IP Mapping Method

Over time, the terminology for the numbers assigned to buildings and members has evolved.

Originally a person registering was issued automatically a “Node Number”. The Node Number was then used to assign an IP address range via an algorithm to the set of devices to be installed on the building. In addition this was a bit confusing since there could be more than one member in the same building to be connected to the same set of roof devices - so the Node Number of the first installed member was used to determine the IP range. The second member connected to the same set of devices did not get their Node number used. As the IP address range were assigned based on the Node Number a big number of IP address ranges where "unused" and thus "wasted". Same for members registering but not able to connect yet, a Node Number was attributed but not used.

Since 2020 we use the terms “Install Number” (#) to refer to the number assigned to the member when a member joins (*in place of Node Number*) at the same time it was decided that registration getting a numerical above 8000 as (#), and needing a set of roof devices (building not yet connected to NYC Mesh) were going to get a Network Number (NN), a number that has not been used so far.

The “Network Number” (NN) refers to the number assigned to the devices on the roof. When a member joins a (NN) may or may not already exit for the building. A (NN) gets assigned to the devices when they get actually installed.
Note: The (NN) is not assigned to a member but to a set of devices installed (or to be installed) on the building. The member install number (#) is then linked to the (NN) where it connects to, and noted like *#123* NN*456* (Install number #123 connected to the devices at the node NN456).

Assigning the NN only at install time preserves address space and allows us to reclaim unused Network Numbers (e.g., if an install is abandoned for a period of time).

As we completed more and more installs, we needed a way use the IP address ranges that were unused.

The algorithm for mapping IP addresses to Node Number and later to Network Numbers (NN) was the result of a lot of brainstorming in the #architecture channel on Slack.

Since 2019 we have implemented the “Human Easy Split” strategy (Strategy 3 below), allowing up to 25,599 Network Numbers to be assigned.

```
Hi #architecture,


I'd like to propose an algorithm for making a Node Number to IP mapping programmatically.
```

This is not a *new* idea. Several of us have thought about it and taken a stab at it, and I'd like to officially see if we can all agree on one.
Also, the below ideas represent generally a few models of what can be done. There are infinite variations, but the generally follow the below pattern.

The idea:
- For a given Node Number (X), and a given IPv4 Slash 16 address space (ex: 10.0.0.0/16), Get a unique IP in the range for that node.
- The mapping is generic. It could be used as a node identifier, a peering LAN, link local, public, or private.
- If we can agree, it will be much easier to peer, login to node by name/number, and generate and automate node setup without a central authority.

Use cases:
- We could for example give each node an identifier so we can always find the node and it will announce itself
- We could OSPF peer without needing additional filtering or mapping. Config can be generated
- We could BGP peer any to any without filtering, just by typing the node number of the peer
- WDS, litebeam, and wired connections can be used simultaneously without doubling the number of addresses needed each time.

For the below, two numbers will be given, Y and Z, they can be seen as 10.0.Y.Z for mapping in to the /16 block

---
Strategy 1 - As Needed:
For each node that needs an address, pick the next unused address and assign it.
This is what we do now.
Ex:
node 1 = □0 1
node 500 = □0 2
node 392 = □0 3

Advantages:
- Can fit the most nodes in the space as possible (65K)
Disadvantages:
- Need a huge lookup table
- Supereasy to overlap and make a mistake with another node

Strategy 2 - Straight Allocation:
For each node, split the upper and lower bits of the number across the octets, resulting in each 256 node rolling over to the next /24.
Ex:
node 10 = □0 10
node 200 = □0 200
node 256 = □1 0
node 500 = □1 245
node 2218 = □8 178
node 5000 = □19 155
node 7000 = □27 115
node 10000 = □39 55
node 11000 = □43 35

Advantages:
- Can fit the most nodes in the space as possible (65K)
Disadvantage:
- Not human understandable, will need to reference a calculator.
- Only one node per site is assumed

Strategy 3 - Human Easy Split:
For each node number, split the number phyiscally in two parts and map the last two digits to the last octet, and the first two (or three) digits to the second-to-last octet
If a node will have more than one router, increase the last octet by 100. Since we only use 0-99, 100-199 can be secondary for all nodes without hurting the human readability.
This also reserves an extra 55 IPs (200-255) for every 100 nodes as more-than-secondary IPs that would have to be assigned statically but unlikely to hurt or overlap.
Ex:
node 10 = □0 10
node 200 = □2 00
node 256 = □2 56
node 500 = □5 00
node 2218 = □22 18
node 5000 = □50 00
node 7000 = □70 00
node 7998 = □79 98
  rtr 1   □79 198
node 7999 = □79 99
node 8000 = □80 00

node 8001 = □80 01

node 10000 = □100 00

node 11000 = □110 00

node 25599 = □255 99

Advantage:

- Human understandable
- Built-in allowance for more than one router per node
- Can be calculated using string tools in addition to math tools

Disadvantages:

- Inefficient allocation, Only 25599 nodes can be used
- No logic for more than 2 routers per site (will need to manually assign)

Strategy 4 - Bit Shift with span:

For each node number, map it to the linear numbers in the range, but shift the bits up two so that each node gets 4 addresses. This will allow for 4 routers at each site, and create a virtual /30 for each site (4 addresses)

Ex:

node 1 = 00000000 000001XX

node1rtr1□□00000000 00000100

node1rtr2 00000000 00000101

node1rtr3 00000000 00000110

node1rtr4 00000000 00000111

node 2 = 00000000 000010XX

node1rtr1 00000000 00001000

node1rtr2 00000000 00001001

node1rtr3 00000000 00001010

node1rtr4 00000000 00001011

node 10 = 0 40

rtr 1□ 0 40

rtr 2□ 0 41

rtr 3□ 0 42

rtr 4□ 0 43

node 11 = 0 44

rtr 1□ 0 44

rtr 2□ 0 45

rtr 3□ 0 46

rtr 4□ 0 47

```
node 200 = 3 44
node 256 = 4 16
node 500 = 7 236
node 2218 = 35 52
node 5000 = 79 92
node 7000 = 111 28
node 7998 = 126 240
  rtr 1     126 241
node 7999 = 126 244
node 8000 = 126 248
node 8001 = 127 0
node 10000 = 158 184
node 11000 = 174 152
node 16118 = 255 212
ndoe 16127 = 255 248

Advantage:
- Very efficient allocation of space
- Support for four routers per site
- No manual mapping will be needed even for special cases
- Even though the mapping is complicated, all routers at a site will be adjacent numerically.
Disadvantage:
- Not human readable, even more complicated mapping that will require a calculator.
- Not as many nodes as other strategies, only 16128 nodes can be assigned.
```

Revision #7
Created 9 December 2023 04:39:49 by Willard Nilges
Updated 29 July 2024 00:43:42 by Olivier