IPv6 IPAM and provisioning management

Aaron Hughes, President & CTO 6connect
aaron@6connet.com
10/18/2012

IPv4 Depletion a Reality

Expected Problems For Service Providers and Enterprises

Number of devices exceeds address pool

NAT

CGN

DS-Lite

Internet performance degradation

Many speak about IPv6 but really we mean technology evolution

	1994 - 2010	Now	2020
Internet of Things Devices	0.5B (2003 – Forrester)	20B	50B – M2M 200B - IoT
Protocols	Appletalk, X.25, IPX/SPX, IPv4	IPv4 – IPv6	IPv6 Only
Name Resolution	Host files, DNS	DNS / DNSSEC	DNSSEC
Services Location	On Premise Servers	Edge -> Core	Cloud
Demarcation	NAT	Grey Area Transition	End to End Reachability
Security	Firewall/NAT	At Risk	IPv6 Integrated Security

How do we manage all this stuff today?

- IPAM: Spreadsheets, memorization, DB+web, open source tools, home grown tools, get-next
- DNS: Bind, vi/vim, Microsoft, Apple, External provider
- DNSSEC
  - Managing keys manually or with open source tools
  - Tracking key rolls
  - Integration with RIR and/or domain registry (DS upload)
  - Do I have authenticated data validation?
- DHCP: Conf files, helper addresses, distributed pools
- Assets: Salesforce? Microsoft? Shared Spredsheet?
- Provisioning: Policy, scripts?

Why is this a challenge now?

- Time is money
- Critical not make human errors in provisioning process
- Fast, Accurate provisioning means booking revenue faster
- Math is hard, let's go shopping
  - 32 bits 2^32 = ~4 billion
  - 128 bits 2^128 = ~340 undecillion
  - Hex vs. decimal
  - Memorable vs. challenging
  - SLAAC vs. DHCP
  - 4 octets vs. 8 biglets
  - 8 bit dec (0 – 255), 16 bit dec (0 – 65536)
- Even a simple table is scary

IPv6 table with nibbles

```
  /16   /32   /48   /64   /80   /96   /112
    v     v     v     v     v     v     v
2001:aaaa:bbbb:cccc:dddd:eeee:ffff:1111
```

Prefix	/48 count	/56 count	Number of /64 Subnets	Number of Hosts
/64			1	18,446,744,073,709,551,616 (2^64) (quintillion)
/63			2	36,893,488,147,419,103,232
/62			4	73,786,976,294,838,206,464
/61			8	147,573,952,589,676,412,928
/60			16	295,147,905,179,352,825,856
/59			32	590,295,810,358,705,651,712
/58			64	1,180,591,620,717,411,303,424 (sextillion)
/57			128	2,361,183,241,434,822,606,848
/56		1	256	4,722,366,482,869,645,213,696 (2^72)
/55		2	512	9,444,732,965,739,290,427,392
/54		4	1,024	18,889,465,931,478,580,854,784
/53		8	2,048	37,778,931,862,957,161,709,568
/52		16	4,096	75,557,863,725,914,323,419,136
/51		32	8,192	151,115,727,451,828,646,838,272
/50		64	16,384	302,231,454,903,657,293,676,544
/49		128	32,768	604,462,909,807,314,587,353,088
/48	1	256	65,536	1,208,925,819,614,629,174,706,176 (2^80) (septillion)
/47	2	512	131,072	2,417,851,639,229,258,349,412,352
/46	4	1,024	262,144	4,835,703,278,458,516,698,824,704
/45	8	2,048	524,288	9,671,406,556,917,033,397,649,408
/44	16	4,096	1,048,576	19,342,813,113,834,066,795,298,816
/43	32	8,192	2,097,152	38,685,626,227,668,133,590,597,632
/42	64	16,384	4,194,304	77,371,252,455,336,267,181,195,264
/41	128	32,768	8,388,608	154,742,504,910,672,534,362,390,528
/40	256	65,536	16,777,216	309,485,009,821,345,068,724,781,056
/39	512	131,072	33,554,432	618,970,019,642,690,137,449,562,112
/38	1,024	262,144	67,108,864	1,237,940,039,285,380,274,899,124,224 (octillion)
/37	2,048	524,288	134,217,728	2,475,880,078,570,760,549,798,248,448
/36	4,096	1,048,576	268,435,456	4,951,760,157,141,521,099,596,496,896
/35	8,192	2,097,152	536,870,912	9,903,520,314,283,042,199,192,993,792
/34	16,384	4,194,304	1,073,741,824	19,807,040,628,566,084,398,385,987,584
/33	32,768	8,388,608	2,147,483,648	39,614,081,257,132,168,796,771,975,168

