Network Layer: IPv6

CS 204: Advanced Computer Networks

Oct 30, 2023

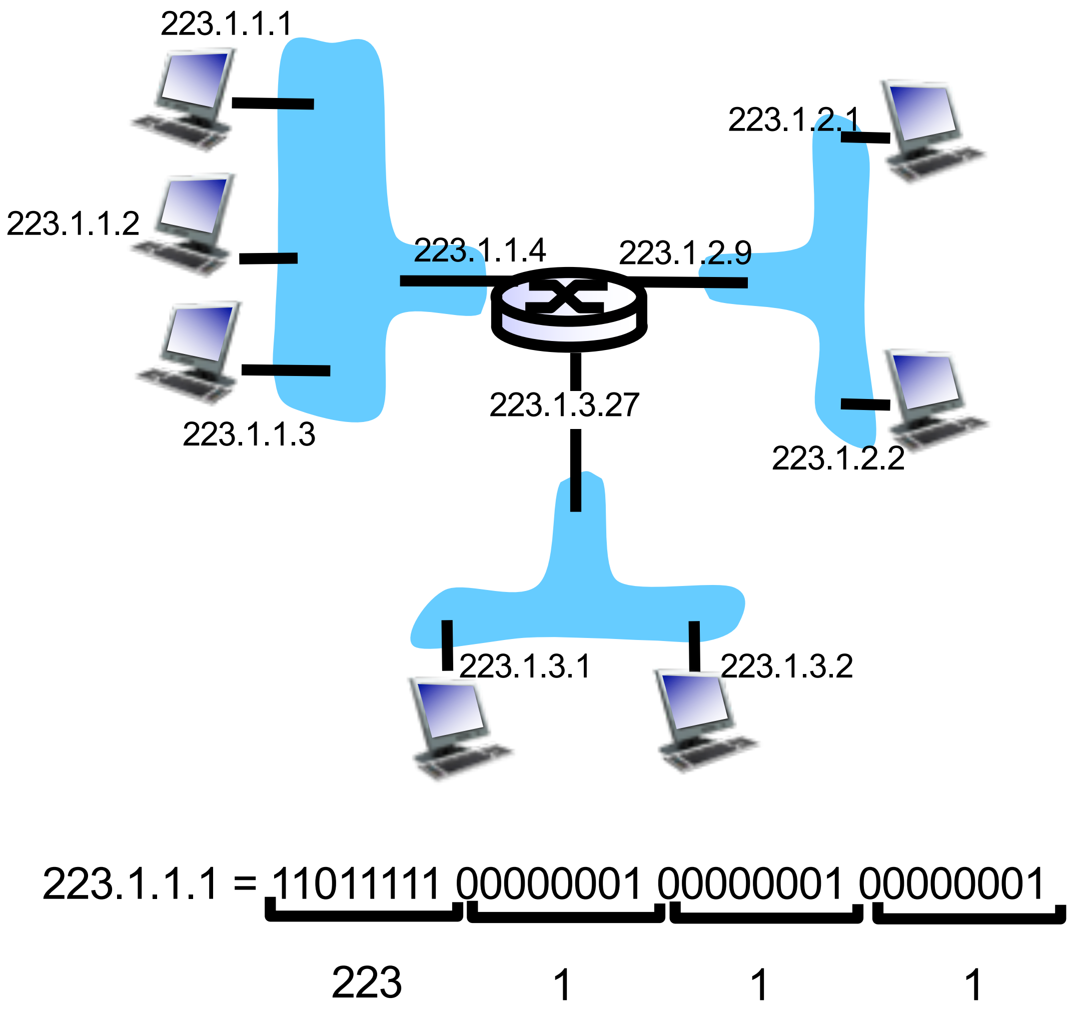
Outline

- From IPv4 to IPv6
- Techniques for IPv6
- Adoption

Q: Why we need IPv6?

