-Qualified Domain Names (FQDNs) again. A common experience is that even when FQDNs are stored in configuration files, they are resolved only once, when the application starts, and they are cached indefinitely thereafter. This is insufficient. Of course, this does not apply to all application software; for example, several well-known web browsers have short default cache lifetimes.

There are even more 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.)

In contrast, there are also many application suites that actively deal with connectivity failures by retrying with alternative addresses or by repeating DNS lookups. This places a considerable burden of complexity on application developers.

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 many applications use APIs that require the application to see and use lower layer objects, such as network-layer addresses. However, the BSD Sockets API was designed and deployed before the Domain Name System (DNS) was created, so there were few good options at the time. This issue 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. An application using the socket API is obliged to contain explicit logic if it wishes to benefit from the availability of multiple addresses for a given remote host. If a programming model were 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 environments. A common example is Java, where only FQDNs need to be handled by application code in normal circumstances, and no additional application logic is needed to deal with multiple addresses, which are handled by the run-time system. This is highly beneficial for programmers who are not networking experts, and insulates applications software from many aspects of renumbering. It would be helfpul to have similarly abstract, DNS oriented, Networking APIs openly specified and widely available for C and C++.

Some web browsers intentionally violate the DNS TTL with a technique called "DNS Pinning." DNS Pinning limits acceptance of server IP address changes, due to a javascript issue where repeated address changes can be used to induce a browser to scan the inside of a firewalled network and report the results to an outside attacker. Pinning can persist as long as the browser is running, in extreme cases perhaps months at a time. Thus, we can see that security considerations may directly damage the ability of applications to deal with renumbering.

Server applications might 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.

As noted in Section 2.6, the Service Location Protocol (SLP), and multicast DNS with SRV records for service discovery, have been available for some years. For example, many printers deployed in recent years automatically advertise themselves to potential clients via SLP. Many modern client operating systems automatically participate in SLP without requiring users to enable it. These tools appear not to be widely known, although they can be used to reduce the number of places that IP addresses need to be configured.

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:

- Some routers cache IP addresses in some situations. So routers might need to be restarted as a result of site renumbering.
- Addresses might be used by configured tunnels, including VPN tunnels, although at least the Internet Key Exchange (IKE) supports the use of Fully-Qualified Domain Names instead.

On the latter point, the capability to use FQDNs as endpoint names in IPsec VPNs is not new and is standard (see [RFC2407] Section 4.6.2.3 and [RFC4306] Section 3.5). This capability is present in most IPsec VPN implementations. There does seem to be an "educational" or "awareness" issue that many system/network administrators do not realise that it is there and works well as a way to avoid manual modification during renumbering. (Of course, even in this case, a VPN may need to be reconnected after a renumbering event, but most products appear to handle this automatically.)

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 might be renumbered and might need a configuration change during a renumbering event. One of the authors has a NAT-enabled home gateway that obtains its exterior address from the residential Internet service provider by acting as a DHCP Client. That deployment has not suffered operational problems when the ISP uses DHCP to renumber the gateway's exterior IP address. A critical part of that success has been configuring IKE on the home gateway to use a "mailbox name" for the user's identity type (rather than using the exterior IP address of the home gateway) when creating or managing the IP Security Associations.

5.3.2. Mobility issues

A mobile node using Mobile IP that is not currently in its home network will be adversely 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 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.

For both IPv4 and IPv6, changing the unicast source address of a multicast sender might also be an issue for receivers, especially for Source-Specific Multicast (SSM). Applications need to learn the new source addresses, and new multicast trees need to be built

For IPv4 or IPv6 with Any-Source Multicast (ASM), renumbering can be easy. If sources are renumbered, from the routing perspective things behave just as if there are new sources within the same multicast group. There may be application issues though. Changing the RP address is easy when using Bootstrap Router (BSR) [RFC5059] for dynamic RP discovery. BSR is widely used, but it is also common to use static config of RP addresses on routers. In that case router configurations must be updated too.

If any multicast ACLs are configured, they raise the same issue as unicast ACLs mentioned elsewhere.

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 routing prefixes per interface and multiple addresses per interface are standard practice, the database and the configuration files will need to allow for this (rather than for a single address per host, as is normal practice for IPv4).

