ipv6man
Internet-Draft
Expires: April 6, 2008

B. Dickson
Afilias Canada, Inc

October 4, 2007

A New Method of Constructing Interface Identifiers for Expanded Autoconfiguration in IPv6
draft-dickson-v6man-new-autoconf-00

Status of this Memo

Copyright Notice

Abstract

This Internet Draft discusses a proposed extension to the set of Interface Identifier construction methods for 802 networks. The purpose of this is to allow autoconf RA announcments of prefix length other than 64 bits. It is intended to be fully backward compatible for /64 announcements. Instead of having one Interface Identifier construction method for all purposes, this adds an alternate method which is only used for autoconf, and only if the prefix length is not /64. No other IPv6 methods or protocols require modification. However, without modification, use of prefixes other than /64 likely won't support many IPv6 enhanced functions.

The ultimate goal it providing enough bits between the top level allocation by Regional Internet Registristry (RIR) and the smallest autoconfiguration allocation, to allow both external aggregation by ISPs into one prefix, as well as internal hierarchical aggregation to support a variety of ISP topologies and practices. Current policies are driven from below by the current 64 bit Interface Identifier. Only by relaxing this to 48 bits for such technologies as 802 (Ethernet), does the number of bits available reach a level deemed "sufficient".

Author's Note

This Internet Draft is intended to result in this draft or a related draft(s) being placed on the Standards Track for 6man.
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [5].
Intended Status: Proposed Standard.

Table of Contents

1. Background

"The only problem is scale. If you can solve the scaling problem, nothing else matters." - Paraphrasing Mike O'Dell, founder of UUnet.

IPv6 began as IPng, the "next generation internet protocol". It had many ambitious goals, and has certainly achieved some of them.

However, along the way, some compromises were made, which have been overlooked in terms of impact on the scaling up the deployment, as an overlay on the current generation Internet. This document will attempt to illustrate where those scaling issues exist, how they impact operators, and why they are fundamental - they cannot be worked around, because you cannot work around scaling problems.

2. Description of the Problem: Scaling

The problem being addressed is the intersection between the theoretical design of IPv6, and the practical deployment issues on the hybrid IPv4+IPv6 Internet.

IPv4 space is nearing exhaustion, and is anticipated to be fully allocated by some time in 2010. Network operators now anticipate adopting IPv6 widely, to address the need for additional address space. This in turn is driving demand for initial IPv6 allocations, and for deployment on infrastructure (especially servers) of IPv6.

It is this timeframe, and demand, that means any scaling problems need to be addressed on as short a time-frame as is feasible, so as to minimize the impact to the Internet, and to realize maximum benefit by having as small a deployed base as possible before introducing any changes.

There are several places in the combined infrastructure where scaling issues are of pracitical concern. Management of address assignment is an obvious one, but of low order of importance from the perspective of network traffic.

Routing table size is another, which is impacted by address aggregation and address allocation - two activities that goe hand-in-hand.

This is true both in the context of the global routing table, as well as the internal routing table requirements of individual ISPs.

Routing convergence time, while less obvious, is another function which is scale-sensitive.

And hardware forwarding tables (where they exist) are very scale-limited, expensive, and generally only upgradeable by upgrading the routers containing them. These grow approximately 1:1 with the routing table for any given router.

Currently, the places where the largest potential impact to current and future scaling issues, is the Default-Free Zone (DFZ), where routers are required to hold a complete routing table, since they do not have or use a "default route".

While traditionally this may have been considered to be primarily the region collectively known as "the Tier 1 networks", it now includes any place where routers hold one or more copies of the full routing table. Typical use of a full routing table is where a network operator runs BGP and is multi-homed, and has in addition to multiple

upstream transit providers, a substantial number of routes heard from non-transit (peering) relationships with other network operators.

This now comprises a substantial proportion of large network operators, certainly in the thousands.

In the IPv4 DFZ, the number of prefixes is on the order of 230,000. Also noteworthy is the number of ASNs present in the DFZ, on the order of 30,000. In IPv4, the ratio of prefixes:ASNs is about 8:1. Much of this ratio is a consequence of allocation policies which were necessarily short-term.

