Networks “just got easy!”

Spoiler, they never changed!

Networks are just counting!

We think ipv4 with its dotted decimal notation ie 192.168.1.123 might be “clever”. Well, if we look at the numbers we are using a pool capable of representing 999.999.999.999 and constraining it do 255.255.255.255 this isn’t Decimal counting, its inherently false! Where did all the other numbers go! To elaborate, let’s look at the last “octet” well at least this is somewhat honest. It represents 8 binary symbols, so 0-255 in decimal. Presenting it this “easy” way is at best a bit patronising; we all know that after 256 comes 257! It isn’t 255 then 1.255!

Then we have the “honest” but “weird” hexadecimal notation. Just stop for a second if I place this pair of symbols up XV most children know this is 15! But in Roman numerals (symbols) so you know 2 other symbols, and I assume you know 0-9 so you have done the equivalent of 12 out of 16 or 1/3 of the extra bit – you knew it already! So you might know the Roman C and M and maybe D… you’ve already done it! And in a far less confusing way than IV or CM !!!

Hexadecimal is straight forward

Hexadecimal	Decimal	binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
a	10	1010
b	11	1011
C	12	1100
d	13	1101
e	14	1110
f	15	1111

Some might say easier than I II III IV V VI VII VIII IX X XI XII XIII XIV and XVI

Note that Roman Numerals don’t have 0!

We used to concern ourselves with subnet masks, and Classless inter-domain routing notation. That felt very clever as shorthand, and stopped expressing it as fake decimal lead outs. Another “clever” thing to show off down the pub to your mates with (said no human in a pub ever!) is being able to convert /32 to 255.255.255.255! IPv6 did a neat trick. 1. Let’s get lots of numbers (128 bits worth to be exact!) and let’s do away with some silliness – right, the smallest subnet is 64 bits – so half the address! In a 32-bit IPv4 Internet giving half of the address to a subnet seems insanely generous, and

tut tutting profligacy ensues! 128 bits represents an astronomical number! And 64 bits is gigantic! Coupled with banning broadcast messaging (finally), we are onto a winner.

Advantages – routing becomes child's play! It's a /64 network to another /64 network. This gives me 64 bits of "host addresses or 18,446,744,073,709,551,616 IP addresses in a single subnet. OK so that is a very big number to write out in decimal, hence hex – its ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff for the last host. That's a lot of F's I can hear you thinking – well, that's a lot of computers for your home (or anywhere else) network. So we know there will be lots of 0's so 1 is 0000:0000:0000:0000:0000:0000:0000:0001 – wait, I hate counting loads of 0's – well, don't we all!, maybe we don't need to. In IPv6, if there is a long block of 0's 1 is now ::1 the :: means, all the numbers are 0, let's not bother writing them out!

Why IPv6 is good

The "conventional" take back in 2009 was that ipv6 wasn't secure. This was mainly down to "scarcity" marketing. In fact IPv6 is just as (actually more) secure than IPv4. Back in those days we were in a world of NAT – Network Address Translation being thought of as an obfuscatory security shield. We have moved the fiction to VPNs being "secure" (they really aren't). security through obscurity has been at least a philosophical myth since the 19th Century (Kerkhoff). IPv6 has (apart from the "native" although less used these days, IP-Sec capability), unlike IPv4 where one only had a single "IP Address" per interface (on a normal desktop or laptop) to being able to simultaneously have multiple IP addresses. Yes you can globally route from the host, if that is your want. However, the "link local" space is huge (remember no broadcast traffic!) and an individual application could bind to its "own" ip stack, for specific tasks, whether global or local. With billions of available addresses available to a user, we can do the right thing for the task in hand.

Furthermore, routing is now quite trivial. Once the preserve of the "scarcity marketeers" routing protocols have reduced to the big one – BGP – the routing protocol that runs the Public Internet! All a routing table does is keep a table of which interface gets to another network, and has various metrics with how "far away" it is, and how fast the message could get through, when compared with other options. We can still "supernet" if we want. However this all gets handled just like the rest of the Internet. The real security consideration is up at layer 7 the application layer. Software firewalls are the guardians of this. Pretending anything useful got done below this is more bloatware. Yes – encrypt messages – a trivial thing to do with modern microprocessors, and allow the pipe to move bits. Anything that needs to make routing "hard" is getting in the way, or didn't read the memo!

The last word the curtain falls to reveal… well nothing (as it turns out)

The big pain was multicast in IPv4. It was quite rightly described as "evil". This is because, the perfect storm happened. It was "clever" – so clever it was dumb. We used to fire it off to a destination. In IPv6 it requires a source. Now instead of I want this multicast address go find it and hand it over. I now say "I want this multicast from this address. Now the "multicast router" which is just a regular ipv6 stack now goes and asks the sender (because it has an address, to hand over what is requested. The sender says yes or no. and the packets arrive. In IPv4 we had to ask a RP (rendezvous point) if it had the multicast I wanted, it would check (because we didn't necessarily know where it was coming from) and then we had to route this using PIM – protocol independent multicast – if the multicast

was actually there or not. Now we can use multicast for what it's great for – lots of listeners, and only 1 stream, this makes CDNs Content Distribution Networks, much smaller and cheaper! This is excellent news for someone who enjoys media networking!