Furthermore, because of routing policies and VPNs, a site or network

might well embed addresses from other sites or networks in its own configuration data. (It is preferable to embed FQDNs instead, of course, whenever possible.) 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. Synchronizing updates between independent sites can cause unpredictable delays. 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 adresses 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 might 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 renumbering 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.

However, it should be noted that the notion of simultaneously operating multiple prefixes for the same network, although natural for IPv6, is generally not supported by operational tools such as address management software. It also increases the size of IGP routing tables in proportion to the number of prefixes in use. For these reasons, and due to its unfamiliarity to operational staff, the use of multiple prefixes remains rare. Accordingly, the use of ULAs to provide stable on-site addresses even if the off-site prefix changes is also rare.

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 will reduce/mitigate the potential

issues arising from such configuration files or management databases when renumbering is required or otherwise occurs. This is advocated in [RFC1958] (point 4.1). Just as we noted in Section 5.1.4 for applications, this is insufficient in itself; some devices such as routers are known to only resolve FQDNs once, at start-up, and to continue using the resulting addresses indefinitely. This may require routers to be rebooted, when they should instead be resolving the FQDN again after a given timeout.

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 is also possible that some operators may choose to configure addresses rather than names, precisely to avoid a possible circular dependency and the resulting failure modes. This is arguably even advocated in [RFC1958] (point 3.11).

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) form the dominant concern of managers considering the renumbering problem. This has led many sites to continue the pre-CIDR approach of using a provider-independent (PI) prefix. Some sites, including very large corporate networks, combine PI addressing with NAT. Others have adopted private addressing and NAT as a matter of choice rather than obligation. This range of techniques allows for addressing plans that are independent of the ISP(s) in use, and allows a straightforward approach to multihoming. 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.

A relevant 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. Typically, such VPNs are statically configured using numeric IP addresses for endpoints and even with prefix lists for host routing tables. Use of VPN configurations with FQDNs to name fixed endpoints, such as corporate VPN gateways, and with non-address identity types would enable existing IP Security with IKE to avoid address renumbering issues and work well for highly mobile users. This is all possible today with standard IPsec and standard IKE. It just requires VPN software to be configured with names instead of addresses, and thoughtful network administration.

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 might 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. Any automatic renumbering scheme has a potential exposure to such hijacking at the moment that a new address is announced. However, 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 might be blocked after the renumbering event. If the old rules are not removed promptly, illegitimate access might 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 has historically lacked authentication, administrators prefer to use raw IP addresses when the application is

security-sensitive (firewalls and VPN are two typical examples). It might be possible to solve this issue in the next few years with DNSsec (see Section 2.5), now that there is strong DNS Security deployment momentum.

6. Proposed Mechanisms

6.1. SHIM6

SHIM6, proposed as a host-based multihoming mechanism for IPv6, has the property of dynamically switching the addresses used for forwarding the actual packet stream while presenting a constant address as the upper layer identifier for the transport layer [RFC5533]. At least in principle, this property could be used during renumbering to alleviate the problem described in Section 5.1.2.

SHIM6 is an example of a class of solutions with this or a similar property; others are HIP, MOBIKE, Mobile IPv6, SCTP and proposals for multipath TCP.

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 (BRDP)" 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 also seem useful for ordinary non-mobile non-MANET networks [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], might 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 in Section 5.3.4 are simply intolerable in MANETs. Thus, it is possible that when a consensus is reached on autoconfiguration for MANETs, the selected solution(s)

might not be suitable for the more general renumbering problem. However, it is certainly worthwhile to explore applying techniques that work for MANETs to conventional networks also.

6.3. Other IETF work

A DHCPv6 extension has been proposed which could convey alternative routes, in addition to the default router address, to IPv6 hosts [I-D.dec-dhcpv6-route-option]. Other DHCP options are also on the table that may offer information about address prefixes and routing to DHCP or DHCPv6 clients, such as [I-D.ietf-dhc-subnet-alloc] and [I-D.sun-mif-route-config-dhcp6]. It is conceivable that these might be extended as a way of informing hosts dynamically of prefix changes.

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 DNSOP WG has considered a Name Server 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 seeks to identify technology gaps between what is available from existing open specifications and what appears to be needed for site renumbering to be tolerable.

7.1. Host-related gaps

It would be beneficial to expose address lifetimes in the socket API, or any low-level networking API. This would allow applications to avoid using stale addresses.