Furthermore, the impact to the hardware of each IPv6 prefix, is frequently double that of a single IPv4 prefix.

In order to minimize the impact of adding widescale IPv6 deployment on the DFZ, the objective needs to be providing the means for maximum aggregatability of the IPv6 allocations, over time. The specific objective is, one IPv6 allocation per ASN (since that is the theoretical minumum.)

Aggregation itself cannot be controlled directly or mandated, it should be observed. However, the ability to aggregate is driven directly by the allocation schemes used, and the underlying drivers on consumption within that allocation. Internal allocation and aggregation, within an ISP, is one such driver.

It should be observed that the rate of consumption is fundamentally driven by two metrics: the size of the block from which allocations are made, and the unit size of the smallest allocation.

By showing that even with an optimal allocation scheme, the drivers for consumption are likely to result in either additional allocations per ASN or inability to aggregate allocations within an ASN, we identify the only place where this problem can be addressed - the size of the minimum allocation unit.

2.1. Scaling problem examples

The problem space is a classical triangle:

Efficiency (the packing problem): Efficiency is measured in terms of availability of unused space. Inefficient use is characterized by fragmentation of unused space. Optimal efficiency is achieved if none of the unused block sizes could be merged, regardless of location in the binary tree.

Expansion (the reservation problem): Expansion is the reservation of unused space adjacent to used space. A block expands when it gets merged with unused space adjacent to it. Example: Used block FEC0::2:0/112 merges with unused block FEC0::3:0/112 to become used block FEC0::2:0/111.

Existence (the renumbering problem) As soon as space is allocated, the allocatee becomes a ticking time bomb. It must be presumed

that their network is growing, and at some point will need more space. The recipient will not want to renumber an existing allocation, in order to receive a new allocation.

The more room is reserved for growth, the less is available for new allocations, and the lower the apparent global efficiency. This is a zero-sum game, in a finite space. However, the risk-reward, or rather cost-pain, equation pits the allocatee against the allocator: any improvement in efficiency which requires a recipient to renumber will face vociferous opposition.

2.1.1. Allocation vs Aggregation

Aggregation is the act of combining a number of smaller things into a bigger thing. In the context of prefixes in an internet, aggregation can only occur on bit boundaries, and only when objects being combined are contiguous, with sufficiently similar properties (such as as-path).

Most important is the contiguous property. Consequently, one way to view aggregation is as the reverse of de-aggregation.

Unless the initial allocation and reservation for further allocation are contiguous, no further aggregation can be possible between the two.

And in that sense, the first and subsequent allocations in fact look like a de-aggregation followed by aggregation. Internal use and assignment can similarly be viewed as de-aggregation, with the summarization happening at the border of the entity doing the aggregation (e.g. ASN border router.)

2.1.2. Allocation Techniques

There are a number of general techniques for allocation of address space. Each have pros and cons, related to efficiency, expansion, and renumbering. Variants on each can achieve some compromise in the secondary areas, in addition to the primary benefit of the technique.

Sequential Block This technique breaks the large block into smaller

blocks, and assigns prefixes of a given size all out of one sub-block, in a sequential fashion. Variants make allocations paired with reserverations ajacent in the same block, by effectively increasing the size of allocation itself. While simple to implement, this technique is neither terribly efficient, nor very flexible for growth.

Bisection This technique initially reserves the whole space for the first recipient. Thereafter, each new recipient is assigned space by splitting, or bisecting, the space reserved for one recipient, reserving half for the original recipient and the other half for the new recipient. Growth occurs within a recipient's reserved space. This technique achieves expansion at the cost of efficiency. Under bisection, unused space is *maximally* fragmented. Variants may make allowances in bisection algorithm based on size of initial allocation. Another problem with bisection is, it is non-deterministic, in that it is sensitive to the sequence in which requests are recieved - particularly when balancing new allocation requests against allocation increases due to growth.

