

## Mapping the whole internet with Hilbert curves

The internet is big. [Really big](#). You just won't believe how vastly, hugely, mind-bogglingly big it is. I mean, you may think the /22 you got as a [LIR](#) was big, but that's just peanuts to the internet.

Well, actually, it wasn't in the long run, that's why we need IPv6. But that is a different story.

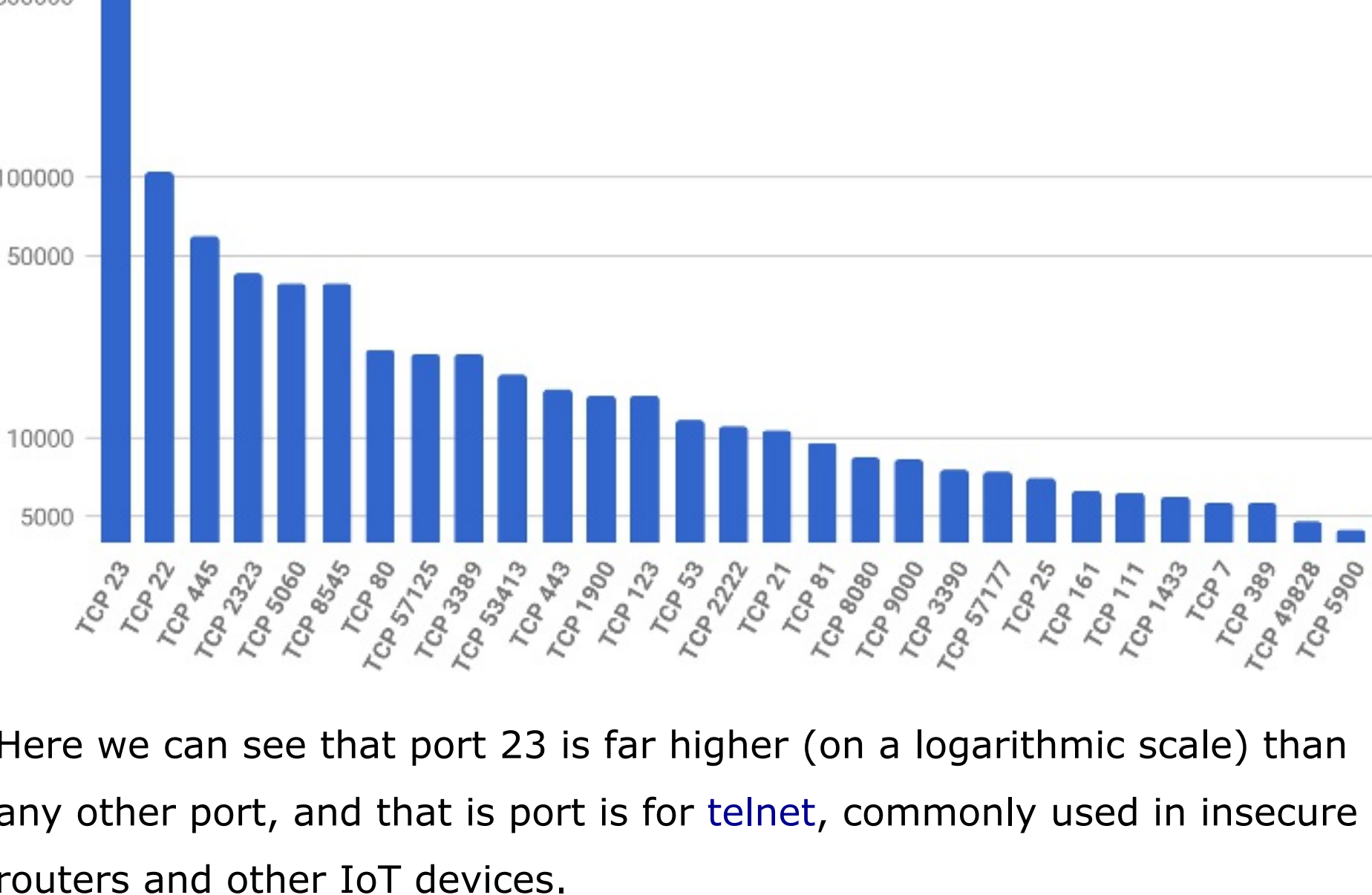
The point is, IPv4 (the most common deployed version of the IP protocol) sets its address limits at  $2^{32}$  addresses. This means you have roughly 4.2 billion IP addresses to work with, except you don't really, because large sections are not usable:

