ipv6 - the missed opportunities

GRNOG 18

This presentation does not reflect the views of current, past, or future employers.
The opinions in this article may or may not be constructed for contrarian effect.

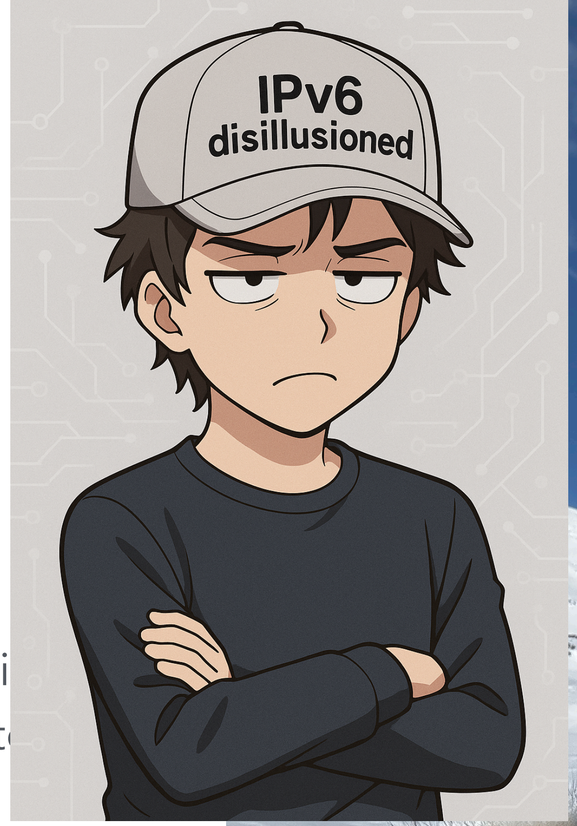
Ole Trøan, 20250620, otroan@employees.org

ipv6: what if we were wrong?

- IPv4 would run out of addresses (which it did) and we urgently needed a replacement protocol (which we didn’t).
- IPv4 is still the dominant protocol. “No-one” notices if IPv6 is disabled.
- We are 30 years in this “transition”. Is that not failure?
- Is this really still a transition, or a failed migration?
- **Current status: No-one cares. The world has moved on.**
- => Who here runs an “IPv6-only” network in production?