Best Fit This techique is guaranteed to be optimal. It uses the smallest unused block big enough to hold the requested allocation, for any new allocation, repeatedly bisecting the selected block (i.e. the smallest block big enough) until the exact fit for the new allocation is recieved. If the smallest is the right size, no partitioning is necessary. This technique guarantees no aggregation of unused space is possible after an allocation, if it wasn't before allocation. It starts with a completely aggregated empty block. Thus, it will always achieve optimal efficiency.

Variants reserve extra space for each allocation, to permit expansion.

For any method of reserving extra space for expansion, in an apples-to-apples comparision, "Best Fit" will have the best efficiency. We are concerned with efficiency at each level, since the more efficient allocation within one block is, the slower the expansion will be within the parent block.

For sake of argument, we will presume optimal allocation at the lowest level, when viewing the impact of growth on multi-level hierarchies of allocations.

2.1.3. Benefits of Aggregation

Aggregation is important in the DFZ, so as to avoid excessive (and expensive) growth which affects all large operators. However, there are benefits outside of the DFZ, which can be achieved by aggregation.

In fact, the scalability of internal networking resources depends on aggregation, and thus the ability to aggregate.

2.1.3.1. Impact of Hierarchical Aggregation

The following example is used to demonstrate the number of prefixes needed, for reaching destinations in an Ingerior Gateway Protocol (IGP) area, of a given size.

The laws governing scalability are identified, and examples used to illustrate how important scaling is to IGPs regardless of specific IGP technology.

The presumption being made is, that allocation is made according to the topology, and that aggregation is done to the maximum degree possible. The point is to demonstrate the benefit of maximum aggregation, which in turn establishes the need for allocation that supports HIERARCHICAL aggregation.

We start with a reasonably big address block, which is allocated out to end networks - a /40, where the end assignments are /64. This gives us 24 bits of allocation ability, or 2^24 networks, approximately 16M. This is a flat network topology, where the allocations are direct and in no way aggregated. The IGP needs to carry all 16M prefixes - and likely will have a lot of difficulty coping with storing those, let alone achieving routing convergence.

Now let us consider a two-level hierarchy. We'll use a couple of examples, and extrapolate the rules on prefix counts in a 2-level aggregation regime.

If we put the delineation point at the /48 boundary, this gives a top-level IGP carrying 8 bits of aggregated prefixes, or 2^8 = 256 prefixes. In addition, each aggregation point, for example an ABR from OSPF, it will need to have all the delegated prefixes for the aggregation it is doing, meaning 2^16 prefixes, or 65k prefixes. That is still a substantial number.

If instead, we put the delineation point between the two levels at

the /52 boundary, this gives both the top level, and the subordinate level, a total of 2^12 of each kind of prefix. The routers in the

IGP doing aggregation would have 2 * 2^12, or 2^13 prefixes, about 8k. That' is quite a bit better.

Now, let us look at a three-level hierarchy. The top level would contain all the top level prefixes. The second level would need all the top level prefixes, plus the second level prefixes. The router bordering level 2 and level3 would also need the third level prefixes.

If each of the boundaries is an 8-bit boundary, the bottom tier routers in the IGP would need 2^8 + 2^8 + 2^8 prefixes, or 768 prefixes. What an inprovement hierarchical assignments make.

The general rule on hierarchical maximum prefix count is, the sum of all values 2^Bi, where Bi is the number of bits used for level "i" of the heirarchy.

Clearly, the greatest benefit is achieved by putting an upper bound on max(Bi), i.e. the section of the hierarchy with the greatest number of bits.

Conversely, the flexibility on creating a topology is limited by the number of PLACES an ISP can do aggregation. The smaller min(Bi) is, the LESS flexible and useful the hierarchical scheme becomes.

Thirdly, the flexibility achieved by the NUMBER of levels in the hierarchy, is the freedom for innovation among ISPs, in terms of the network designs and network topologies that are possible. Fixing, or putting an upper bound on the number of levels of hierarchy, is overly restrictive.

Note well, the first two items, reducing max(Bi) and keeping min(Bi) from collapsing, combine to make the usefulness of a hierarchy dependent on the range of bit sizes per level of the hierarchy. Combined with the last item, this creates an argument for a large number of bits within which to build a hierarchy.

