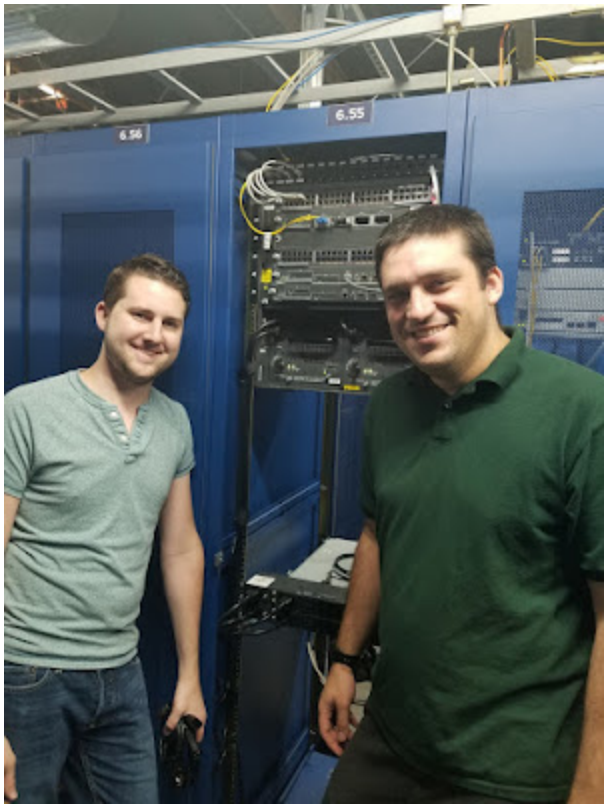
November 15, 2017

# CREATING AN AUTONOMOUS SYSTEM FOR FUN AND PROFIT

—




At its core, the Internet is an interconnected fabric of separate networks. Each network which makes up the Internet is operated independently and only interconnects with other networks in clearly defined places.

For smaller networks like your home, the interaction between your network and the rest of the Internet is usually pretty simple: you buy an Internet service plan from an ISP (Internet Service Provider), they give you some kind of hand-off through something like a DSL or cable modem, and give you access to "the entire Internet". Your router (which is likely also a WiFi access point and Ethernet switch) then only needs to know about two things; your local computers and devices are on one side, and the ENTIRE Internet is on the other side of that network link given to you by your ISP.

For most people, that's the extent of what's needed to be understood about how the Internet works. Pick the best ISP, buy a connection from them, and attach computers needing access to the Internet. And that's fine, as long as you're happy with only having one Internet connection from one vendor, who will lend you some arbitrary IP address(es) for the extend of your service agreement, but that starts not being good enough when you don't want to be beholden to a single ISP or a single connection for your connectivity to the Internet.

That also isn't good enough if you *are* an Internet Service Provider so you are literally a part of the Internet. You can't assume that the entire Internet is *that* way when half of the Internet is actually in the other direction.

This is when you really have to start thinking about the Internet and treating the Internet as a very large mesh of independent connected organizations instead of an abstract cloud icon on the edge of your local network map.

### Which is pretty much never for most of us.

Almost no one needs to consider the Internet at this level. The long flight of steps from DSL for your apartment up to needing to be an integral part of the Internet means that pretty much regardless of what level of Internet service you need for your projects, you can probably pay someone else to provide it and don't need to sit down and learn how BGP works and what an Autonomous System is.

### But let's ignore that for one second, and talk about how to become your own ISP.

To become your own Internet Service Provider with customers who pay you to access the Internet, or be your own web hosting provider with customers who pay you to be accessible from the Internet, or your own transit provider who has customers who pay you to move their customer's packets to other people's customers, you need a few things:

1. Your own public IP address space allocated to you by an Internet numbering organization
2. Your own Autonomous System Number (ASN) to identify your network as separate from everyone else's networks
3. At least one router connected to a different autonomous system speaking the Border Gateway Protocol to tell the rest of the Internet that your address space is accessible from your autonomous system.

Once your router tells other networks that you're now the home of some specific range of IP addresses, and that advertisement propagates out through the rest of the Internet, everyone else's routers will have an entry in their routing tables so if they see any packets with your address on them, they know which direction to send them so they eventually end up at your door step.

### Wait, but why don't you need any of this for your home Internet?

So why doesn't your home router need to speak BGP or you need to own public IP space to be reachable from the rest of the Internet? Because your ISP takes care of that for you. In addition to funding the wiring from their data center to your house, the $50/month you pay to your ISP funds them getting address space allocated for you, advertising it to the rest of the Internet, and getting enough connectivity to the rest of the Internet that your packets can get where they're headed.