IP Range	Use
0.0.0.0/8	Local System
10.0.0.0/8	Local LAN
127.0.0.0/8	Loopback
169.254.0.0/16	"Link Local"
172.16.0.0/12	Local LAN
224.0.0.0/4	Multicast
240.0.0.0/4	"Future use"

The blocks ( shown as [CIDR notation](#) ) above already wipe out 588,316,672 addresses, or about 13% of all addresses.

However giving the remaining 3,706,650,624 addresses, When you consider it, isn't that many and is perfectly within reach of sending a packet to *every single one*.

Now. This isn't the first time someone has done this, the internet has a considerable amount of "background noise" (unsolicited packets) on it. Mostly created by systems looking for other systems to hack.



Here we can see that port 23 is far higher (on a logarithmic scale) than any other port, and that is port is for [telnet](#), commonly used in insecure routers and other IoT devices.

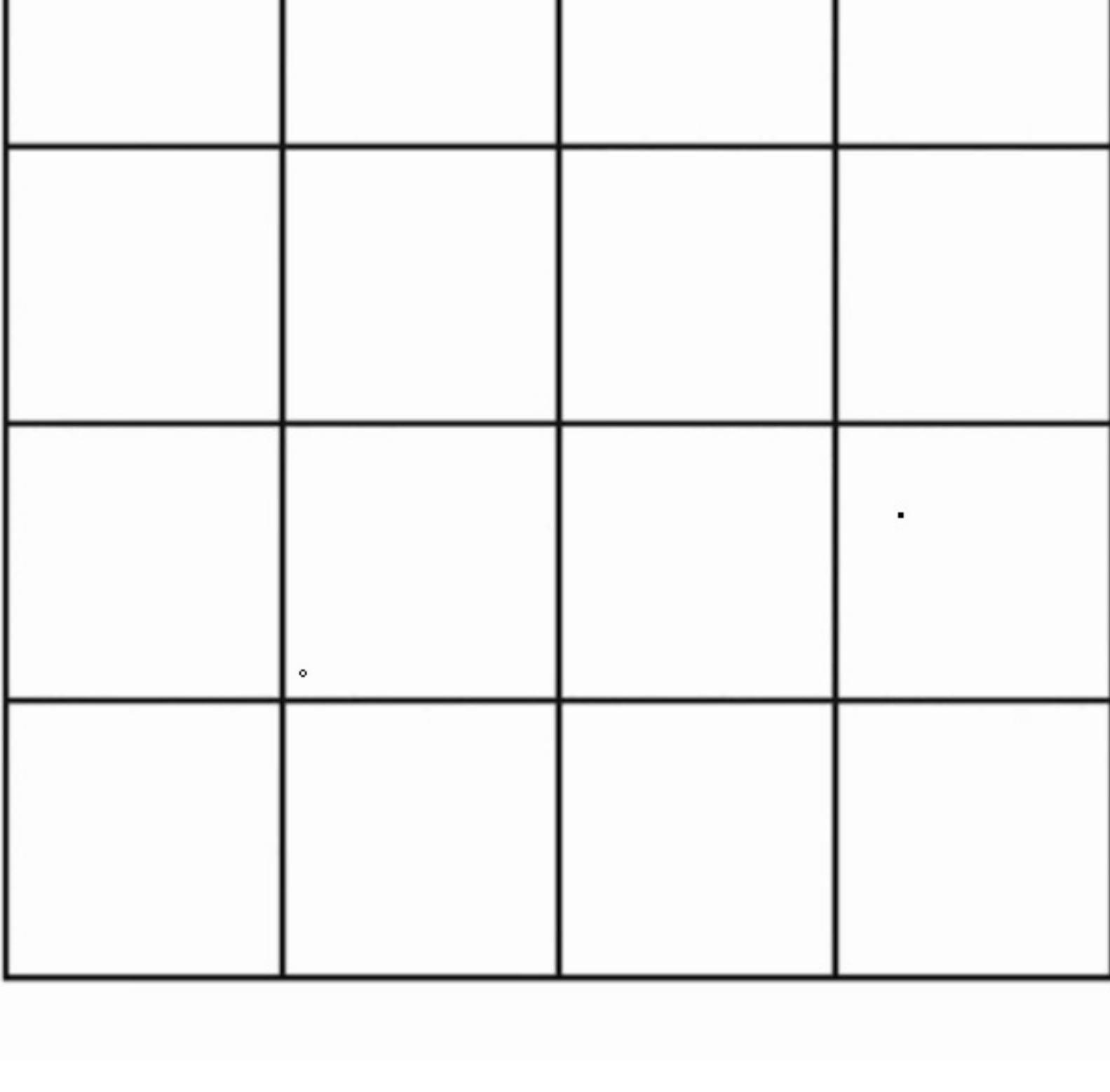
With that known, I speeded ahead and send a ICMP ping to every host on the internet to see how much of the internet responds to a ping (and thus indicating there is a connected computer on the other side of it)

