Cutting Down on IP Address Waste

Seth Schoen
IPv4 Unicast Extensions Project

LibrePlanet 2022
March 20, 2022

IPv4 Unicast Extensions Project

- An effort to reduce waste of IPv4 addresses that are currently completely unused
- Established by John Gilmore, with technical work by Paul Wouters, Dave Täht, Seth Schoen
  - Mike Karels has also joined as co-author of one draft RFC
- Thanks to many colleagues who have offered comments and historical insights

When to plant a tree

"The best time to plant a tree was 30 years ago. The second best time is now." – Proverb

IPv4 address scarcity

- IPv4 has 32-bit addresses, so 2^{32} possibilities
  - Example: www.gnu.org = 209.51.188.116
- World population is currently about 2^{33} people
  - It was traditional to remark that most had never used the Internet, but that's changed very fast!
  - 4,294,967,296 addresses
    7,883,944,138 people (Census Bureau est.)
- Many technical limitations like this seemed unconcerning at first!

IPv4 address scarcity

- In the 1980s, it wasn’t clear that IPv4 would outcompete other network protocols, or that the Internet would outcompete other networks
  - Or that it would be worldwide or used outside of research and technology-oriented institutions
- Many early choices have had lasting impact
- In the 1990s, it became apparent that IPv4 addresses were scarce and would run out

IPv6

- This prompted development of IPv6, which has 128-bit addresses (2^{128}, about 340 undecillion)
  - Try `echo 2^128 | bc | number`
  - Example: www.gnu.org = 2001:470:142:5::116
- Finalized in 1998, then strikingly slow adoption
  - Strong in: Major Internet brands, mobile data, developing countries, Northern Europe
- Surprisingly, *most* Internet traffic is still IPv4, almost 25 years later!

Scarcity impacts

- Totally-unused new addresses ran out in 2010s
- A used IP address market emerged, especially useful for early Internet participants and hosting companies (e.g. MIT sold off its 18/8 allocation)
- The IPv4 address crunch is especially taxing for hosting companies, whose customers usually still require IPv4 addresses
  - Now said to be a measurable and growing part of the cost of hosting public Internet services!

Our proposals

- Unreserve four kinds of reserved IPv4 address, asking implementers to treat them as unicast
  - These addresses are reserved for historical reasons, to minimal or no useful purpose today
  - This will free up a substantial amount of IPv4 space, for which there is huge continued demand
- With the measurement community, test the effects of using these addresses on the Internet
  - If useful, they can be allocated some day

Historical decisions

- Throughout the 1980s—when IP’s future was less clear, and scarcity a less prominent concern—various decisions treated large numbers of addresses specially
- With decades of hindsight, some of those decisions are not helpful and are now preventing large amounts of otherwise useful address space from being used for unicast addressing

IPv4 address scarcity

- Depending on how you measure, IPv4 address space was exhausted sometime last decade
  - Most obviously, in the sense that RIRs could no longer routinely make new initial allocations to network operators
  - Address space became an important economic resource
- IPv6 was inspired largely by this looming problem, but people still want to interoperate with the IPv4 network
- Surprisingly, *enormous* amounts of space remain unallocated or unused

You might be surprised...

- Apart from organizations that received large amounts of space early on and never used it (some of which are now selling that space in the secondary markets),
- About 7% of *all* of IPv4 was reserved or given special meanings during the 1980s
  - For functions that are unnecessary in retrospect
  - But, these decisions were never reversed!