The same answer is true on the other end when you spin up a web server somewhere like Digital Ocean or Amazon Web Services; they handle IP addressing and BGP for you so all you need to worry about is setting up the software for your own corner of the Internet on the one specific address they set aside for you from their big pools of addresses that they manage.

### Wait, but why am I blogging about BGP then?

If you've made it this far, you're probably pretty curious why I'm talking about BGP at all, and what this blog post is leading up to.

So... I recently set up my own autonomous system... and I don't really have a fantastic justification for it...

My motivation was twofold:

1. One of my friends and I sat down and figured it out that splitting the cost of a rack in Hurricane Electric's FMT 2 data center marginally lowered our monthly hosting expenses vs all the paid services we're using scattered across the Internet which can all be condensed into this one rack.

And this first reason on its own is a perfectly valid justification for paying for co-location space at a data center like Hurricane Electric's, but isn't actually a valid reason for running it as an autonomous system, because Hurricane Electric will gladly let you use their address space for your servers hosted in their building. That's usually part of the deal when you pay for space in a data center: power, cooling, Internet connectivity, and your own IP addresses.

2. Another one of my friends challenged me to do it as an Autonomous System.

So admittedly, my justification for going through the additional trouble to set up this single rack of servers as an AS is a little more tenuous. I will readily admit that, more than anything else, this was a "hold my beer" sort of engineering moment, and not something that is at all needed to achieve what we actually needed (a rack to park all our servers in).

But what the hell; I've figured out how to do it, so I figured it would make an entertaining blog post. So here's how I set up a multi-homed autonomous system on a shoe-string budget:

## Step 1. Found a Company

You're going to need a legal entity of some sort for a few steps here, so you're going to need a business name. I already happened to have one from other projects, so at the minimum you'll want to go to your local city hall and get a business license. My business license cost me the effort to come up with a kick-ass company name and about $33/year, and I've never even gotten around to doing anything fancy like incorporating it, so it's really just a piece of paper that hangs in my hallway and allows me to file 1099-MISC forms on my tax returns within the city of Sunnyvale, CA. In the context of this project, this business license primarily just needs to look official enough to get me approvals when I apply for an autonomous system number needed to set up my own network.

## Step 2. Get Yourself Public Address Space

This step is, unfortunately, probably also the most difficult. You need to get yourself a block of public IP addresses big enough to be advertised over BGP (there's generally agreed upon minimums to keep the global routing table from getting ridiculous) and allocated for you to advertise over BGP yourself. You'll probably want both IPv4 addresses, which have to be at least a /24 subnet (256 addresses) and IPv6 addresses, which have to be at least a /48 subnet (65536 subnets of /64).

The big problem is that there are no IPv4 addresses left. There was only 4 billion of them in the first place, and we've simply run out of them, so the "normal" procedure of going to your local Internet numbers organization like ARIN isn't that productive. If all you need is IPv6 space (which is unlikely) and you happen to be in North America, you actually can still go to ARIN and request resources, but IPv4 addresses are generally still really needed. There's other solutions like buying IPv4 space on the second hand market, but that's getting expensive, so here's probably the least helpful part of this whole blog post:

*I just borrowed some address space from my friends.*

For example, I've got another friend who, for a different project, got a /32 IPv6 allocation from ARIN, which is a metric TON of addresses, so I asked him if I could have a (relatively small) /48 sub-allocated from his /32, so he drafted me an all official looking "Letter of Authorization" on his company letterhead that literally just says:

**"Dear Sirs,**
**"Please accept this letter of authority on behalf of [FRIEND'S COMPANY NAME] to permit the BGP announcement of [/48 IPv6 SUBNET INSIDE HIS /32 SUBNET] by [KENNETH'S COMPANY NAME].**
**"Sincerely, [FRIEND'S SIGNATURE]"**

It's not as impressive as having IP space with my name on it in ARIN's database, but it's also a whole hell of a lot cheaper than even the smallest address allocation you can get from ARIN (a couple beers vs $250/year).

This letter of authorization is also the first instance of where learning about how the Internet *actually* works gets a little weird. That letter is literally all it took for me to take control of a sub-block of someone else's public address space and get it routed to my network instead of theirs. Some of my network peers later asked for me to provide this LoA when we were setting up my network links, but that means I just sent them a PDF scan of a letter with my friend's signature on it. And I mean an actual signature; not some kind of fancy cryptographic signature, but literally a blue scribble on a piece of paper.

To be fair, the letterhead looked very official.