After around a day later, I had sent 3.7 billion packets and had a large text file. Now we just had to find a way to draw it!

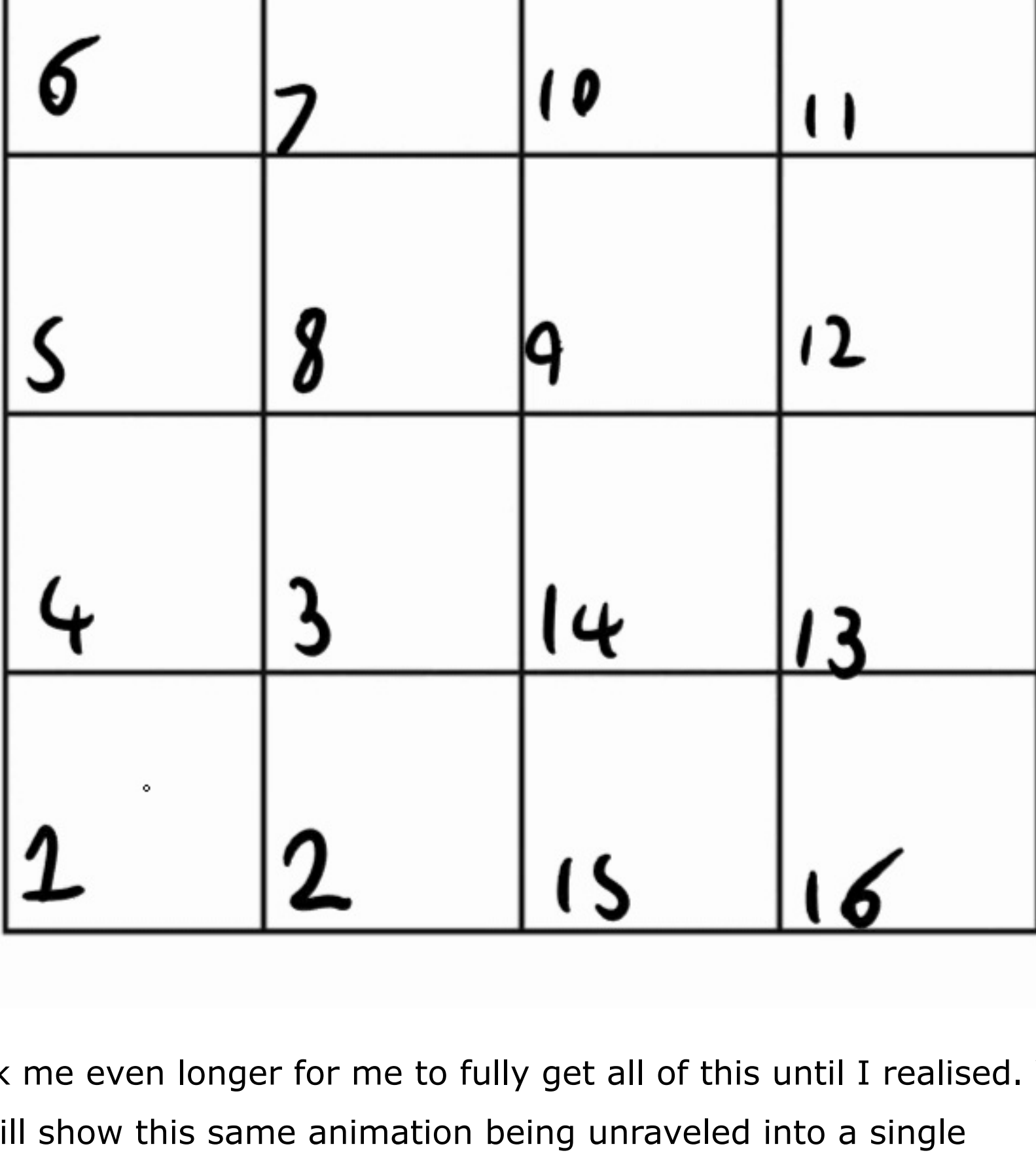
### Introducing Hilbert curves

The problem with displaying IP addresses, is that they are a single dimensional, they only move up and down, however humans are not good at looking at a large amount of single dimensional points. So there has to be a way to fill a 2 dimensional space that can also help the structure of the graph stay in shape.

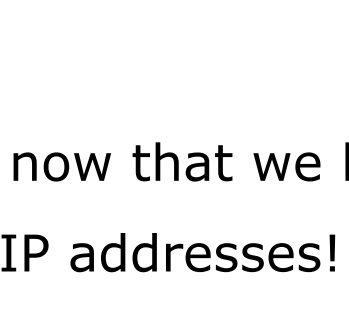
Luckily maths has our back again, with [space filling curves](#)



For me it didn't make much sense until I numbered the nodes it was passing through.