Current status

- Four Internet-Drafts proposing to unreserve addresses for unicast use
  - draft-schoen-intarea-unicast-lowest-address
  - draft-schoen-intarea-unicast-240
  - draft-schoen-intarea-unicast-0
  - draft-schoen-intarea-unicast-127
- Presented first two at IETF112 to some controversy, presenting the other two at IETF113 this Tuesday

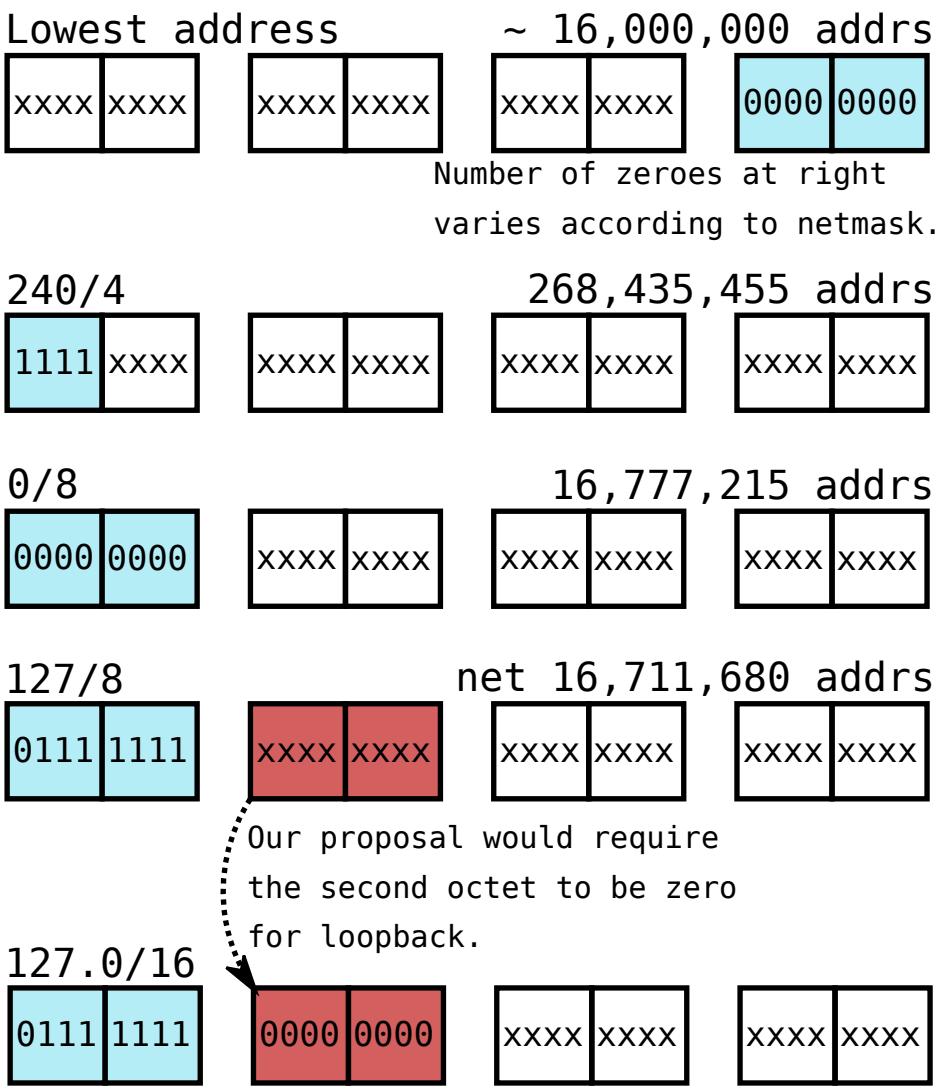
New draft on maintaining IPv4

- For discussion at IETF 113 on Tuesday, addressing the meta-issue of ongoing IPv4 work, which some people have questioned

  (draft-schoen-intarea-ietf-maintaining-ipv4)

  - An attempt to form consensus that IETF *will* continue to maintain IPv4 in the interest of its user community
  - While maintaining IETF’s policy to promote IPv6 implementation and adoption

Details!