## Step 3. Find Yourself Multiple Other Autonomous Systems to Peer With

So the name of the game with the Internet is that you need to be interconnected with at least one other part of it to be able to reach any of it, but that isn't necessarily good enough here. If you were only peering with one other autonomous system, you probably wouldn't even need to run BGP, and if you did, you could even do it using a "private" autonomous system number (ASN) which your upstream provider could just replace with their own before passing your routes on to the rest of the Internet.

But that's not good enough here. I didn't want to use some kind of lousy non-public ASN! This project was a personal challenge from a friend and the network engineering equivalent of driving a pickup with a lift kit, so we need a *public* ASN. We're going to later need to apply to ARIN to get one allocated and we'll need to provide at least two other autonomous systems we're going to be peering with to justify the "multi-homed" routing policy we're using to justify ARIN allocating us an ASN.

This multi-homed policy where we're peering with multiple other networks is also kind of neat because it means that if one of our upstream providers decides to take the day off, or only provide us a certain amount of bandwidth to the rest of the Internet, we have alternatives we can use from our peering links into other autonomous systems.

This whole concept of peering and all the different types of peering policies you might want for your network is a pretty

deep rabbit hole, so I actually ended up buying a whole book just on peering, which was very helpful: The 2014 Internet Peering Playbook, Norton. He also has a website, which is a significant fraction of the content of his book in a less curated form.

Peering is definitely one of these "how the sausage gets made" sorts of topics that a lot of networks tend not to like to talk about. Exactly how well connected one network is to other networks is hugely important to their customer's quality of service, so everyone wants to make it appear that they're extremely well connected without showing their hand and letting others see their weaknesses. This means the peering community is rife with quiet backroom handshake deals that are never publicly disclosed, and you can spend hours digging through online looking glass servers that show you the global BGP tables trying to figure out what in the world networks are doing with their peering links.

Long story short, I'm getting a "paid transit" peering link from Hurricane Electric due to renting one of their cabinets, and managed to find a few friends in the Hurricane Electric FMT 2 data center who had spare Ethernet ports on their router and were willing to set up free peering links for what little traffic happens to go directly between our own networks. Free peering links tend to be pretty common when both networks are at about the same level in the network provider / customer hierarchy, so tier 1 transit providers tend to peer for free to make the Internet happen, and lower tier small networks tend to peer for free to by-pass both needing to pay higher level ISPs to transit their traffic when they can move it directly, but if either network thinks they can charge the other for money, that might happen as well.

This is obviously where human networking becomes exceedingly important in computer networking, so start making friends with the peering coordinators for other networks which you expect to be trading a lot of traffic with. Every packet I'm able to shed off onto one of these lateral peering links with another AS is traffic that doesn't tie up my primary 1Gb hand-off from HE and makes my network faster.

## Step 4. Apply for an Autonomous System Number

There are five Internet number organizations world-wide, and since I'm in North America the one I care about is ARIN, so I created an account on ARIN's website and:

1. Created a Point of Contact Record for myself - Pretty much just a public profile for my identity: "Kenneth Finnegan, address, phone number, etc etc"
2. Requested an Organization Identifier for "[MY COMPANY NAME]" and tied my point of contact record to it - This was by opening a ticket and attaching my business license to prove that my company actually exists. Since my company isn't its own legal identity, ARIN internally tracks it as "Kenneth Finnegan (doing business as) [MY COMPANY NAME]", but this doesn't show up on the public listing, so it wasn't a big deal.
3. Requested an ASN for my Organization Identifier - This is where I needed to be able to list at least two other ASes I was planning on peering with.
4. Pay the $550 fee for getting an ASN issued per ARIN's current fee schedule for resources.

The whole process took about a week between setting up the orgID and requesting the ASN, mainly because I didn't quite get either support ticket request right on the first try due to me not quite knowing what I was doing, but in the end ARIN ultimately took my money and issued me an ASN all my own.

## Step 5. Source a Router Capable of Handling the Entire Internet Routing Table

Remember how your home router only needs two routes? One for your local computers and one for the rest of the Internet, so the two routes are probably something like "192.168.1.0/24 (local) 0.0.0.0/0 (WAN)"

Processing a full BGP Internet routing table is a little more involved than that, and means you need a much more powerful router than one you can go buy at Office Depot. You could also probably build a router yourself out of a server running a router operating system like pfsense or just your favorite Linux distro with the right iptables voodoo and a BGP daemon like quagga, but that wasn't part of the originally thrown gauntlet challenge for this project.

The challenge was to use a real Cisco router capable of handling the entire Internet routing table, and I wanted one that can switch it at line speed. Hurricane Electric alone is giving us a 1Gb Ethernet hand-off, not including all the additional bandwidth out of our rack available due to other peering links, so we wanted a router that could definitely handle at least 1Gbps.