(We explore examples in Appendix A.)

2.1.3.2. Routing Table Size

Hierarchical results in routers typically seeing only prefix information at the same level of the hierarchy as itself, plus summarization prefixes for higher levels of the hierarchy. This drastically reduces the requirements for the size of routing tables within an organization.

2.1.3.3. Routing Stability

Aggregation also limits the impact of routing updates. In a hierarchy of aggregations of prefixes, aggregation typically suppresses reachability regarding more-specific prefixes. This limits the scope of routing flaps, and improves network-wide routing stability. Routing flaps propogate only to the aggregator, and not higher in the hierarchy.

2.1.3.4. Routing Convergence

Thus, we can see that not only external aggregation at the top level, but hierarchical aggregation within a block of addresses, has benefit and is likely to be done by any organization with sufficient resources allocated to it. Routing convergence scales by an order of N*log(N) where N is the number of prefixes. Reducing N by ORDERS of magnitude have profound benefits on speed of convergence, i.e. also orders of magnitude.

The fewer prefixes there are in a routing table, the faster routing can converge. This is especially true for SPF protocols, such as OSPF or ISIS. Convergence time is on the order of Log(N) for N prefixes. The smallest number N ie achieved with routing topologies that implement hierarchical aggregation which mirrors topology. (This is classic OSPF summarization methodology.)

3. Motivation for Change

The growth of the IPv4 space has come at considerable expense, and some of the lessons from the early stages of public IPv4 Internet growth, as has impacted the latter stages of the IPv4 Internet, do not appear to have been heeded.

In particular, the IPv6 design was made prior to extensive experience by operators with the growing pains of IPv4. Where appropriate, consideration of the operator experience can lead to considerable improvement in the scalability and robustness of Internet architecture documents.

Additionally, there are costs which are necessarily borne by the combined IPv4+IPv6 infrastructure - placing an unfortunate, but unavoidable restriction, on deployment of IPv6 - specifically, allocation and aggregation of unicast IPv6 addresses.

The ultimate objective, is to support IPv6 allocations which have sufficient room for growth, so as to ensure that it is highly unlikely that subsequent allocations will be needed for ISPs, at any level, for a substantial length of time (5-10 years).

However, the secondary objective is to make the allocation schemes available at the second level, the ISP, as well suited to the

internal needs of ISPs, so as to prevent harm to individual operators as well as to the global set of network operators.

Only by achieving that objective, will the goal of 1 prefix per ASN be possible. Without achieving this, the routing table in the DFZ may bloat sufficiently to cause significant harm to the operators who operate in the DFZ, ultimately harming the Internet. Doing so without causing internal difficulty for operators who need to aggregate internally, is an essential part of this proposal.

3.1. Current Allocation Techniques

Currently at the top level, IANA has allocated /12 prefixes to the Regional Internet Registries (RIRs). These organizations allocate space to ISPs, as Provider Aggregatable (PA) blocks, and to direct recipients, as Provider Independent (PA) assignments. While there is some variation among the operators, all have the following in common:

Autoconf Block Size: As per [4], currently /64

End User Assignment Size: Typically /56

Initial Assignment, No Paperwork: Normally /48; anthing shorter requires justification.

Minimum Direct Assignment: Also /48 - typically used for Internet Infrastructure use

Initial ISP Allocation Size: Typically /32, with many RIRs allocating much larger blocks depending on ISP requirements (e.g. /19, /20, /21).
Note that these large initial allocations actually *support* the proposition that more bits are needed - otherwise, why would such large allocations be being made?

Reservation for Growth to the ISP: Typically 4 bits

Reservation by the ISP for customer growth: Also typically 4 bits

If the ISP is given a /32, and allocates /48's out of it, and for each /48, reserves the encompasing /44 in its entirety, this means that only 12 bits of range are availabe for both allocation and internal aggregation by the ISP. This is simply too few bits. A three level hierarchy would support at most 4 bits per level (16 prefixes). A two-level hierarchy would provide only either 32 blocks of 128 networks, or 16 blocks of 256 networks. Both of these are too small for even a small ISP's needs on a 10-year basis.