IPv4 addressing

- *IP address:* 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- *IP addresses associated with each interface*



Issues with IPv4

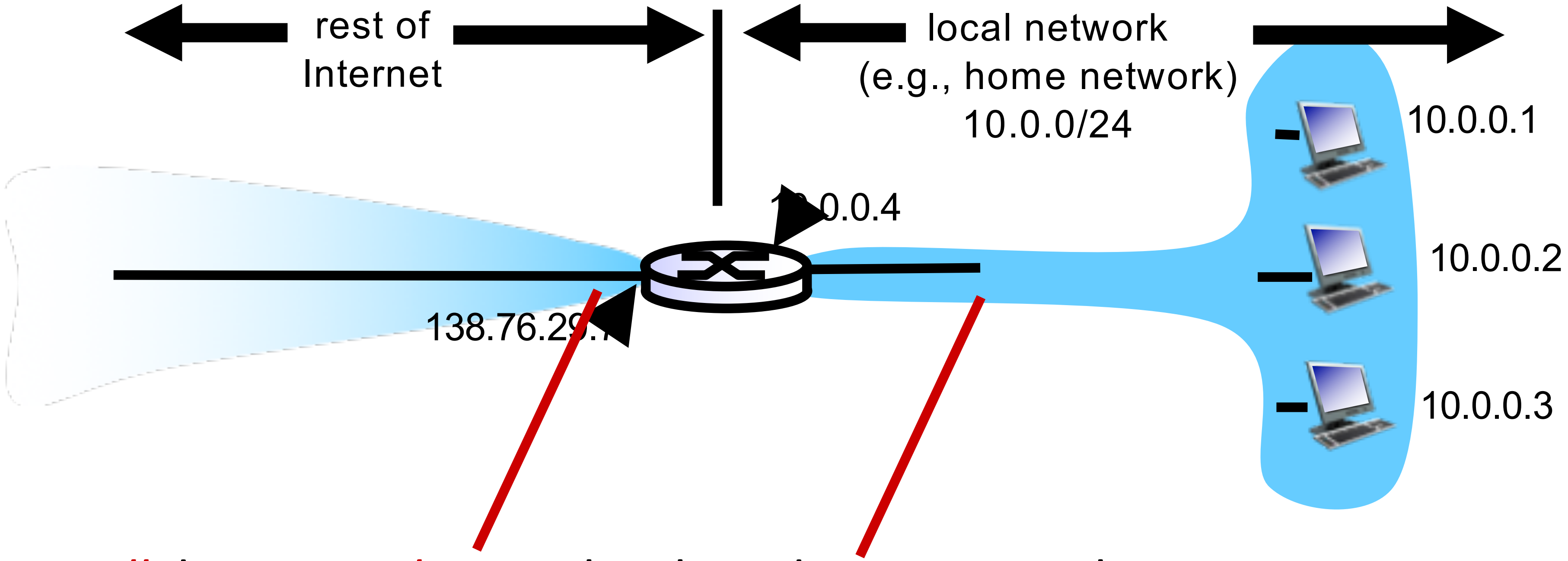
- 32-bit address space soon to be completely allocated
  - Already several address exhaustion milestones in early 2010s
  - Internet Assigned Numbers Authority (IANA), as well as two of its five subordinate regional Internet registries (RIRs) either completely exhausted address space or resorted to rationing their final address block
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS

One possible solution: NAT

motivation: local network uses just one IP address as far as outside world is concerned:

- range of addresses not needed from ISP: just one IP address for all devices
- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly addressable, visible by outside world (a security plus)
  - Private IP addresses used locally
  - Carrier-grade NAT addresses