Meet the Cisco Catalyst 6506. Yes, it's rack mount, on a 19" rack. And is 12U high, which since a U is 1.75", means that this router is almost two feet high. And 150 pounds. And burns about 1kW.

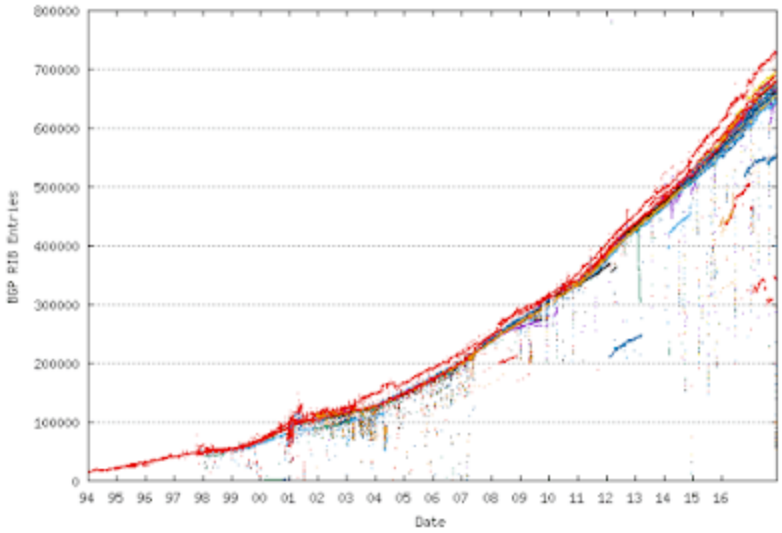
Yes. It's size is ridiculous. Which for this project, isn't entirely out of line.

But it's also kind of a really awesome router, particularly for being a nearly 2 decade old product. The 6500 series is a line of switch/router chassis which support 3,4,6,9, or 13 modular line cards/router supervisors. In the early 2000s this was the best switch that money could buy, and it is definitely showing its age now, but that's perfect. Network engineers love to hate their 6500s because they're so old, but its relatively limited "only" 30 million packets per second throughput is plenty for an autonomous system that fits in a single rack, and its age means I was able to pick up a 6506 configured with dual supervisors and three (3!) x 48 port Gigabit Ethernet line cards on eBay for $625 shipped!

I probably could have found a more reasonably sized router for what I needed, but the 6506 has the appeal that it

definitely has more switching horsepower than I'll ever need for this project, and its biggest downsides are it's size and power, which are both not *that* big of issues since I've got a whole 44U rack for just a few servers and I don't get billed for my power usage. More desirable routers have the big downside that they're actually desirable, so other people are willing to spend a few thousand dollars on them, where I didn't really want to drop $2k on a well kitted out 3945.

The 6506 probably deserves blog posts of its own, but the main thing is that low end configurations of it like this are cheap on eBay, with the one disadvantage that they don't come with a supervisor card with enough memory to handle a full Internet table. This means I did need to scrounge a sup720-XL supervisor that can handle 1 million routes in its routing table. Another few hundred bucks on eBay, or a friend with access to the right kind of e-waste pile solves this problem.




Granted, a million routes isn't actually that much. The public IPv4 table is about 675,000 routes, and IPv6 is another 45,000, and they're both growing fast, so in another 2-3 years the Internet is going to exceed the capabilities of this router. When that happens, I'm going to need to either replace it with something more advanced than this ancient beast or start using some tricks to trim down the full table once we get there. If you'd like to follow along at home and watch the IPv4 routing table march towards the demise of the cat6500, you can find a link to a bunch of graphs here.

I also added a 4 port 10Gb line card, because 10Gb is awesome, and took one of the 48x1Gb line cards out because I really didn't need 144 Ethernet ports on this thing. That's just a ridiculous number of Ethernet ports for a single rack.

So the final configuration is:

1. 48x1Gb line card for my four copper peering links with other autonomous systems, including Hurricane Electric
2. 4x10Gb line card for my peering link with one of my friends who happened to have a spare 10Gb port on his router, and who also thinks 10Gb is awesome. This will probably also serve some local 10Gb links in the rack once I grow beyond one server.
3. A blankoff plate instead of my third 48x1Gb line card to save power.
4. 48x1Gb line card for the local servers in the cabinet. Since we've only got two servers installed so far, there's currently only a 2x1Gb bonded link to my server and a 4x1Gb bonded link + out of band management to my friend's server.
5. The sup720-BXL which does the actual router processing and makes this whole mess a BGP router. The one cable from this card runs up to a console server letting me manage this beast remotely from the comfort of my apartment without standing in a cold data center.
6. One of my spare sup720 (not XL) which can't handle the full Internet table pulled out an inch so it doesn't power up because this seemed like the best place to store it until I figure out what to do with it.