Lowest address
~ 16,000,000 addrs
xxxx xxxx
xxxx xxxx
xxxx xxxx
0000 0000
Number of zeroes at right
varies according to netmask.
240/4
268,435,455 addrs
1111 xxxx
xxxx xxxx
xxxx xxxx
xxxx xxxx
0/8
16,777,215 addrs
0000 0000
xxxx xxxx
xxxx xxxx
xxxx xxxx
127/8
net 16,711,680 addrs
0111 1111
xxxx xxxx
xxxx xxxx
xxxx xxxx
Our proposal would require
the second octet to be zero
for loopback.
127.0/16
0111 1111
0000 0000
xxxx xxxx
xxxx xxxx

Wasted addresses: Lowest

- Suppose we have a network 42.43.44.0/24
  - Berkeley chose the *lowest address* (42.43.44.0) for broadcast
  - Developers elsewhere chose the *highest address* (42.43.44.255) for broadcast
- The highest address won out in all recommendations and documentation, but the lowest address remained reserved, explicitly for backwards compatibility

  … with systems that haven't existed for decades!

  (wasting one address per subnet, Internetwide)

Lowest address fix is local (!!)

- **Under existing RFCs**, distant (non-subnet-local) hosts must not assume the netmask of your hosts (they don’t know where subnet boundaries fall in networks to which they’re not attached)
- If just **your router and LAN** support the lowest address as unicast, the rest of the Internet should already interoperate with the lowest address on your subnet!
  - Try examples at http://ec2-reachability.amazonaws.com/

Wasted addresses: Experimental

- All the addresses from 240.0.0.0 upward (2^{28} addresses) are “reserved for future use” due to a decision in 1983
  - Futureproofing IPv4 for potential new addressing modes (e.g. dedicated anycast or encoding >32-bit addresses)
  - That was reasonable at the time, but 240/4 has *still* never been used for anything
  - New IPv4 addressing modes are very unlikely to be invented now

Wasted addresses: Zero network

- All the addresses from 0.0.0.0 to 0.255.255.255 (2^{24} addresses) are reserved due to a decision in 1981
  - Mainly intended to be used for autoconfiguration
  - But the autoconfiguration solutions that won out (BOOTP → DHCP) use only *one* of these addresses (0.0.0.0), not 2^{24}; the system that would have used all of them was deprecated in 1989

Wasted addresses: Loopback

- All of the addresses from 127.0.0.0 to 127.255.255.255 (2^{24} addresses) are reserved due to a decision in 1986.
  - All of these mean “this system”
  - By contrast, IPv6 only has the single loopback address ::1
  - It’s not common for loopback addresses outside of 127.0.0.0/16 (65536 addresses) to be used at all
    - Apparently one VPN product in Japan uses them

How many addresses?

draft-schoen-intarea-unicast-lowest-address

- “One address per subnet, Internetwide”

draft-schoen-intarea-unicast-240

- 2^{28}-1 = 268,435,455 (6.25% of all IPv4)

draft-schoen-intarea-unicast-0

- 2^{24}-1 = 16,777,215 (0.389% of all IPv4)

draft-schoen-intarea-unicast-127

- 2^{24}-2^{16} = 16,711,680 (0.389% of all IPv4)

(+unallocated 224/4 multicast: hundreds of millions?)

Software support

- **240/4** : Most popular Unix-based systems (mostly inspired by a prior proposal in 2008!)
- **Lowest address** : Linux kernel, FreeBSD
- **0/8** : Linux kernel
- **127/8** : None known
  - Changes mostly consist of identifying and *removing* special cases in IP stacks, and testing interoperability
  - Generally, no one has noticed
  - We continue to work on and propose software patches

Contributing -2 lines of code?

- I have some recollection of having proposed a software patch to change “/usr/bin” to “/bin”
  - Representing -4 bytes of code in the resulting OS
- This project also provides an opportunity to get negative lines-of-code contributions in various systems :-)
- Special cases not only in kernels, but in some networking-related userspace applications

Pseudocode

```
if (packet.destination_address >=
240.0.0.0) {
    reject(packet);
};          /* This address is too big! */


if (packet.destination_address < 1.0.0.0) {
    reject(packet);
};          /* This address is too small! */

process_packet(packet);  /* Just right! */
```

Linux kernel status

- Linux kernel has accepted:
  - 2008: A patch to implement the then-proposed behavior of unreserving 240/4 (in response to other proposals, before our project existed)
  - 2019: Dave Täht's patch to fix up aspects of the proposed behavior of unreserving 0/8
  - 2021: My patch to unreserve the lowest address in each subnet