It took me even longer for me to fully get all of this until I realised. You can still show this same animation being unraveled into a single dimension again:



Anyway, now that we know these graphs work, we can start applying them to IP addresses!

Thankfully [there are tools that can already produce these graphs](#) in relation to IP addresses so it is just a case of loading that data in and producing the graph:

```
cat ping.txt | pcregrep -o1 ':(\d+\.\d+\.\d+\.\d+)' | ./ipv4-heatmap -a ./labels/iana/iana-labels.txt -o out.png
```

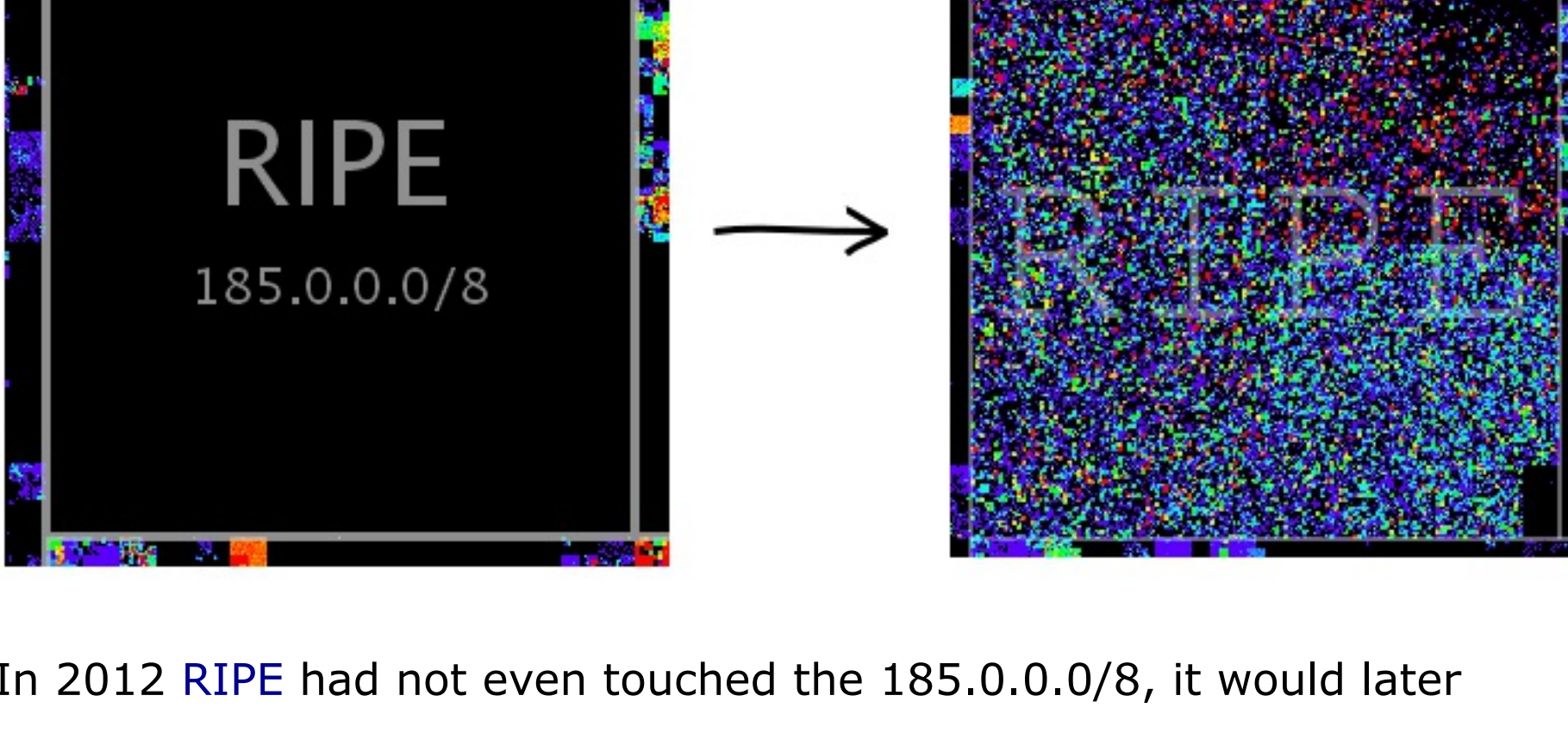
This renders a hilbert curve with a color gradient showing how many systems are online in that /24