4. Proposed Changes

The proposed changes involve only those elements which affect:

- autoconf <auto>
- IPv6 over anything with hardware address length < 64 bits (e.g. Ethernet <ether>

4.1. Autoconf - Changing the Sense of Interface Identifier

As the current RFCs describe it, an Interface Identifier (II) is an atomic element, the majority of uses of which presume 64-bit addressing. Some RFCs have been structured to accommodate possible II lengths other than 64 bits, but currently no RFCs define such an II instance.

The legacy of the process by which IPv6 was developed, appears to still be contain the hallmarks of the one-size-only II. The fixed-size II is analogous to classful networking in IPv4, only instead of three classes (A,B, and C), there is just one class, the /64.

The current instance of the core document for IPv6, the addressing architecture document, by contrast, describes IPv6 as a 128-bit addressing scheme with no internal structure. Essentially, it is CIDR-128.

In order to take advantage of this, however, it is necessary to foregoe the concept of universal II size, and instead, permit the prefix length (of any IPv6 network) to determine the size of the II.

By reversing the sense of association, from having IPv6 addresses associating "naturally" to a fixed-size Interface Identifier, to having IPv6 addresses associating to a prefix, the very concept of an Interface Identifier changes. Instead of a (fixed-size) constant value, it becomes a deterministically-constructed yet variably-sized entity, for uniquely numbering a given host interface on a given subnet.

Within the context of autoconf, rather than the Interface Identifier being tied tightly to the Link-Local address, it becomes an object which is constructed only after receipt of an RA (with autoconf bit set).

4.1.1. Impact to Existing RFCs

The RFCs for autoconf and Ethernet, plus any other 48-bit MAC layer-2 types (FDDI, etc.). The respective normative references need to be

modified to accommodate this autoconf-specific behaviour, or may need to be updated to reflect better scoping models for aggregation (e.g. RFC 3531).

RFC 4291 [3]IPv6 Addressing Architecture - directly affected

RFC 3587 [7]IPv6 Global Unicast Address Format - directly affected

RFC 3531 [6]A flexible method [...] - examples and appendices may obsolete this RFC, as may some BCP RFCs and/or wiki pages

RFC 2464 [1]Transmission of IPv6 packets over Ethernet Networks - directly affected

RFC 4862 [4]IPv6 Stateless Address Autoconfiguration - this is the main RFC most directly affected

RFC 4941 [2]Privacy Extensions for [...] - presume 64 bit II, but can be easily modified by replacing "top 64" and "bottom 64" references with "top N" and "bottom 128-N", where N is II size in bits.

4.1.1.1. Autoconf

The proscribed behaviour is: if a prefix whose prefix-length is not 64 is received, and which satisfies other conditions (hw_length + prefix_lenth <= 128, and prefix_length > 10), a new Interface Identifier MAY/SHOULD/MUST (language to be decided by consensus and/or AD and/or IESG) be constructed by OR-ing the prefix with the modified hardware address (e.g. EUI-48/MAC-48 with left-padded zeros), the modification being the U and G bits (identical to the Modified EUI-64 bit locations in the first byte).

The previous behaviour for such a prefix would be that autoconf would just fail, silently.

The new (optional) behaviour, if implemented, is that autoconf will now succeed, and be free from duplicates. However, DAD MUST still be done. Wide deployment of implementations of the new behaviour, will make prefix lengths up to /80 useable, which is a necessary step (but not the only step) in solving the scaling problem identified.

Clearly, a prefix other than /64 would not be very useful unless this is adopted, and widely. However if it is, then longer prefixes (especially /80) would then be generally considered usable.

5. Possible (and desired) Impact on Global Allocation Schemes

Without this change, the smallest prefix for which autoconf would work is /64.

With this change, the smallest prefix for which autoconf would work is /80.

DHCPv6 is unaffected by this change. DHCPv6 already supports prefix lengths other than /64. This change brings autoconfiguration in line with DHCPv6.