IPv6 with nibbles cont.

/32	65,536	16,777,216	4,294,967,296	79,228,162,514,264,337,593,543,950,336 (2^96)
/31	131,072	33,554,432	8,589,934,592	158,456,325,028,528,675,187,087,900,672
/30	262,144	67,108,864	17,179,869,184	316,912,650,057,057,350,374,175,801,344
/29	524,288	134,217,728	34,359,738,368	633,825,300,114,114,700,748,351,602,688
/28	1,048,576	268,435,456	68,719,476,736	1,267,650,600,228,229,401,496,703,205,376 (nonillion)
/27	2,097,152	536,870,912	137,438,953,472	2,535,301,200,456,458,802,993,406,410,752
/26	4,194,304	1,073,741,824	274,877,906,944	5,070,602,400,912,917,605,986,812,821,504
/25	8,388,608	2,147,483,648	549,755,813,888	10,141,204,801,825,835,211,973,625,643,008
/24	16,777,216	4,294,967,296	1,099,511,627,776	20,282,409,603,651,670,423,947,251,286,016
/23	33,554,432	8,589,934,592	2,199,023,255,552	40,564,819,207,303,340,847,894,502,572,032
/22	67,108,864	17,179,869,184	4,398,046,511,104	81,129,638,414,606,681,695,789,005,144,064
/21	134,217,728	34,359,738,368	8,796,093,022,208	162,259,276,829,213,363,391,578,010,288,128
/20	268,435,456	68,719,476,736	17,592,186,044,416	324,518,553,658,426,726,783,156,020,576,256
/19	536,870,912	137,438,953,472	35,184,372,088,832	649,037,107,316,853,453,566,312,041,152,512
/18	1,073,741,824	274,877,906,944	70,368,744,177,664	1,298,074,214,633,706,907,132,624,082,305,024 (decillion)
/17	2,147,483,648	549,755,813,888	140,737,488,355,328	2,596,148,429,267,413,814,265,248,164,610,048
/16	4,294,967,296	1,099,511,627,776	281,474,976,710,656	5,192,296,858,534,827,628,530,496,329,220,096
/15	8,589,934,592	2,199,023,255,552	562,949,953,421,312	1,038,459,371,706,965,525,706,099,265,844,0192
/14	17,179,869,184	4,398,046,511,104	1,125,899,906,842,624	207,691,874,341,393,105,141,219,853,168,80,384
/13	34,359,738,368	8,796,093,022,208	2,251,799,813,685,248	41,538,374,868,278,621,028,243,970,633,760,768
/12	68,719,476,736	17,592,186,044,416	4,503,599,627,370,496	83,076,749,736,557,242,056,487,941,267,521,536
/11	137,438,953,472	35,184,372,088,832	9,007,199,254,740,992	166,153,499,473,114,484,112,975,882,535,043,072
/10	274,877,906,944	70,368,744,177,664	18,014,398,509,481,984	332,306,998,946,228,968,225,951,765,070,086,144
/9	549,755,813,888	140,737,488,355,328	36,028,797,018,963,968	664,613,997,892,457,936,451,903,530,140,172,288
/8	1,099,511,627,776	281,474,976,710,656	72,057,594,037,927,936	1,329,227,995,784,915,872,903,807,060,280,344,576 (undecillion)
/7	2,199,023,255,552	562,949,953,421,312	144,115,188,075,855,872	2,658,455,991,569,831,745,807,614,120,560,689,152
/6	4,398,046,511,104	1,125,899,906,842,624	288,230,376,151,711,744	5,316,911,983,139,663,491,615,228,241,121,378,304
/5	8,796,093,022,208	2,251,799,813,685,248	576,460,752,303,423,488	10,633,823,966,279,326,983,230,456,482,242,756,608
/4	17,592,186,044,416	4,503,599,627,370,496	1,152,921,504,606,846,976	21,267,647,932,558,653,966,460,912,964,485,513,216
/3	35,184,372,088,832	9,007,199,254,740,992	2,305,843,009,213,693,952	42,535,295,865,117,307,932,921,825,928,971,026,432
/2	70,368,744,177,664	18,014,398,509,481,984	4,611,686,018,427,387,904	85,070,591,730,234,615,865,843,651,857,942,052,864
/1	140,737,488,355,328	36,028,797,018,963,968	9,223,372,036,854,775,808	170,141,183,460,469,231,731,687,303,715,884,105,728