and so I present, The IPv4 Internet on 16th April 2018:



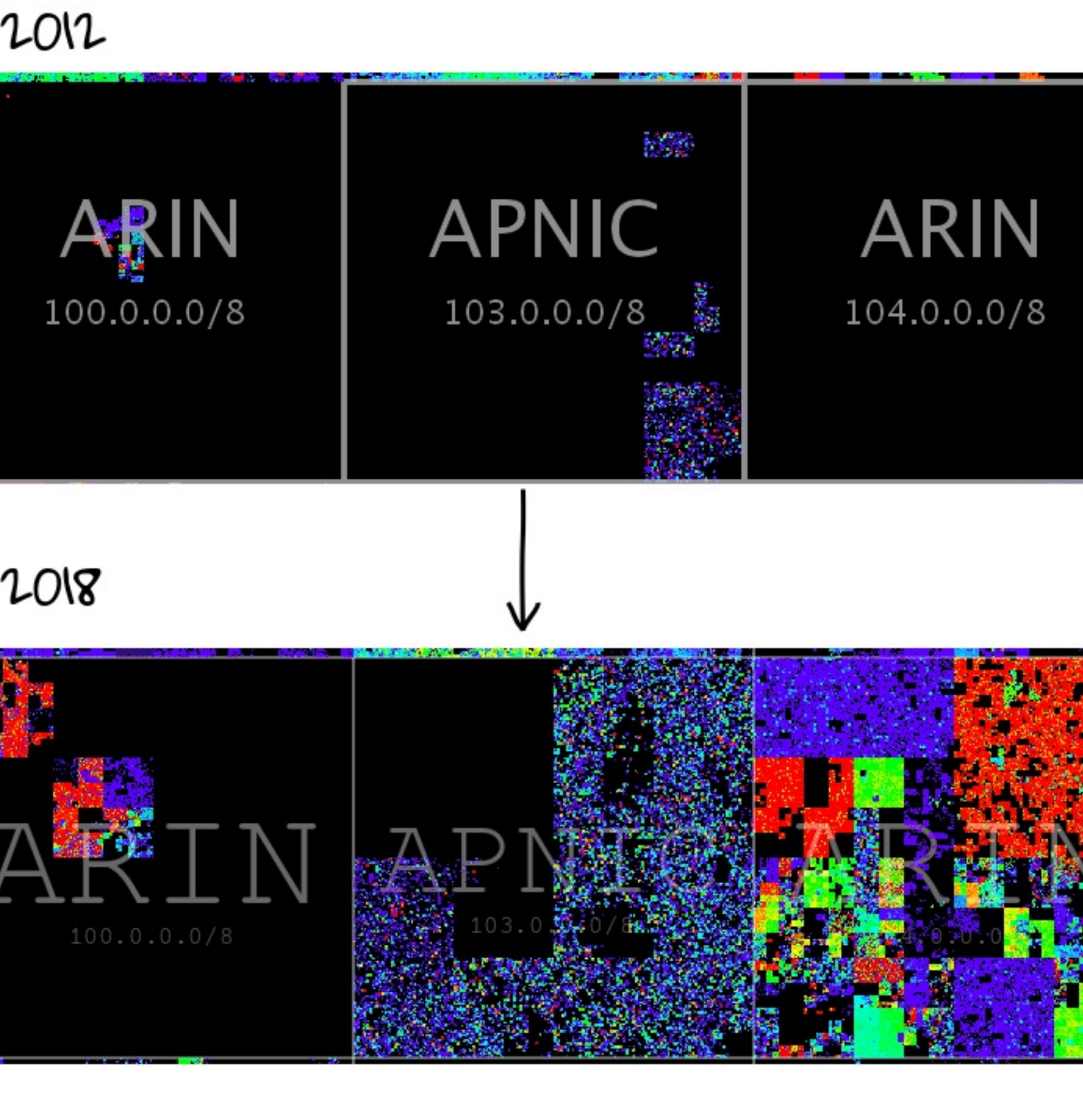
You can click on the image for a lossless and full resolution version, however do be warned that it's 9MB.