All of the changes to allocation policies discussed below, cannot be considered unless longer prefixes are useable with autoconfiguration, since autoconfiguration is widely deployed and used.

This conclusion is based on a straw poll, where about 90% of respondents indicated that autoconf was used partly or entirely on prefixes they operated. However, substantial portions of newly deployed networks are likely to use DHCPv6 (e.g. cable company DOCSIS access networks.)

The implication is that an allocation needs to be at least as large as the smallest block usable for autoconf, to be of use to 50-90% of recipients of allocations.

Special note: DHCPv6 supports Prefix Delegation (PD). PD does not presume /64 network size.

PD is, however, usable by routers in RA announcements, which can then be used by autoconfiguration.

This means that without this proposed change, only PD of /64 can be usable by autoconfiguration.

The proposed change would remove that limitation, and any prefix supportable by the hardware address length, would be usable via PD + autoconfiguration.

5.1. Reduction in size of smallest end-user allocation

Under the modified scheme, the smallest anticipated allocation would shrink from /48 or /56, to anywhere from /60 to /76, although the most likely candidates are /60, /64, and /72.

/60, in addition to being the largest of these, would still support a mix of /64 assignments as well as /80 assignments. For example, 2^3

(8) /60's and 2^17 (131k) /80's. The 17 bits available are plenty

for even end-customer internal aggregation/allocation needs, for all but the biggest enterprises.

5.2. Reduction in size of initial allocations to ISPs

An initial allocation of /32, when assignments to customers are /48's and 4 bits are reserved per customer for growth, gives 12 bits of aggregation range.

However, if the customer assignments are /60, an initial allocation of /40 would give 20 bits of aggregation range. This is substantial for both aggregation, and assignment volume and lifetime.

Reserving the whole /32 for the recipient of the /40, is likely to give a substantial time frame within which the customer can grow, without needing to renumber or receive an allocation from a non-continguous netblock.

5.3. Increase in available bits for subnetting

By using /40's, with allocation units of /60, 20 bits are available for subnetting. This is more important for the hierarchical assignment and aggregation within an ISP, than it is for purposes of keeping allocation information organized, although the latter is a beneficial side effect.

5.4. Increase in bits reserved for growth

Rather than a /32 with an extra 4 bits reserved, the new minimum allocation to ISPs could be /40 with 8 bits for growth. That is a factor of 16 more growth, or an additional time-interval to the initial quantity, presuming exponential growth.

5.5. Working Demonstration Code

Example code for current versions of Linux has been produced by the author. The patch needed to support the new method is illustrated below:

5.5.1. Linux Patch example

```
*** /usr/src/linux/net/ipv6/addrconf.c  2007-06-22 13:08:58.000000000
--- addrconf.c  2007-10-04 21:47:44.000000000
***************
*** 1690,1695 ****
--- 1690,1712 ----
                          }
                          goto ok;
                  }
+                 elseif (pinfo->prefix_len > 3 && (pinfo->prefix_len +
+                         8*(dev->addr_len) <= 128)) {
+                     /* II needs to fit in available space */
+                     u8 my_ii[16];
+                     int i;
+                     /* prefix, plus zeros */
+                     memcpy(&addr, &pinfo->prefix, 16);
+                     /* use HW_ADDR from dev as II */
+                     memcpy(my_ii+16-(dev->addr_len),dev->addr,
+                         dev->addr_len);
+                     /* Global/Local bit */
+                     my_ii[16-(dev->addr_len)] ^= 2;
+                     /* guaranteed to be INSIDE the HW_ADDR portion,
+                        and at proper location of EUI-64 */
+                     for(i=0;i<16;i++){ addr.s6_addr[i] |= my_ii[i];}
+                     goto ok;
+                 }
                  if (net_ratelimit())
        printk(KERN_DEBUG "IPv6 addrconf: prefix with wrong length %d\n",
                                  pinfo->prefix_len);
```

6. Security Considerations

This document raises no new security issues.

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgements

The author wishes to acknowledge the helpful guidance of the Working Group chair of what is now 6man, previously ipv6wg, Brian Haberman, and of the Internet Area Director, Jari Arrko.