Early adoption is painful at times

- Every person in this room can empathize with getting support from a vendor at your implementation speed
- Consulting related to Edge to Core, Dual Stacking, DNSSEC, Application Cataloging, High Availability, Virtualization, etc.
- MANY gaps identified
  - Data Center as a unit doesn’t exist yet
  - Design, Implementation, Operations tools way behind
  - This presentation focus on provisioning

Identifying the gap

- The search results
  - There are very few provisioning systems
    - There are a lot of DDI systems (not the same)
  - Almost none that support IPv6 or DNSSEC
  - None integrated with RIR RESTFul APIs (ARIN/RIPE)
  - None of them supported templatization
  - Open source tools are all behind (Ipplan and the like)
  - None took a holistic approach
  - None managed discovery
  - They were all slow
  - Most proprietary
  - None of them were really thinking about scale or speed

The choices

- Beat potential vendors into solving the problem
- Write my own

The choices

- ~~Beat potential vendors into solving the problem~~

- Write my own

Goals

- Good planning for the long term
- Follows my business logic not forced into tools logic
- Planner/designer/policy maker can create complex policy
- Easy for Operator / Resource Requestor / IP Analyst / IT
- No real need to understand subnetting
  - RIR/Region/Purpose/Size
  - DNSSEC
  - Templatized “New Cable Customer” / “New Campus” / “New PoP”
- Forced to stay within policy
- Easy reporting on utilization / run-out
- Integration to RIR (or LIR)
- Views / Management up and down