## Step 6. Turn it All On and Pray

Wait, I mean, carefully engineer your brand new network and swagger into the data center confident that your equipment is all correctly configured.

But seriously, I found a few textbooks on BGP network design and happened to have a 13 hour flight to China and back to take a crash course in BGP theory, and spent a week in my apartment with ear plugs in taking a crash course in how to interact with Cisco devices more sophisticated than just setting up VLANs on an Ethernet switch, which is about all my experience with Cisco IOS before this month.

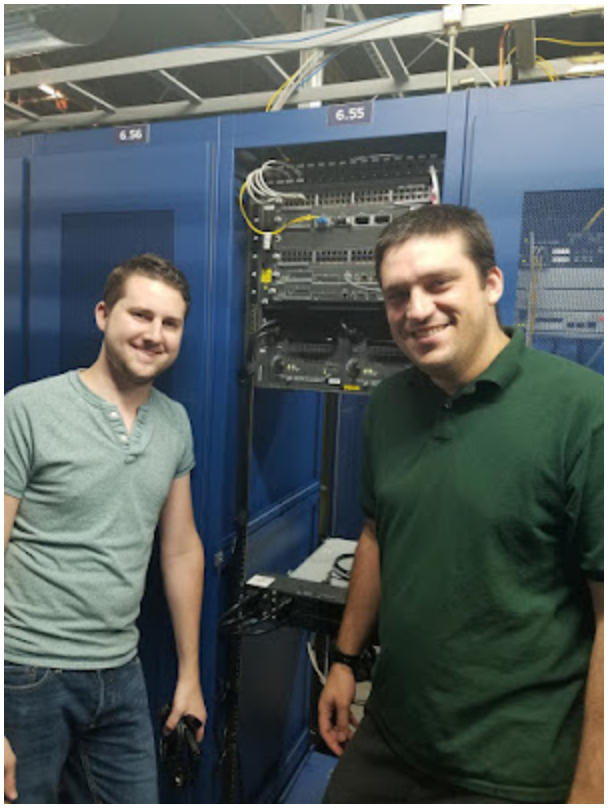
After spending a week lovingly hand crafting my router configuration (while listening to networking podcasters bagging on how ridiculous it is that we still lovingly hand craft our routing configurations), I was ready to deploy my router plus all of our servers in the data center.

When I signed my service agreement with Hurricane Electric, it consisted of:

- One full 44U rack with threaded posts.
- Two single phase 208V 20A feeds.
- A single 1Gbps copper network drop

The network operations center then also emailed me and asked how many of Hurricane's IP addresses I needed, which was two: one for my router's uplink interface, and a second for a serial port console server so if I ever manage to really bork my router's configuration I can still reach it's console port without having to trek over to the data center and stand there in the cold. This means that my hand-off from HE is a /29, so I actually have 5 usable addresses, but that Ethernet drop goes into a fixed eight port GigE switch which breaks out the console server, then plugs into the 6506 for the majority of my Internet traffic.

Once I confirmed that my network feed from HE was live, I then opened a support ticket with HE saying "My BGP router is on [IPv4 ADDRESS] and [IPv6 ADDRESS] and will be advertising these specific routes per attached letters of authorization" and waited for them to set it up on their side, which took less than an hour before I got an email from them saying "we turned it on, and your router connected, so it looks good from here"




And we're off to the races.

At this point, Hurricane Electric is feeding us all ~700k routes for the Internet, we're feeding them our two routes for our local IPv4 and IPv6 subnets, and all that's left to do is order all our cross-connects to other ASes in the building willing to peer with us (mostly for fun) and load in all our servers to build our own personal corner of the Internet.

In the end, setting up my own autonomous system wasn't exactly simple, it was definitely not justified, but some times in life you just need to take the more difficult path. And there's a certain amount of pride in being able to claim that I'm part of the actual Internet. That's pretty neat.

And of course, thanks to all of my friends who variously contributed parts, pieces, resources, and know-how to this on-going project. I had to pull in a lot of favors to pull this off, and I appreciate it.

POPULAR POSTS

May 05, 2021

# UNLOCKING THIRD PARTY TRANSCEIVERS ON OLDER ARISTA SWITCHES

Share



Kenneth Finnegan

VISIT PROFILE

Archive