

IPv6

The Future of the Internet ... Some Day

Do You Need IPv6 ?

NO!

(Well, Probably Not)

Should You Know About IPv6?

YES!

Standard Disclaimer

- I am certainly not an expert on this.
- What I'm showing here is my own experience and the result of my implementation and digging around the internet to find out stuff. It's what happens when you're naturally curious.
- What you see here should be enough to support the home user and even a small business.
- For anything bigger – hire a real expert. Or better yet, **BECOME** a real expert.
- No, I am not planning on becoming an expert – I'm **RETIRED!**

How I Got Involved With IPv6

- When Spectrum boosted my internet speed to 100 Mbps I realized that the router I had (WRT54G) wouldn't do the job.
- When I upgraded to something newer and found that I was seeing IPv6 addresses on my network I had to find out more.
- Being the curious type I started doing a bit of research to see what was in this for me – other than just keeping up with some small part of new technology.
- I made the changes I needed to my server and DNS provider to let outsiders access my web server via IPv6.

The Journey Begins

- There were various trials and tribulations along the way and it was sometimes frustrating.
- It was a great learning experience.
- Most of the journey was easy.
- But the journey should be preceded by some background information ...

IP Protocol Types

- IPv1 through IPv3
 - These were experimental protocols in TCP/IP before the release of IPv4. They are no longer used but their function is served within IPv4 which tied the early research together.
- IPv4 – what we have been and are currently using
- What happened to IPv5 ?
 - This was an experimental protocol that never got off the ground.
- IPv6 is the future (for now) – and the future has arrived.

IPv4 Review

- IP Address are written as four single byte segments with dots between: 192.168.1.123, 71.74.241.26, 8.8.8.8, 255.255.255.0
- Total possible addresses: 4,294,967,296.
- Assigned in blocks of 256 to 16,777,216 addresses.
- Assignable addresses have been exhausted in all regions except Africa which is expected to exhaust theirs in early 2019.
- The only way to get any allocation now is to buy a segment from someone with a surplus.

Stretching Out IPv4

- We have used Network Address Translation (NAT) to stretch out the useful life of IPv4. Our routers do this translation.
- NAT lets you use one address from your ISP to work with several devices on your local network. Your router translates your local IP address (192.168.1.123) to your ISP address (71.74.241.26) and keeps track of what it did so when you get a response it knows who to send the incoming data to.
- NAT is a stopgap solution. It depends on the router to keep track of and redirect incoming traffic to the correct recipient. This will slow down the response time although it mostly goes unnoticed.

How NAT Works

- NAT provides the ability for your private network to access the greater internet through a single IPv4 address.
- Your router sends your request out to the internet after 'remembering' what source port you used and your IP address. The request is sent out with the 'public' IP address of the router substituted for your own IP address.
- When the response comes back the port number is looked up in the internal table and the response is then routed to your 'local' IP address.
- Unsolicited traffic is just thrown away since the router doesn't know where to send it.

The IPv6 Solution

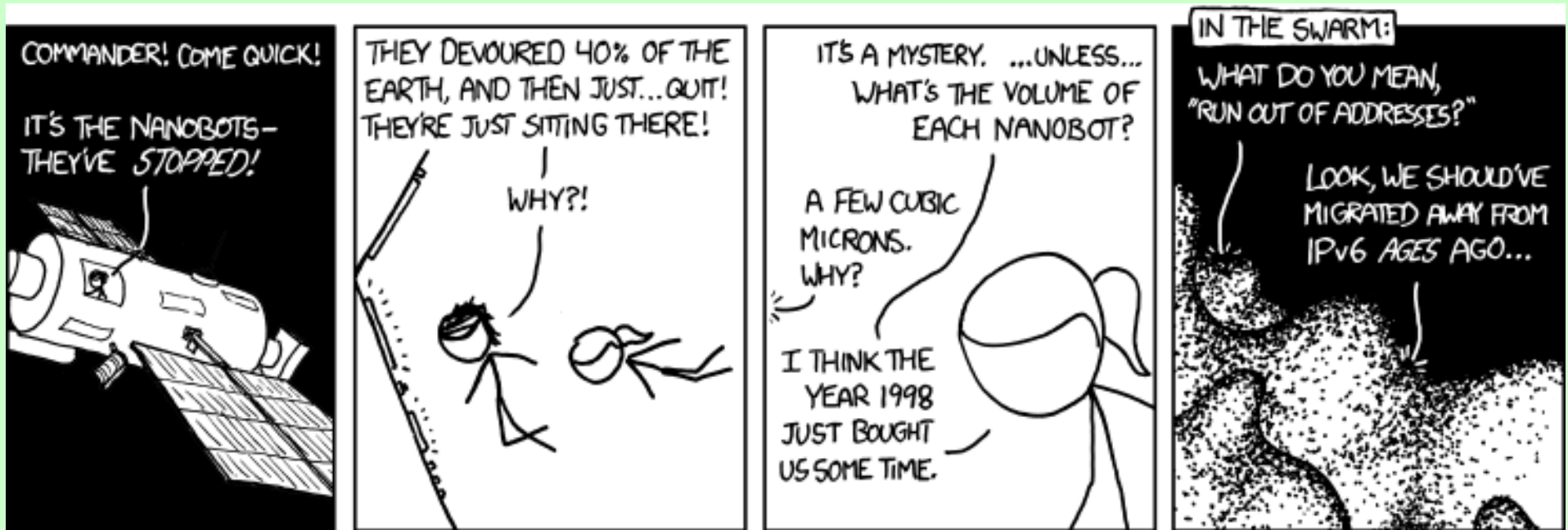
- IPv6 lifts (mostly) the address count limit and eliminates NAT by using what seems to be a really big hammer.
- IPv6 addresses are written as eight groups of 16 bit numbers (in hexadecimal). That's a total of 128 bits as opposed to IPv4's 32 bits.
- The total possible IPv6 addresses:

340,282,366,920,938,463,463,374,607,431,768,211,456

- **That's a REALLY big number!**

Is This Overkill?

Nanobots



<https://xkcd.com/865/>

How IPv6 Addresses are Written

- There are 8 groups of 4 hexadecimal digits:
 - `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`
 - Each group is separated by colons.
 - Leading zeroes in a group are not necessary.
 - Any group that is all zeroes may be skipped. Write as `::`.
 - Any number of consecutive all zero groups may be skipped.
 - `2001:4860:4860:0000:0000:0000:0000:8888` can be written as:
`2001:4860:4860:0:0:0:0:8888` or just `2001:4860:4860::8888`
 - Localhost is: `::1` - nice and simple.

Forcing IPv6 Usage

- **Most commands have a way to force usage of IPv6:**
- **Windows:**
 - `PING -6 ak8b.us`
 - `TRACERT -6 ak8b.us`
- **Linux:**
 - `ping6 ak8b.us`
 - `curl -6 ak8b.us`
- **Web browsers and other internet applications should just use IPv6 and do fallback to IPv4.**

Windows IPCONFIG Output

```
c:\>ipconfig
```

```
Windows IP Configuration
```

```
Ethernet adapter Ethernet 2:
```

```
Connection-specific DNS Suffix . : hiddenrouter.ak8b.us.local
IPv6 Address. . . . . : 2605:a000:1240:a00d:c829:6b4f:7145:531a
Temporary IPv6 Address. . . . . : 2605:a000:1240:a00d:e5f8:3051:ced8:38fa
Link-local IPv6 Address . . . . . : fe80::c829:6b4f:7145:531a%5
IPv4 Address. . . . . : 192.168.1.10
Subnet Mask . . . . . : 255.255.224.0
Default Gateway . . . . . : fe80::429b:cdff:fe4d:5e79%5
                             192.168.1.1
```

My Home Page – <http://www.ak8b.us>

Bruce Ferry's Home Page

Nothing pretty, just useful links

- [Amateur Radio Links](#)
- [Solar and Space Weather](#)
- [Weather Links](#)
- [Misc Stuff](#)
- [Neat Stuff](#)
- [Model Train](#)
- [AK8B Amateur Radio Info Page](#)

Other Pages Hosted Here

- [AK8B Hamfest and Contest Calendar](#)
- [AK8B VE Testing Page](#)
- [DMR Codeplugs and Other Files](#)
- [The NeoGeoHams](#)
- [Cuyahoga Valley Terminal](#)
- [Trip Log](#)



Solar-Terrestrial Data	Solar-Terrestrial Data
13 Jan 2018 0309 GMT	13 Jan 2018 0309 GMT
SFI: 69 SN: 12	Calculated Conditions
A 4 K 0	Band Day Night
X-Ray: A3.3	80m-40m: Good Good
304A: 99.0 @ SEM	30m-20m: Poor Poor
Ptn Flx: 0.12	17m-15m: Poor Poor
Elc Flx: 6.11	12m-10m: Poor Poor
Aurora: 1/n=1.99	
Mag (Bz): 1.2	http://www.n0nbh.com
Solar Wind: 332.6	Copyright Paul L Herrman 2011
http://www.n0nbh.com	
Copyright Paul L Herrman 2012	