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 noted in this document. Application

development would be enhanced by the addition of a more abstract network API that supports the C and C++ programming languages. For example, it could use FQDNs and Service Names, rather than SockAddr, IP Address, protocol, and port number. This would be equivalent to similar interfaces already extant for Java programmers.

Moving to a FQDN-based transport layer might enhance the ability to migrate the IP addresses of endpoints for TCP/UDP without having to interrupt a session, or at least in a way that allows a session to restart gracefully.

Having a single registry per host for all address-based configuration (/etc/hosts, anyone?), with secure access for site network management, might be helpful. Ideally, this would be remotely configurable, for example leveraging the IETF's current work on networked-device configuration protocols (NetConf). While there are proprietary versions of this approach, sometimes based on LDAP, a fully standardized approach seems desirable.

Do we really need more than DHCP or SLAAC for regular hosts? Do we need an IPv4 equivalent of SLAAC? How can the use of DHCP FORCERENEW and DHCPv6 RECONFIGURE for bulk renumbering be deployed? FORCERENEW in particular requires DHCP authentication [RFC3118] to be deployed.

The IETF should resolve the 'IPv6 ND M/O flag debate' once and 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.

It would be helpful to have multi-path, survivable, extensions for both UDP and TCP (or institutionalise some aspects of SHIM6).

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 good starting point.

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.

A useful short-term measure would be to ensure that [RFC2894] and [RFC3633] can be used in practice.

7.3. Operational gaps

Since address changes are usually communicated via the DNS, the latter's security is essential for secure renumbering. Thus we should continue existing efforts to deploy DNSSEC globally, including not only signing the DNS root, DNS TLDs, and subsidiary DNS zones, but also widely deploying the already available DNSsec-capable DNS resolvers.

Similarly, we should document and encourage widespread deployment of Secure Dynamic DNS Update both in DNS servers and also in both client and server operating systems. This capability is already widely implemented and widely available, but it is not widely deployed at present.

Deploy multi-prefix usage of IPv6, including ULAs to provide stable internal addresses. In particular, address management tools need to support the multi-prefix model and ULAs.

Ensure that network monitoring systems will function during renumbering, in particular to confirm that renumbering has completed successfully or that some traffic is still using the old prefixes.

Document and encourage systematic site databases and secure configuration protocols for network elements and servers (e.g., NETCONF). The database should store all the information about the network; scripts and tools should derive all configurations from the database. An example of this approach to simplify renumbering is given at [dleroy].

Document functional requirements for site renumbering tools or toolkits.

Document operational procedures useful for site renumbering.

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

Recommend strongly that all IPv6 deployment plans, for all sizes of site or network, should include provision for future renumbering. Renumbering should be planned from day one when the first lines of the configuration of a network or network service are written. Every IPv6 operator should expect to have to renumber the network one day and should plan for this event.

7.4. Other gaps

Define a secure mechanism for announcing changes of site prefix to other sites (for example, those that configure routers or VPNs to point to the site in question).

For Mobile IP, define a 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]. DHCP authentication is defined in [RFC3118].

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. As noted in Section 5.1.4, this may result in defensive techniques such as "DNS pinning" which create difficulty when renumbering.

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], which reflects work carried out during the 6NET project funded by the Information Society Technologies Programme of the European Commission. The authors of that draft have agreed to their text being submitted under the IETF's current copyright provisions. Helpful material about work following on from 6NET was also provided by Olivier Festor of INRIA.

Useful comments and contributions were made (in alphabetical order)

by Jari Arkko, Fred Baker, Olivier Bonaventure, Teco Boot, Stephane Bortzmeyer, Dale Carder, Gert Doering, Ralph Droms, Pasi Eronen, Vijay Gurbani, William Herrin, Cullen Jennings, Eliot Lear, Darrel Lewis, Masataka Ohta, Dan Romascanu, Dave Thaler, Iljitsch van Beijnum, Stig Venaas, Nathan Ward, James Woodyatt, and others.

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

11. Change log [RFC Editor to remove]

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

draft-carpenter-renum-needs-work-02: added discussion of 802.1X, SLP, FORCERENEW, reverse DNS, FQDN-based configuration, DNS pinning, RA and DHCPv6 route preferences; minor edits, additional references, 2009-02-18

draft-carpenter-renum-needs-work-03: updated following IETF74 feedback, expanded discussion of multicast, more discussion of multi-prefix issues, 2009-05-07

draft-carpenter-renum-needs-work-04: updated following IETF Last Call comments, 2009-10-22

draft-carpenter-renum-needs-work-05: updated following IESG comments, 2009-12-19

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-02 (work in progress), July 2009.