The last public scan I know if was in 2012 and was done by the Carna botnet, using this data we can easily see some changes.

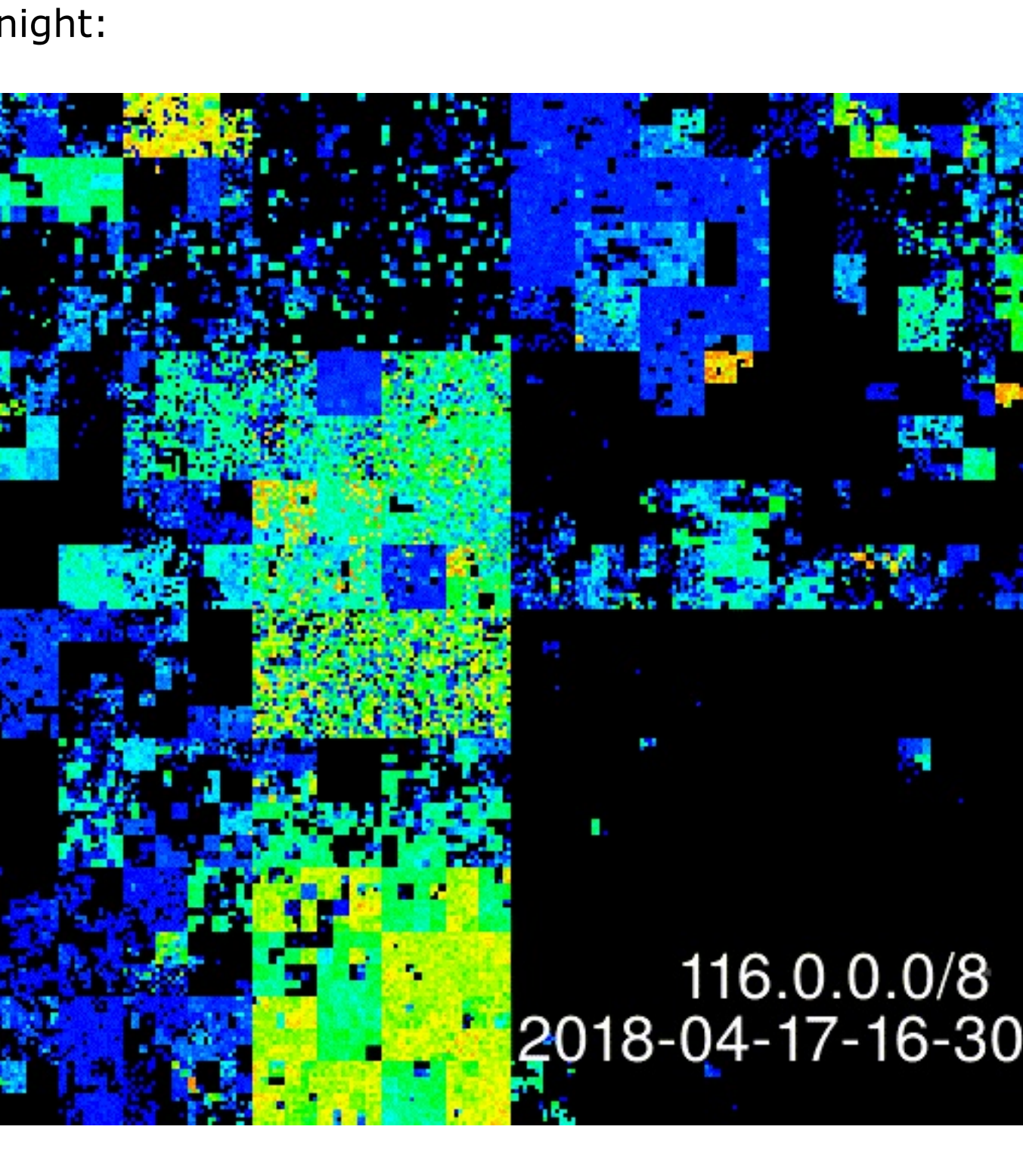


In 2012 [RIPE](#) had not even touched the 185.0.0.0/8, it would later become the block they would use for the last allocations, and would only give out a /22 of IP space to every new member