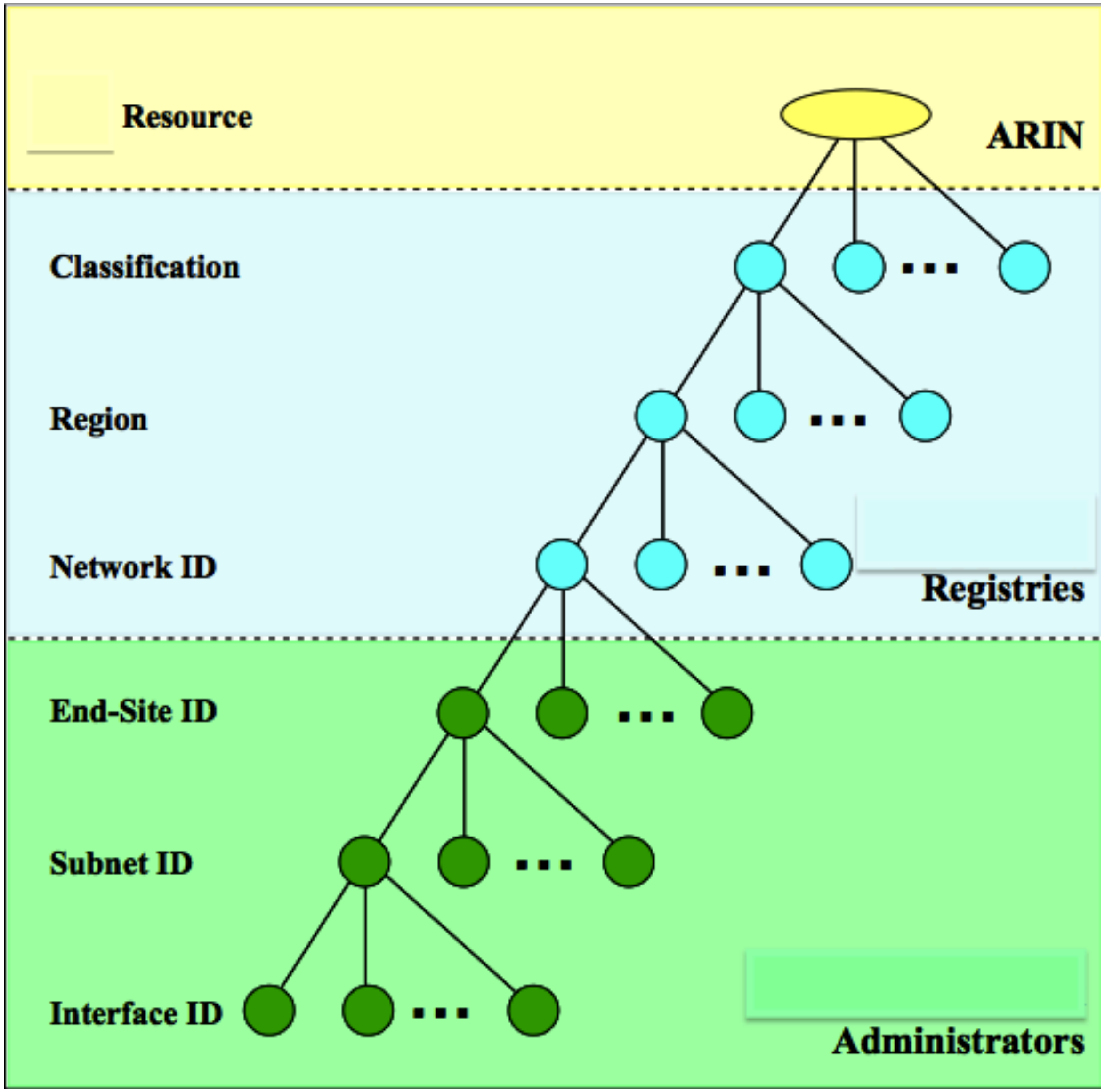
A little more on provisioning problems…

- Before we get into the solution, let's define the problem in a bit more details