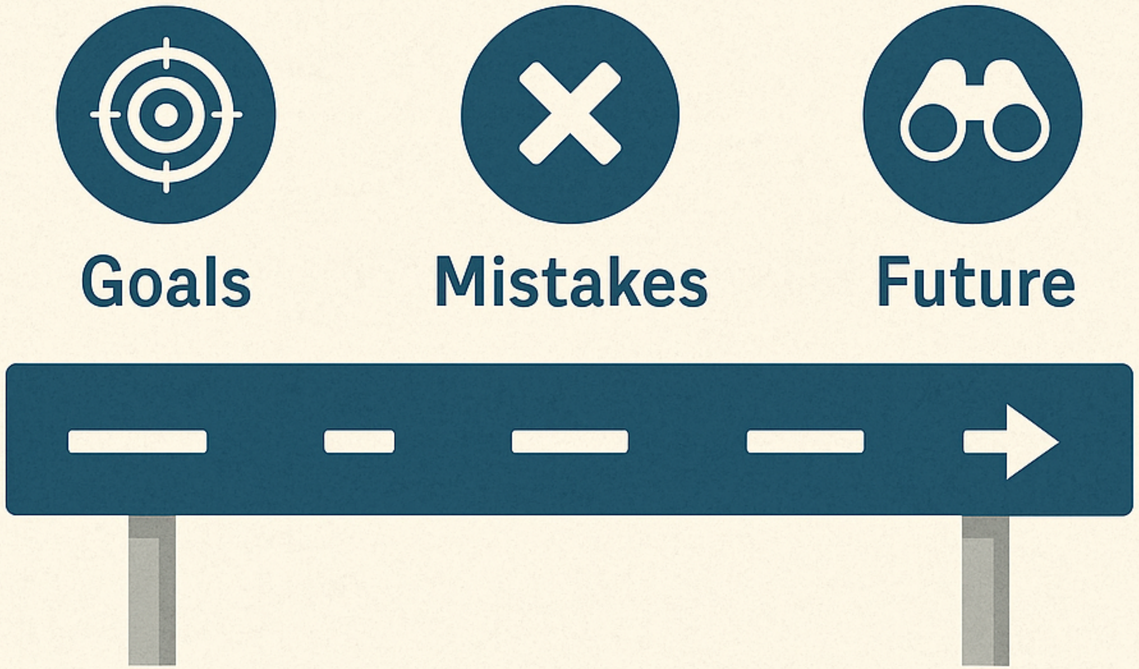
bio

Ole Trøan, PE @cisco



- IETF IPv6 WG (6man) working group chair 2012-2024
- Active contributor to tons of IPv6 transitioni and IPv4 sunsetting mechanisms. Main edit of IPv6 in CPE documents
- Worked on multiple IPv6 implementations. (cisco IOS, fd.io VPP)
- Contributor on the fd.io VPP project.
- https://ipv6.hanazo.no/posts/

AGENDA



IPv6 deployment

learning. any new protocol must:

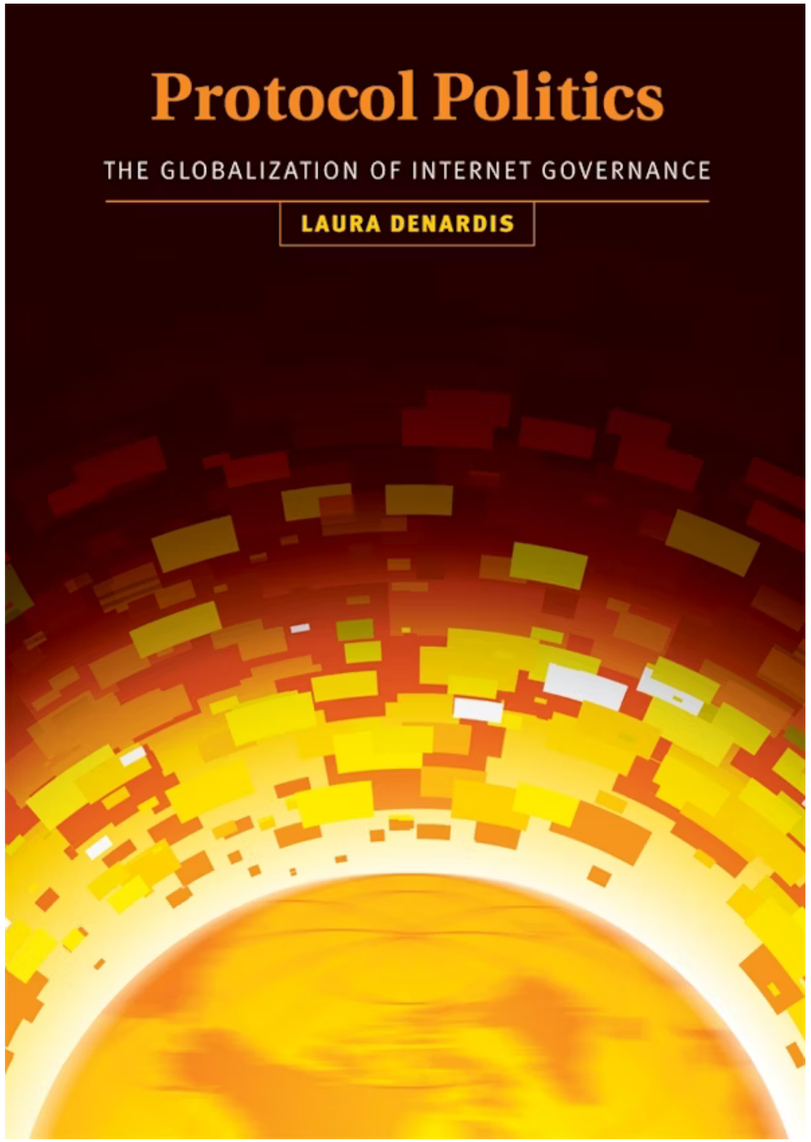
- Offer immediate benefits
- Enable low-friction transition
- Be robust to middleboxes
- Align with economic incentives
- A design supporting permission-less extensions, infinite recursion / self-similarity