of RIPE. This makes 185.0.0.0/8 odd looking among the other blocks, and there are no mass allocations, and so the blocks looks very "spotty" compared to others.

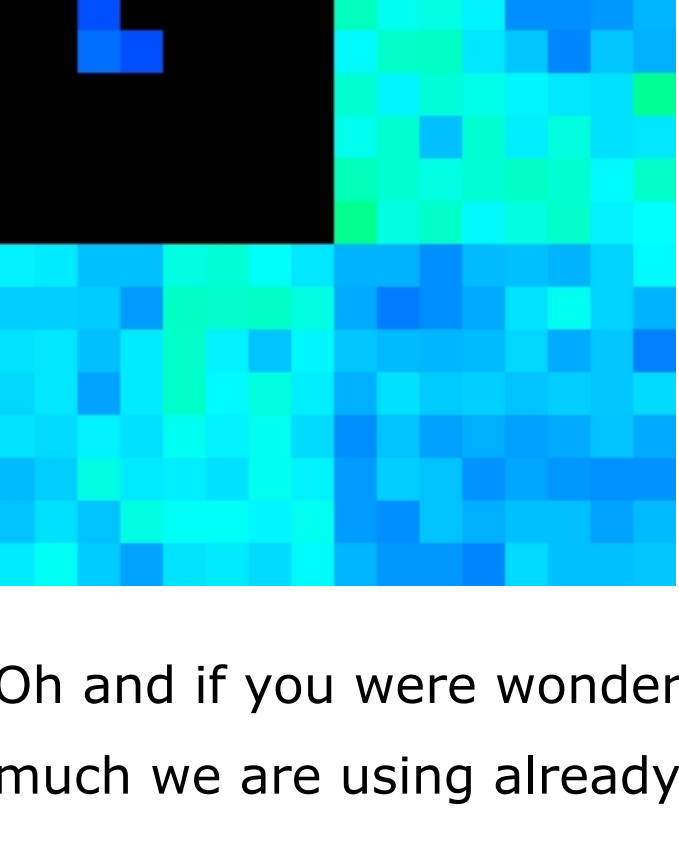
RIPE is not the only one to have completely used up blocks in this time. Below we see 3 different RIRs consume their blocks in the space of 6 years.