The author also thanks the contributors on the ipv6 mailing list for pushing him to detail and clarify his concerns, which has resulted in a better Internet Draft. Specific contributors include Thomas Narten, Scott Leibrand, Bob Hinden, Iljitsch van Beijnum, Fred Baker, James Woodyatt, Mark Smith, Brian E. Carpenter, David Conrad, itojun, Christien Huitema, Fred Templin, Michael Dillon, and Ignatios

Souvatzis.

9. References

9.1. Normative References

[1] Crawford, M., "Transmission of IPv6 Packets over Ethernet

Networks", RFC 2464, December 1998.

[2] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[3] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[4] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

9.2. Informative References

[5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[6] Blanchet, M., "A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block", RFC 3531, April 2003.

[7] Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global Unicast Address Format", RFC 3587, August 2003.

Appendix A. Appendix A: Allocation Technique Examples

In the following, we compare tools which do allocations according to either the "best fit" or "bisection" method. We observe the results of the two methods, and examine the details and implications to global allocation policies.

In the following, the allocations are kept in a file, whose structure is described in the comments block. Comments are preserved at the top.

The transaction file is a list of address size requests, and the name to associate to the request.

We illustrate several scenarios, using the same set of allocation requests in different sequence. The resulting allocation files are shown at intermediate steps, so the differences between methods and the sensitivity to sequence of transactions is clearer.

The final allocation files, shows allocations, reservations for growth, and empty space.

Each prefix/length range, has the name assigned to the allocated block, or the empty string indicating unallocated space.

(Bisection uses reserved space, and does not have "unallocated" space, per se.)

Input Files:

Empty allocation file (start from scratch):

```
# File for storing tree of allocations and free blocks
# default base is 10
# default arrangement is flat (vs dotted or colon separated hierarchy)
#
# format of each line is:
# network/[reservation-]length[,customer]
#
# if no [,customer] label exists, the block is available
# if [reservation-] is specified, the following are true:
#       network/length is allocated to customer
#       network/reservation is tentatively reserved for customer,
#         but can be bisected
universe=/6
0/0
```

Transaction file containing sequential requests for new allocations:

```
  # Set of requests (for batch processing of requests for allocations)
  # name /size
  c1 /5
  c2 /6
  c3 /3
  c4 /6
  c5 /4
  c6 /3

  Results for allocation strategy "Bisection":
  universe=/6
  0/3-5,c1
  8/3-4,c5
  16/3-3,c3
  24/3-3,c6
  32/2-6,c2
  48/2-6,c4

  Results for allocation strategy "Best":
  universe=/6
  0/5,c1
  2/6,c2
  3/6,c4
  4/4,c5
  8/3,c3
  16/3,c6
  24/3
  32/1

  Additional allocations:
  c7 /2
  c8 /3
  c9 /3
  c10 /3

  Results for allocation strategy "Bisection":
  Unable to allocate prefix size /2 for c7
  Unable to allocate prefix size /3 for c10
  universe=/6
  0/4-5,c1
  4/4-4,c11
  8/4-4,c5
```

```
12/4-4,c12
16/3-3,c3
24/3-3,c6
32/3-6,c2
40/3-3,c8
48/3-6,c4
```



```
56/3-3,c9

Results for allocation strategy "Best":
universe=/6
0/5,c1
2/6,c2
3/6,c4
4/4,c5
8/3,c3
16/3,c6
24/3,c8
32/2,c7
48/3,c9
56/3,c10
```

We can see that the requests should have used up all of the available space, exactly.

The strategy "Best" succeeded in using up all the space.

The strategy "Bisect" did leave some room for growth for some allocations, but not for others.

"Bisect" ultimately fragmented the space too much for allocations that would otherwise have been able to fit.

Most importantly, the "reserved" space resulting from the "bisection" method is distributed in a non-deterministic manner. This reserves differing amounts of space in a haphazard fashion, which while fair, in the sense of being the result of blind luck, is still unbalanced.