If all these “mistakes” would have been fixed I don’t think it would have made a difference. There is no technical solution that would have helped IPv6 deployment.

protocol politics

…and values

- IPv6 is built as how we believed networking should be done in 1990
- Vision: IP should be like electricity, just plug it in and it should work. End to end transparency.
- Not just IPv4 with longer addresses.
  security, service discovery, multicast, mobility, renumbering. multi-homing, plug and play networks / autonomous networking
- Difficult balance:
  - Don’t change too much (e.g. upper layer protocols) because it was believed to hinder uptake.
  - Don’t change to little, because then it wouldn’t be worth transitioning.
- Anti NAT design exercise: If we prohibit ourselves from using NAT as a tool in the toolbox, how could we solve “this” set of problems?
- Turned out to be too conservative and too radical. E.g. L4 still tied to L3 addressing.



ipv6 design goals scorecard

Design Goal	Achieved?	Comments
🧮 Vast Address Space	✅	Solved address exhaustion technically, but underutilized in practice.
🚀 Simplified Header	✅	Cleaner parsing, but marginal real-world performance gain.
🧭 Routing Aggregation	❌	De-aggregation persists; BGP table growth continues.
🔐 Built-in Security (IPsec)	❌	Required in theory, but TLS dominates in practice.
📱 Mobile Support	❌	Mobile IPv6 not adopted; app-layer solutions prevail.
⚙️ Stateless Autoconfiguration	🟡	Works, but DHCPv6 coexistence creates complexity.
📡 Multicast / Anycast	❌	Rarely used; CDNs and HTTP dominate distribution methods.
♻️ Protocol Extensibility	🟡	Technically extensible, but often blocked by middleboxes.

Hosts do not trust the network
The network does not trust hosts

RFC1958: 2.3 It is also generally felt that end-to-end functions can best be realised by end-to-end protocols.

“Tussle in Cyberspace: Defining Tomorrow’s Internet”, Clark et al 2002. “hosts vs network”

ipv6 mistakes #1

- not solved: Fragmentation and MTU discovery
- 128-bit addresses (versus 64-bit addresses)
- extension headers
- multiple addresses of different scopes and properties
- conflicting configuration methods (DHCPv6 vs SLAAC)
- service Discovery
- global addressing to end-hosts.
  No addressing domain demarcation

ipv6 mistakes #2:

- no private addressing. Banned NAT. NATs are good.
  https://www.youtube.com/watch?v=v26BAIfWBm8
- didn’t solve the identifier/locator split problem
  8+8, GSE, LISP, ILNP
- L4 addressing dependent on L3 addressing
- multi-homing
- no permission-less network extensions

the consequences of !NAT

lack of identifier / locator split

- Multiple addresses with different properties (reachability, scope, “privacy”, uniqueness) per interface -> indeterministic address selection. n-square SA/DA combinations
- No working small-site multi-homing
- Address stability depending on upstream -> ephemeral addresses
- No permission-less extensions of the network
- Abstract addresses / VIPs to abstract services.
- No locator rewrite for return routeability. E.g. for “cloud services” like SSE/SASE

!NAT #2

- IPv6 applications need to deal with NAT anyway. E.g. NAT64 to interoperate with IPv4
- Need some sort of port opening protocol for firewalls
- VIP, Load balancers, Basically L2, L3, L4, L7 NATs.
- IPv6 applications will not be simpler, will need to do NAT and firewall traversal.

global endpoint addresses

- Turns out ISPs can not guarantee address stability. Nor follow the best practices for renumbering (old and new in parallel for longer than the longest session).
- Instead we got “flash” renumbering. Which none of the onlink IPv6 protocols handle well.
- Host stacks and applications struggle with multiple addresses with different reachability and properties.