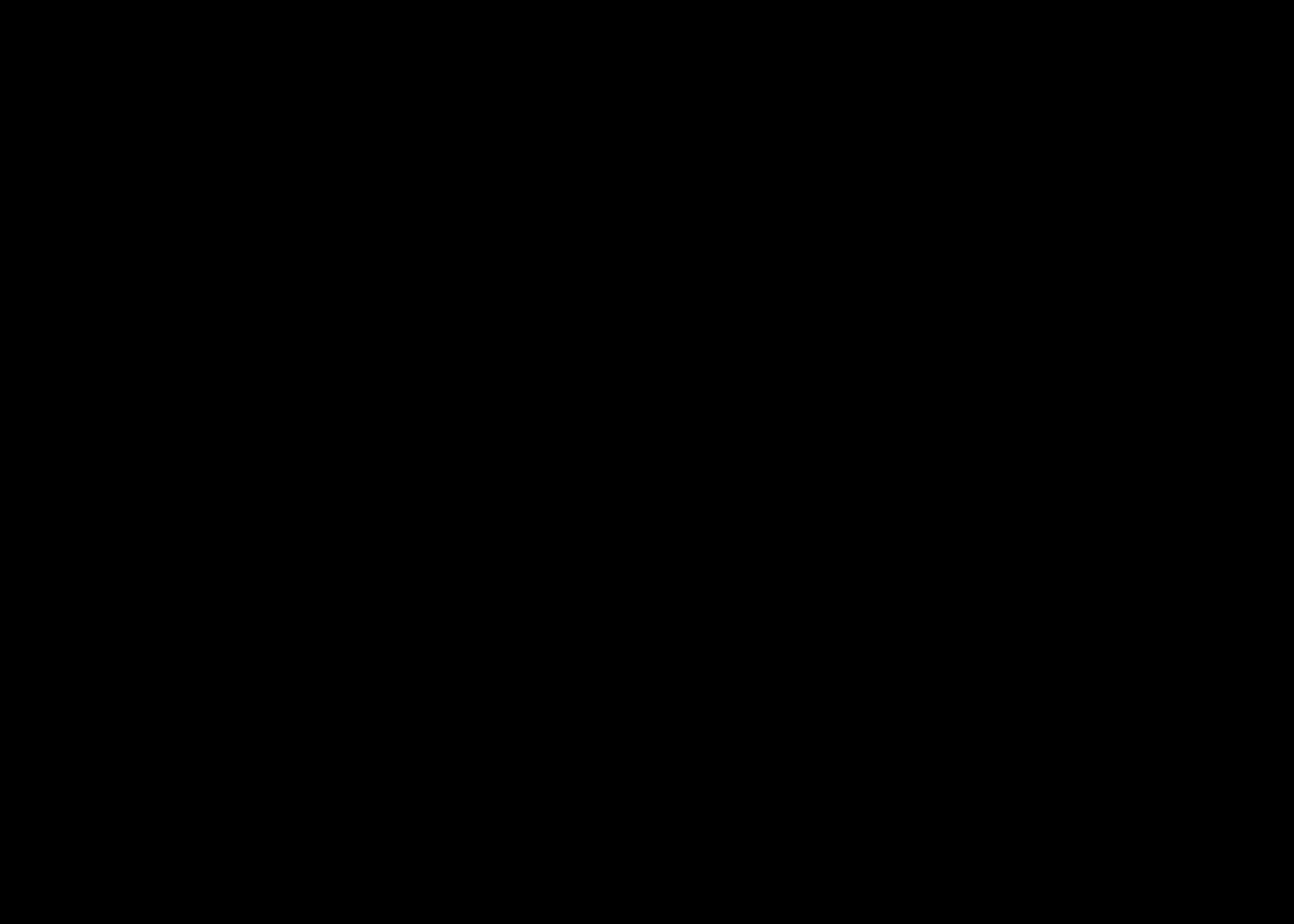
On top of all of this, I also did a bonus scan of a few APNIC IP blocks every 30 mins for 24 hours. The data from that allows you to see the internet "breathe" as clients come online in the morning and offline at night:



One of the more interesting finds in this gif was a what looks like a dynamic IP pool from a ISP, showing clients come online for a short amount of time, and then connecting and getting a new IP address (hence more the more active IP addresses are "moving" during the day)



Oh and if you were wondering what IPv6 looks like in this form and how much we are using already:



And if you enjoyed this, you will be glad to know that I am going to be at [Recurse Center](#) in NY for the next 9 weeks! Meaning you can follow my [Twitter](#) or [RSS](#) to keep up with the other silly (or sometimes sensible) things I will do!