BSD systems

- We proposed a lowest-address patch for FreeBSD; Mike Karels then wrote and merged his own version
  - This is cool because the lowest-address issue exists entirely for compatibility with historic BSDs!
- I’m currently working on fixes for 240/4 in FreeBSD and OpenBSD
  - This is very straightforward to do, although we don’t know for sure that these systems will agree to make this behavior a default

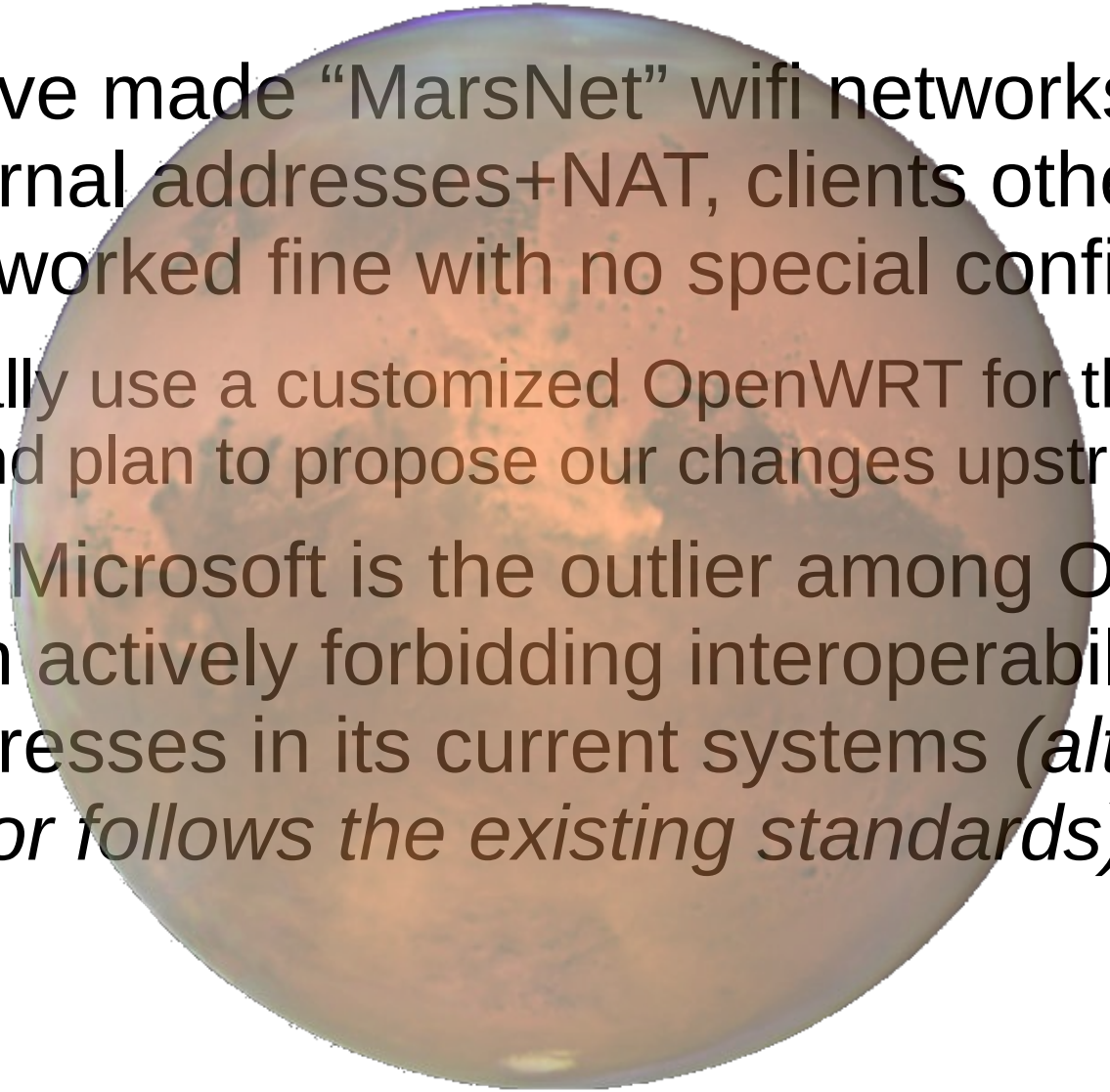
Proprietary systems

- Standardization efforts include attempts to get the proposed behavior implemented in all systems, whether free or proprietary, so they'll all interoperate
  - Free systems have been dramatically easier, because we can formulate and test the necessary changes independently, and can usually then find the right people to propose the changes to
  - I made patches for Darwin (the free kernel underlying macOS), but can't test them and am not positive where to propose them :-(
  - Microsoft has shown no interest in making any of our changes in advance of their standardization at IETF, and users don't have a straightforward path to do this without Microsoft's help

No one noticed?

- Many of the changes we propose landed in various operating systems already (through our and others' work)
  - There was no catastrophe
  - We have yet to find any complaints or bug reports
- You may be watching this presentation on a 240/4-capable device right now!

240/4 experiences



- When we've made "MarsNet" wifi networks with 240/4 internal addresses+NAT, clients other than Windows worked fine with no special configuration
  - We usually use a customized OpenWRT for the wifi router and plan to propose our changes upstream
- Currently, Microsoft is the outlier among OS vendors in actively forbidding interoperability with these addresses in its current systems *(although its behavior follows the existing standards)*

A gradual process

- Problem: if machines A and B disagree about the validity of an address, and one is numbered with that address or asked to route it, communication may not occur
- It takes time to update software
- Our changes have limited backwards compatibility (except for lowest-address), so getting widespread support in devices will take some time
- That’s why we should start in 2008 (with Fuller, Lear, and Meyer’s Internet-Draft); if not then, now!

Measurement

- We’d like to work with the Internet measurement community to get some large-scale metrics about usability of reserved addresses