[I-D.cheshire-dnsext-dns-sd]
Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-05 (work in progress), September 2008.

[I-D.cheshire-dnsext-multicastdns]
Cheshire, S. and M. Krochmal, "Multicast DNS", draft-cheshire-dnsext-multicastdns-08 (work in progress), September 2009.

[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.dec-dhcpv6-route-option]
Dec, W. and R. Johnson, "DHCPv6 Route Option", draft-dec-dhcpv6-route-option-02 (work in progress), October 2009.

[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-dhc-subnet-alloc]
Johnson, R., Kumarasamy, J., Kinnear, K., and M. Stapp, "Subnet Allocation Option", draft-ietf-dhc-subnet-alloc-10 (work in progress), November 2009.

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-05 (work in progress), September 2009.

[I-D.ietf-v6ops-ipv6-cpe-router]
Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", draft-ietf-v6ops-ipv6-cpe-router-03 (work in

progress), December 2009.

[I-D.mrw-behave-nat66]
Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Address Translation (NAT66)", draft-mrw-behave-nat66-02 (work in progress), March 2009.

[I-D.rja-ilnp-intro]
Atkinson, R., "ILNP Concept of Operations", draft-rja-ilnp-intro-02 (work in progress), December 2008.

[I-D.sun-mif-route-config-dhcp6]
Sun, T. and H. Deng, "Route Configuration by DHCPv6 Option for Hosts with Multiple Interfaces", draft-sun-mif-route-config-dhcp6-01 (work in progress), March 2009.

[I-D.templin-autoconf-dhcp]
Templin, F., "Virtual Enterprise Traversal (VET)", draft-templin-autoconf-dhcp-38 (work in progress), April 2009.

[I-D.vogt-rrg-six-one]
Vogt, C., "Six/One: A Solution for Routing and Addressing in IPv6", draft-vogt-rrg-six-one-02 (work in progress), October 2009.

[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.

[IEEE.802-1X]
Institute of Electrical and Electronics Engineers, "Port-Based Network Access Control: IEEE Standard for Local and Metropolitan Area Networks 802.1X-2004", December 2009.

[IEEE.802-1X-REV]
Institute of Electrical and Electronics Engineers, "802.1X-REV - Revision of 802.1X-2004 - Port Based Network Access Control: IEEE Standard for Local and Metropolitan Area Networks", 2009.

[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.

[RFC1958] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 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.

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[RFC2608] Guttman, E., Perkins, C., Veizades, J., and M. Day, "Service Location Protocol, Version 2", RFC 2608, June 1999.

[RFC2610] Perkins, C. and E. Guttman, "DHCP Options for Service Location Protocol", RFC 2610, June 1999.

[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.

[RFC3059] Guttman, E., "Attribute List Extension for the Service Location Protocol", RFC 3059, February 2001.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3203] T'Joens, Y., Hublet, C., and P. De Schrijver, "DHCP reconfigure extension", RFC 3203, December 2001.

[RFC3224] Guttman, E., "Vendor Extensions for Service Location Protocol, Version 2", RFC 3224, January 2002.

[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.

[RFC3421] Zhao, W., Schulzrinne, H., Guttman, E., Bisdikian, C., and W. Jerome, "Select and Sort Extensions for the Service Location Protocol (SLP)", RFC 3421, November 2002.

[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.

[RFC3832] Zhao, W., Schulzrinne, H., Guttman, E., Bisdikian, C., and W. Jerome, "Remote Service Discovery in the Service Location Protocol (SLP) via DNS SRV", RFC 3832, July 2004.

[RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, November 2004.

[RFC3958] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

[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.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 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.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 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.

[RFC5059] Bhaskar, N., Gall, A., Lingard, J., and S. Venaas, "Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)", RFC 5059, January 2008.

[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, Haskins, 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.

[RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.

[dleroy] Leroy, D. and O. Bonaventure, "Preparing network configurations for IPv6 renumbering", International Journal of Network Management , 2009, <http://inl.info.ucl.ac.be/system/files/dleroy-nem-2009.pdf>.

[dnsbook] Albitz, P. and C. Liu, "DNS and BIND (5th edition)", O'Reilly , 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 might 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
Suite 100, 3306 East NC Highway 54
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