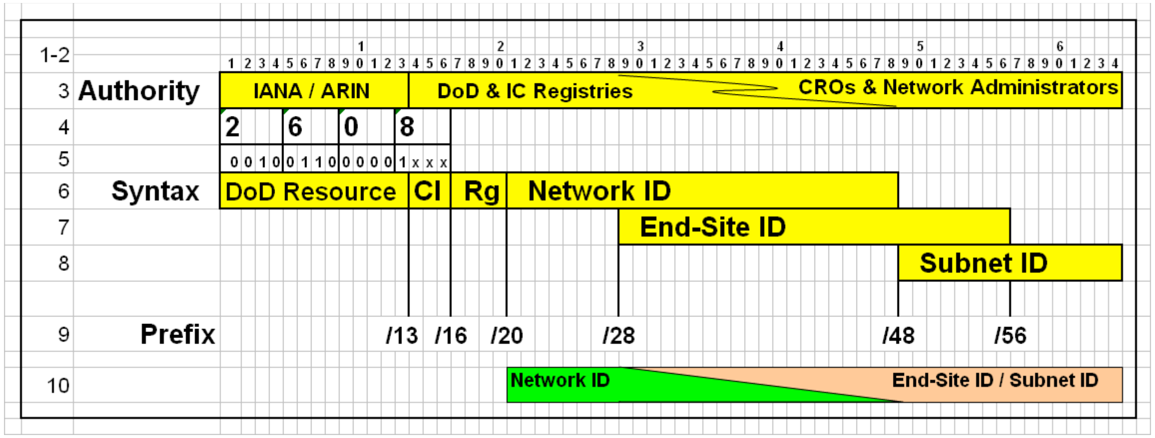
Planning has changed

- Historical 80% utilization -> justification -> dip back in the pool did not allow for long term planning
  - Add new tie down
  - Utilize
  - Rinse Repeat
- Planning for 10-20 years is far harder than 2 years
- Reactive vs. Proactive

Planning: Using bits for purpose (example)



Another example



Name	Syntax			
	Prefix	Length (Bits)	Location (Bits)	Context *
DoD Resource	/13	13	1 – 13	260[1xxx]::/13
Classification	/16	3	14 – 16	260[1XXX]::/16
Region	/20	4	17 – 20	260[1xxx]:Xxxx::/20
Network ID	Variable length (see below)			
	/28	8	21 – 28	260[1xxx]:xXXx::/28 (Q1, Q2)
	to			
	/48	28	21 – 48	260[1xxx]:xXXX:XXXX::/48 (Q1, Q2, Q3)

Interesting example matching IPv4

2607:FFFF::/32	- Some Company Allocation from ARIN
2607:FFFF:0500::/40	- Datacenter
2607:FFFF:0500::/44	- ABC Network Environment
2607:FFFF:0500::/48	- First /48 of the 16 /48s available in this scheme.
2607:FFFF:0500:0000::/64	- VLAN 100 - For GD internal, no customer IPs
2607:FFFF:0500:0011::/64	- VLAN 201 - Customer VLAN
2607:FFFF:0500:0012::/64	- VLAN 202 - Customer VLAN
2607:FFFF:0500:0013::/64	- VLAN 203 - Customer VLAN
2607:FFFF:0500:0014::/64	- VLAN 204 - Customer VLAN
| |	
2607:FFFF:0500:001F::/64	- VLAN 215 - Customer VLAN

From each /64 listed above, we will pull out a single /118 for actual use. From these 1024 addresses, we reserve the first 256 for network use and the rest are available to give out to the customers and will therefore be added to the ipv6-primary-pool. Please note that the /64 is really only defined for clarity in this case. The /118 is the actual VLAN.

Here is an example for VLAN 202:

2607:FFFF:0500:0012::/118	- VLAN 202
2607:FFFF:0500:0012::/128	- First reserved address
2607:FFFF:0500:0012::00FF/128	- Last reserved address
2607:FFFF:0500:0012::0100/128	- First useable address
2607:FFFF:0500:0012::03FF/128	- Last useable address (0100-03FF added to ipv6-primary-pool)

- Is this a good or a bad thing and why?