- Both now and following, or as part of, Internet community consensus on trying to make reserved address space more useful
- Empirical data can inform the later decision to allocate historically reserved address space

Debogonization

- Cloudflare got official permission to use 1.1.1.1 for a DNS server, launched in 2018
- Many networks had hard-coded blocking this range. Cloudflare took > 1 year investigating users' reports of unreachability and working with ISPs to remove blocks
  - But following that, 1.1.1.1 is now extremely widely reachable on the Internet (still not 100%, but very high)
- We believe we can follow a similar process with formerly reserved addresses, once software support for them is widespread by default

Concerns

- We’ve heard a number of concerns from the community, at IETF and on network operators’ mailing lists
- There are both technical and policy concerns (like “will it work?” and “should we do this?”)
  - Like the limitations of debogonization / legacy systems
- We’ve tried to address all of these concerns; it seems that the legitimacy of continued IPv4 maintenance is the deepest disagreement

Bigger questions

- How much of the Internet is made of upgradeable systems?
- How well can we change it if there are reasons to do so?
- Are we still officially allowed to work on IPv4?
- What's the relationship between "rough consensus" and "running code"?
- What is Internet governance, anyway?

Testing

- We’ve been testing the behavior of individual operating systems and routers with regard to reserved addresses
- We’d like to start testing the use of these addresses on the Internet together with the Internet measurement community
- We anticipate that it will be years before these addresses can readily be allocated like other unicast addresses—and that they will probably still be useful at that time

How to help

- Try reserved addresses on your own testbeds and LANs
- Let us know about existing support status and uses
- Support our proposals at IETF
- Encourage vendors to support relevant address ranges
- Host measurement nodes (RIPE Atlas, ndt-server, Ark...)
  - We might be able to run reserved address space experiments with these platforms in the future *(no commitments yet)*
- **Make sure future systems you work on are readily upgradeable in case changes are needed later**

Thanks!

- Questions or comments?
- Contact us:
  - Seth Schoen <schoen@loyalty.org>
  - John Gilmore <gnu@rfc.toad.com>