- IPv6 button to verify page supports IPv6
- My own IPv6 check mark that shows you are accessing the page through IPv6

Testing Your IPv6 – http://ipv6-test.com

The screenshot shows the IPv6 test website interface. At the top, the browser title is "IPv6 test - IPv6/4 connectivity and speed test - Mozilla Firefox" and the address bar shows "ipv6-test.com". The page has a navigation menu with "General", "Speed", "Ping", "Website", "Stats", and "API". A descriptive paragraph states: "IPv6-test.com is a free service that checks your IPv6 and IPv4 connectivity and speed. Diagnose connection problems, discover which address(es) you are currently using to browse the Internet, and what is your browser's protocol of choice when both v6 and v4 are available."

The main content area is divided into several sections:

- IPv4 connectivity:** Shows "Supported" status, IP address "71.74.241.26", hostname "cpe-71-74-241-26.neo.res.rr.com", and ISP "Spectrum".
- IPv6 connectivity:** Shows "Supported" status, address "2605:a000:1240:8032:f843:e06b:6bc1:e8b6", type "Native IPv6", SLAAC "No", ICMP "Reachable", and hostname "None".
- Score:** A progress bar indicates a score of 19 / 20.
- Browser:** Shows "Default" as "IPv6" and "Fallback" as "to IPv4 in 1 second".
- DNS:** Shows "DNS4 + IP6", "DNS6 + IP4", and "DNS6 + IP6" all as "Reachable".
- More:** Contains buttons for "Speed test »" and "Ping test »".

Addresses Assigned by IANA

- Only 2xxx:: and 3xxx:: addresses are being used for the global IPv6 internet (on earth). Early experimenting used some other addresses but they are not on the global internet.
- 2001:: addresses were used in the early allocations and are assigned to the 'early adopters' like Google and IBM.
- 2600::/12, 2610:0000::/23 and 2620:0000::/23 are assigned to ARIN (American Registry for Internet Numbers) which covers the US and Canada.

IPv6 From Your ISP

- My allocation (currently) is: 2605:a000:1240:a00d::/64
- That gives me 18,446,744,073,709,551,616 addresses of my very own. That's 4,294,967,296 times the total size of the IPv4 address space.
- A single /64 address should be big enough for most single location uses. Companies who have multiple locations can get a /48 address range. That's 1,208,925,819,614,629,174,706,176 addresses or 65,536 /64 blocks.
- My server is at: 2605:a000:1240:a00d:210:18ff:fe39:46e
- Each address is a DIRECT allocation. No NAT translation is needed. The router just passes IPv6 traffic straight through.

Where Did That Address Come From?

- The local device gets to pick their own address if your router is using SLACC+Stateless address assignment.
- For Linux and Android it is derived from the MAC address per the IEEE EUI-64 standard algorithm. My server's MAC address is: 00:10:18:39:04:6E
- My server's address:
2605:A000:1240:a00d:0210:18FF:FE39:046E
- Windows does not use this method. Why? It's part of their 'privacy extensions'.
- Google DNS servers for IPv6:
 - 2001:4860:4860::8888 and 2001:4860:4860::8844

Tweaking The Assigned IP Address in Linux

Editing System eth0

Connection name:

Connect automatically
 Available to all users

Wired | 802.1x Security | IPv4 Settings | IPv6 Settings

Device MAC address:

Cloned MAC address:

MTU: bytes

Connection Information

System eth0 (default) Auto ak8b-1ac5g

General

Interface: Ethernet (eth0)
Hardware Address: 02:00:00:00:00:22
Driver: e1000e
Speed: 1000 Mb/s
Security: None

IPv6

IP Address: 2605:a000:1240:a00d:0:ff:fe00:22/64
Default Route: fe80::429b:cdff:fe4d:5e79
▶ More addresses

Primary DNS: 2001:4860:4860::8888
Secondary DNS: 2001:4860:4860::8844

IPv6 ISP Allocation

- One of my ISP's main IPv6 allocations is: 2605:a000::/24 or 2605:a000:: through 2605:a0ff::
- That's 1,099,511,627,776 /64 blocks.
- They have TWO ADDITIONAL allocations of this size plus one smaller (/32) one.
- As part of the agreement to obtain such a large chunk of address space they **MUST** allocate /64 blocks to their customers.

The Numbers Are ...

**Just too big for most
normal human minds
to deal with!**

But Don't Worry

- You don't need to deal with the really big numbers. That's what computers are for.
- You only need to figure out how YOU can use IPv6 for your own devices.
- Your smart phone is probably already using IPv6.
- IPv4 is usually set up as a fallback protocol.
- Spectrum (Time Warner) and AT&T (U-verse) provide IPv6 now. I don't know about other ISPs but I expect most provide IPv6 by now.