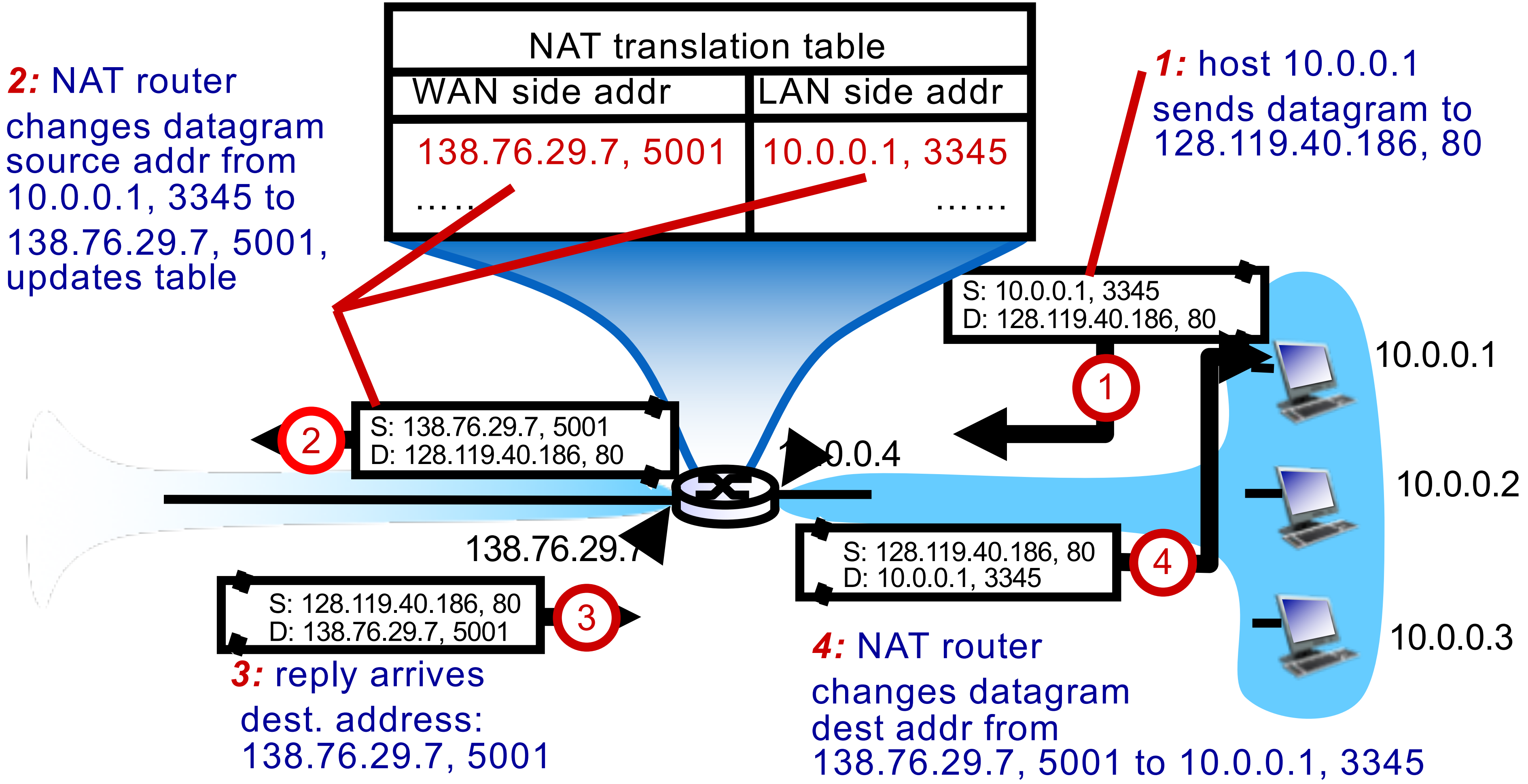
NAT: Network Address Translation



all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7,different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: Network Address Translation



NAT: Network Address Translation

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - address shortage should be solved by IPv6
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P
  - NAT traversal: what if client wants to connect to server behind NAT?
- but NAT is here to stay:
  - extensively used in home and institutional nets, 4G/5G cellular nets