6connect

Other scary things…

Typical ifconfig going forward (now)

```
aaronh@services1.bind.com:/Users/aaronh> ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
	options=3<RXCSUM,TXCSUM>
	inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
	inet 127.0.0.1 netmask 0xff000000
	inet6 ::1 prefixlen 128
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
	options=2b<RXCSUM,TXCSUM,VLAN_HWTAGGING,TSO4>
	ether 40:6c:8f:59:51:72
	inet6 fe80::426c:8fff:fe59:5172%en0 prefixlen 64 scopeid 0x4
	inet6 2607:fae0:1:1:426c:8fff:fe59:5172 prefixlen 64 autoconf
	inet6 2607:fae0:1:2:426c:8fff:fe59:5172 prefixlen 64 autoconf
	inet 75.149.49.37 netmask 0xfffffff8 broadcast 75.149.49.39
	inet6 2607:fae0:1:1:9088:c85c:e31d:766 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:2:9088:c85c:e31d:766 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:1:1da4:87e0:7eae:8459 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:2:1da4:87e0:7eae:8459 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:1:a1fc:241c:c4f9:c5b2 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:2:a1fc:241c:c4f9:c5b2 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:1:80d7:8d8e:738f:4376 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:2:80d7:8d8e:738f:4376 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:1:451c:52d7:129a:767b prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:2:451c:52d7:129a:767b prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:1:d494:2bb0:f184:1c29 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:2:d494:2bb0:f184:1c29 prefixlen 64 deprecated autoconf temporary
	inet6 2607:fae0:1:1:18ef:2957:8b85:ecdc prefixlen 64 autoconf temporary
	inet6 2607:fae0:1:2:18ef:2957:8b85:ecdc prefixlen 64 autoconf temporary
	media: autoselect (1000baseT <full-duplex>)
	status: active
en1: flags=8823<UP,BROADCAST,SMART,SIMPLEX,MULTICAST> mtu 1500
	ether 7c:d1:c3:d7:5b:61
	media: autoselect (<unknown type>)
	status: inactive
p2p0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 2304
	ether 0e:d1:c3:d7:5b:61
	media: autoselect
	status: inactive
fw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 4078
	lladdr 00:3e:e1:ff:fe:5c:a8:f8
	media: autoselect <full-duplex>
	status: inactive
aaronh@services1.bind.com:/Users/aaronh>
```


6connect

DNS is essential

- No longer memorable
- Multi-stack
- MDNS link local access
- How do we deal with temp addresses? Dynamic DNS?
- This process really needs to be automated
  - PTRs are no longer 1 octect
  - E.g. 2001:db8:2101::451c:52d7:129a:767b
  - b.7.6.7.a.9.2.1.7.d.2.5.c.1.5.4.0.0.0.0.1.0.1.2.8.b.d.0.1.0.0.2.ip6.arpa
  - Zone: 1.0.1.2.8.b.d.0.1.0.0.2.ip6.arpa
  - b.7.6.7.a.9.2.1.7.d.2.5.c.1.5.4.0.0.0.0 IN PTR somehost.foo.com.

Debugging is going to get complicated

- Need ability to search for any element and return all relevant data quickly
  - IPv4
  - IPv6
  - Hostname
  - Asset name
  - VLAN
  - Temporary Addresses
  - MAC Address
  - Asset ID / Tag

Numbering plans have changed

- These are no longer guidelines but strict policy
- 2 Years -> 10-20 years
- Tie downs -> Regions permanent and should be 1
- Large nibble bound reservations for purpose
- Many Allocation & Assignment Methodologies
  - Get next
  - Reservations
  - Stagger
  - Sparse (RFC and $myversion)
  - Human Readable (decimal)

Planning and rolling up

- Define _your_ end-site / resource requestor / smallest
- **End-Site Definition**
  - The quantity of end sites belonging to a network represents the allocated objects used for the initial network prefix sizing justification. Each end site is assigned a unique End-Site ID prefix from a Network-ID prefix allocation. An end site is defined as an end-user (subscriber) edge network domain that:
  - Receives transit service from a network under separate administration
  - Does not provide transit service to other end sites
  - Requires multiple subnets (/64).

Calculating size

Prefix Length	Range			Total Sites per Allocation
48	1	To	1	1
44	2	To	5	16
43	6	To	8	32
42	9	To	12	64
41	13	To	18	128
40	19	To	28	256
39	29	To	58	512
38	59	To	91	1,024
37	92	To	142	2,048
36	143	To	223	4,096
35	224	To	350	8,192
34	351	To	549	16,384
33	550	To	861	32,768
32	862	To	2,353	65,536
31	2,354	To	3,822	131,072
30	3,823	To	6,208	262,144
29	6,209	To	10,086	524,288
28	10,087	To	16,384	1,048,576