extension headers

- contentious from the get go
- 3 types of containers for arbitrary other options
  - hop-by-hop
  - destination options
  - routing (segment routing, traditional source routing, mobility)

ipv6 address configuration

- Initially 64 bits to the network and 64 bits to the host. Fixed prefix length of /64.
- Interface-id evolution
- SLAAC vs DHCPv6
- DHCPv6 IA_NA => like IPv4 DHCP but no Android support (out of spite?)
- DHCPv6 IA_TA => to assign one or more temporary addresses
- DHCPv6 IA_PD => delegate a prefix across an administrative boundary
  Now being (ab)used to assign a /64 prefix to individual hosts. RFC9663.
- Fear of making DHCPv6 “complete” would allow it to replace ND/SLAAC.

transition mechanisms:

- Axis: Tunnelling versus Translation
  State: per-flow, per-subscriber, stateless (and location of state)
- IPv6 transition mechanisms:
  Automatic tunnels, 6over4, 6PE, 6to4, 6rd, ISATAP
- IPv4 sunsetting mechanisms:
  MAP-E, MAP-T, LW46, Public 4over6, 4rd, L2TPv2
- Dual-stack, ipv4 only, ipv6 only, ipv6 mostly
- ipv6 mostly gaining popularity: 464XLAT client restricted to separate address (and SLAAC)
- Lots of complexity with Happy Eyeballs, address and address-family selection. Interdeterministic.

IETF: ipv6 evolutions

- Continues to tweak and deprecate. Multiple generations of IPv6 implementations. (address selection, address generation, extension headers, …)
- Tweak SAS/DAS algorithms. Indeterministic host behaviour
- Invent even happier eyeballs. (Throw even more spaghetti on the wall to see what sticks)
- IPv6 Mostly. IPv4 only, Dual-stack, IPv6 only (with DNS64/NAT64) and 464XLAT hosts to co-exist on the same LAN.
- A /64 IPv6 prefix to every host (using DHCPv6 PD). Making IPv6 prefix assignment equivalent to IPv4 address assignment (instead of a /64 to every link).

alternative futures?

- Alternative 1: Dual-stack & IPv4/IPv6 interconnectivity forever (current trajectory)
- Alternative 2: IPv6 Only
- Alternative 3: IPv4 enhancements.
  - 32-bit TCP ports? :-)
  - IPv4E => IPv4 forever where SRv6 similar headers are used for carrying addressing domain addresses in path lists. A multi-addressing domain Internet.
  SA: {10.0.0.1, 130.67.0.1}, DA: {8.8.8.8, 144.254.0.23}
- Alternative 4: Clean(-ish) slate: SCION, multiple others?
- Alternative 5: Simpler IPv6?

alt5: run ipv6 like ipv4? :-)

- Single private address + NPTv6/NAT66
- Evolve ARP to support IPv6. Throw away ND.
- Extend DHCPv6 to include default gateway etc
- Drop extension headers
- ILNP, GSE, 8+8?

alt3: ipv4 improvements #1

- ipv4 link-locals as default gateway
- use more than port 80, 443 and 53. Endpoint Dependent NAT needs more entropy.
- 32 bit port numbers?
- IPv4E (IPv4 forever): multiple addressing domains connected with gateways. record route in extension header in forward direction and source route in reverse direction. Allows for stateless addressing domain gateways (NATs)
- NAT door knocking protocols?
- add SRv4?

conclusion:

- IPv4 is not going away.
  IPv6 is not fully replacing it.
  Do we work on optimizing dual-stack or do we look beyond it?
  Do we go back and continue to evolve IPv4?

- IPv6 has evolved to become so complex with regarding to addresses, IPv4/IPv6 interaction, and DNS interaction, the probability that all implementations will get it right is near nil.

- **Best ROI is probably to be the last one to transition from IPv4 to IPv6.**
  In the IPv6 race, last place might just be the smartest move.

- Question IPv6 dogma!

- Don’t let ideology block progress


MAKE IPv6
OPTIONAL AGAIN