However, it is clear that to ensure both optimized allocation efficiency, and total fairness in growth, that allocations need to be made using the "best" approach, with a fixed (constant) amount of

room for growth, measured in extra "bits" of prefix length.

Due to the particulars of ip6.arpa. reverse delegation, the prefered choice should be on nibble (4-bit) boundaries, with one or two extra nibbles reserved for growth.

It ultimately makes the most sense for growth reservations to be made at each level of inter-organizational allocation (as opposed to internal aggregation points).

RIR->LIR assignments should have growth space appended to assignment lengths, and LIR->customer assignments should also have extra space for growth.

Appendix B. Appendix B: Subnetting Choices by Length

This section enumerates examples of hierarchical subnetting, based on:

Range of bits available This is the number of bits of what has traditionally been called the "subnet mask", between the "network mask", and the "host portion". E.g. If X/16 is subnetted, and the presumed host portion at the LAN level is /64, then the bit range is 64-16 = 48 bits.

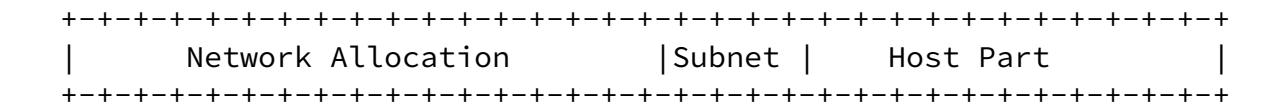
Bits per level (min) We will make some distinctions on the usefulness of different subnetting patterns. For example, nibble boundaries are very convenient, while single-bit subnetting schemes are not likely to be used.

Non VLSM hierarch We presume that at each level of the hierarchy, siblings will have the same subnet mask. E.g. x::12:0/16 subnetted as /17's 12:0/17 and 12:8000/17. If 12:0/17 is subnetted into /19's, then so is 12:8000/17 as /19's.

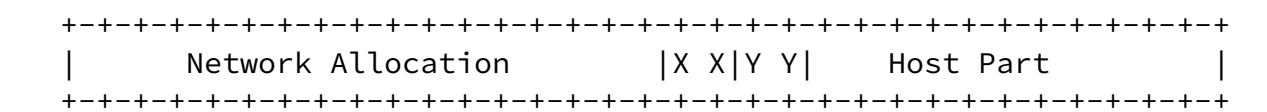
Number of Subnets Total Including varying the number of subnetting steps in the hierarchy, ranging from 0 to the most that will fit based on bits-per-level.

For example, here is a trivial or nearly-trivial subnetting scheme: Range = 4 bits, Bits-per-level = 2 bits Subnet bit-length patterns: 4; 2/2. Total number of subnet patterns: 2.

Detailed layout of the two subnet mask patterns:



Each value of Subnet indicates a different discrete network, and aggregation is presumed only to take place a the level of the Network Allocation.



Where XX is the top level subnet, and YY is the next level subnet, each being 2 bits in size. Network aggregation is presumed to take place at the XX level as well as at the Network Allocatiion.

Range:8, Bits/level: 2 or more 8 2/6 3/5 4/4 5/3 6/2 2/2/4 2/3/3 2/4/2 3/2/3 3/3/2 4/2/2 2/2/2/2 Total: 13

Range: 12, Bits/level: 4 or more 12 4/8 5/7 6/6 7/5 8/4 4/4/4 Total: 7

Range: 12, Bits/level: multiples of 4 12 4/8 8/4 4/4/4 Total: 4

Range: 12, Bits/level: 3 or more Total: 19

Range: 16, Bits/level: 3 or more Total: 88

Range: 16, Bits/level: 4 or more Total: 26

Range: 24, Bits/level 4 or more Total: 345

Range:19, Bits/level 3 or more Total 277

Range: 20, Bits/level: 4 or more Total: 95

Range: 20, Bits/level: multiples of 4 (nibble boundaries only)

Total: 16

Author's Address

Brian Dickson
Afilias Canada, Inc
4141 Yonge St,
Suite 204
North York, ON M2P 2A8
Canada

Email: briand@ca.afilias.info
URI: www.afilias.info

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).