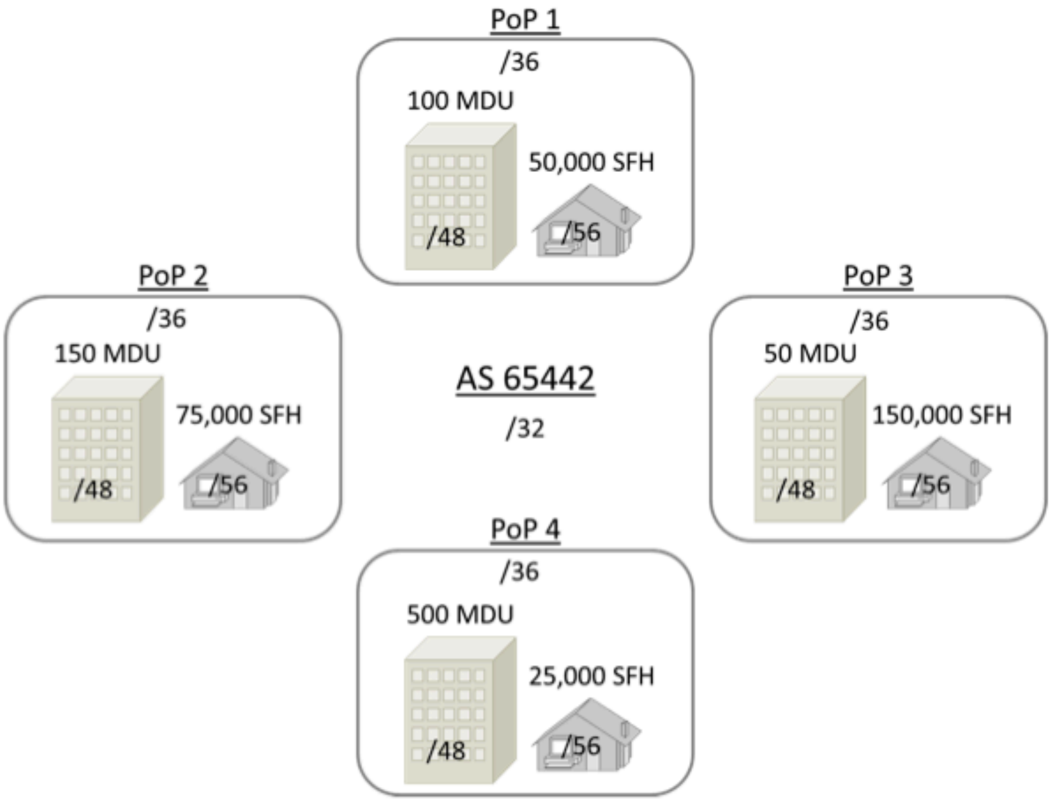
Small Allocation (48–40)

Medium Allocation (39–33)

Large Allocation (32–28)

Calculate your tie-down

- Add add add add multiply and round up to nibble



Write a detailed plan and enforce it

- Find a provisioning platform which will allow you to enforce your IPAM IPv4, IPv6 policies along side of DNS, DHCP, Assets and templatize your process.

- Modify your own provisioning system

- Write your own

Discovery?

- In an IPv4 world this means ICMP packets and SNMP for a few hundred possible hosts per subnet
- In an IPv6 world this means 18,446,744,073,709,551,616 quintillion (1 /64) hosts per subnet. Scanning time would be impossible so this means using real NM on routers and switches.

DHCP pool management

- Now v4 and v6
- Could be mix of SLAAC and v4 DHCP short term
- Long term don’t plan on v4 nameservers
- DHCP-PD use will increase as vendor support does

Conclusions

- Management tools are no longer spreadsheets
- Provisioning has changed dramatically and is no longer easy to roll your own.
- Planning has changed and is for a much longer term and can be templatized and automated.
- Policy becomes much more important as Resource Requestors become more automated / critical
- All Provisioning systems must support IPv6 and DNSSEC, Device growth, automation, reporting, easy search.

Quick example

- IPv6 subnetting example..

Questions?

- Aaron Hughes
  - President & CTO, 6connect
  - aaron@6connect.com