What You Need to Do

- Find out if your ISP provides IPv6. It's reasonable to assume they already do.
- Does your router support IPv6? If it does, make sure you turn it on. It probably is on unless you turned it off.
- **TURN ON YOUR FIREWALLS!** I had turned mine off in Windows because NAT was protecting me. Not with IPv6!
- Turn on IPv6 in your computer(s). Unless you turned it off it probably is.

IPv6 Firewall Configuration in Linux

- One of the first things I discovered after turning on IPv6 and adding an IPv6 address to my DNS was that I instantly opened my server up to the usual attacks. You REALLY need to protect publicly exposed servers from attack.
- You must at least protect port 22 and your Samba ports. You should leave the web server and mail server (if you use one) ports open.

Continued ...

IPv6 Firewall Configuration in Linux - 2a

My IP6TABLES file:

```
# Generated by ip6tables-save v1.4.7 on Thu Dec 28 06:59:00 2017
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [894397:325869715]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p ipv6-icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 21 -j REJECT --reject-with icmp6-port-unreachable
-A INPUT -p tcp -m state --state NEW -m tcp --dport 993 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 25 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 631 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 631 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 631 -j ACCEPT
... more
```

IPv6 Firewall Configuration in Linux - 2b

My IP6TABLES file (continued):

```
-A INPUT -p udp -m state --state NEW -m udp --dport 137 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 138 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport xxx22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 110 -j REJECT --reject-with icmp6-port-unreachable
-A INPUT -p tcp -m state --state NEW -m tcp --dport 995 -j REJECT --reject-with icmp6-port-unreachable
-A INPUT -j REJECT --reject-with icmp6-adm-prohibited
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p ipv6-icmp -j ACCEPT
-A FORWARD -i lo -j ACCEPT
-A FORWARD -i eth+ -j ACCEPT
-A FORWARD -j REJECT --reject-with icmp6-adm-prohibited
COMMIT
```

IPv6 Firewall Configuration in Linux - 3

- **What I changed:**
 - I reject POP3 ports. I don't need them since I use IMAP on port 993 (secure) anyway.
 - I changed the port number for SSH from 22 to xxx22. I did this years ago.
 - I reject FTP port 21. I'll use the SSH port to get into SFTP instead.
 - Everything else is the default file.
- **The BEST guard against intrusion, as always, is to use STRONG passwords everywhere. I almost always use 12 or more characters for any password.**

About Password Strength

- Complexity doesn't matter nearly as much as length. A 12 character password with lower case only and symbols will take centuries unless you have a massive password cracking array which still takes months.
- Use several common words strung together for passwords. The password 'correcthorsebatterystaple' (25 characters) would take hundreds of billions of centuries to crack with the massive array.
- Judge the password length you need based on how important the data being protected is. The more important the longer the password. My WiFi password is 28 characters long but easy for me to remember.
- Password strength test: <https://www.grc.com/haystack.htm>
- The logic (Google 'xkcd password'): <https://xkcd.com/936/>

One Day My IPv6 Was Just Gone

- One day earlier this year I discovered that I no longer had any IPv6 access. I checked all of my devices (PC, laptop, phone, etc.), I rebooted the router, I rebooted the modem. I even tried my laptop directly on the modem. Nothing I did helped.
- I did an online chat with the Spectrum support people and they did a good job of helping troubleshoot the problem without insulting my intelligence. They gave up and had me call the tech support number.
- Calling in was a step backwards until I finally got them to escalate it up to someone who at least knew what IPv6 was.

One Day My IPv6 Was Just Gone

... continued

- After much discussion with the tech and a lot of the tech talking to an even higher level tech they told me my solution would be to switch out my modem.
- I'm sure there was some other way but swapping out the modem was probably the easy way to fix the problem.
- After swapping out the modem I got IPv6 back – complete with a new IPv6 address. My IPv4 address did not change. I suspect some router configuration change somewhere caused this situation. It could also be a firmware ‘upgrade’ in my modem that didn't work right.

Your IP May Change from Time to Time

- From Spectrum the IPv4 address seems pretty stable. Mine has changed a couple of times. The IPv6 address has changed at least twice.
- For me, this is important. I have to update the DNS servers for my domains.
- If you don't have a public server on your network then you don't really care.
- I spent some time looking for a script to keep my my DNS servers updated. After looking at the effort to set one of them up I gave up and wrote my own in Python.

Questions?