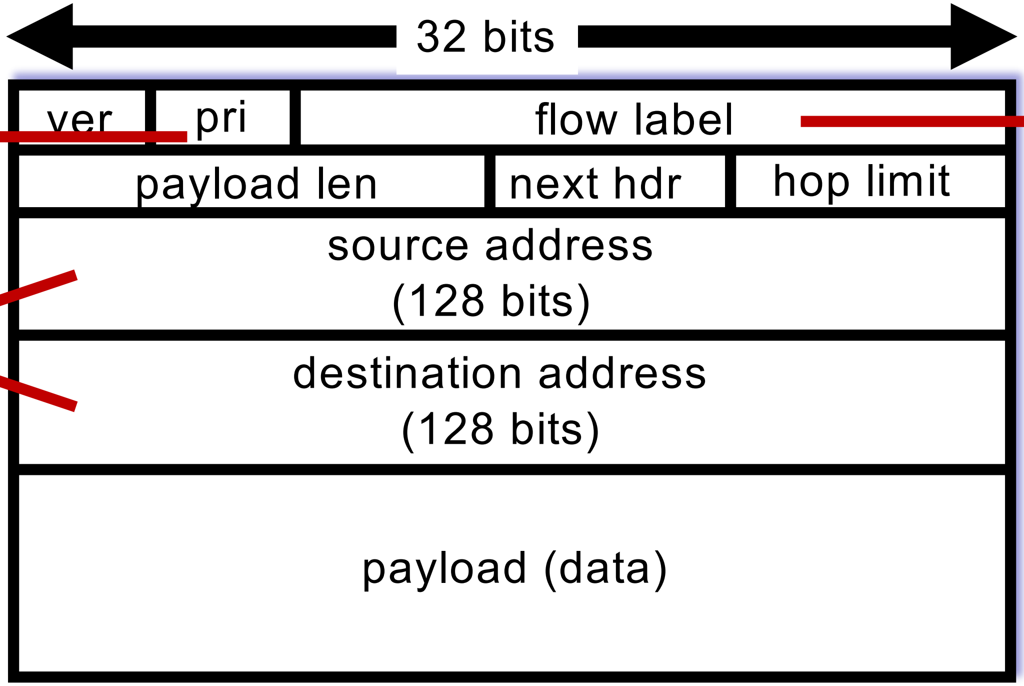
IPv6

- IPv6: 128 bit addresses
  - fixed-length 40 byte header
  - enable different network-layer treatment of “flows”

IPv6 Datagram Format

priority: identify priority among datagrams in flow

128-bit IPv6 addresses



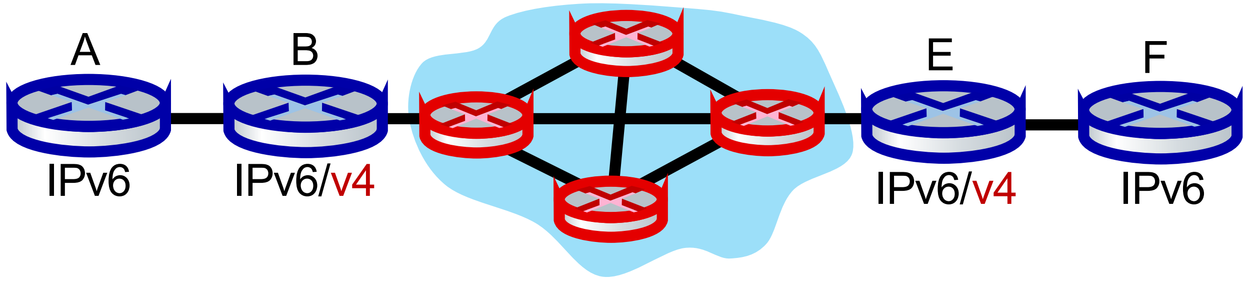
flow label: identify datagrams in same "flow." (concept of "flow" not well defined).

What's missing (compared with IPv4):

- no checksum (to speed processing at routers)
- no fragmentation/reassembly
- no options (available as upper-layer, next-header protocol at router)

Challenges to adopt IPv6

- High overhead to transit all the network nodes
  - Some will use IPv4, some will use IPv6
  - How to ensure communication such a mixed of v4 and v6?



Outline

- From IPv4 to IPv6
- Transition
- Adoption

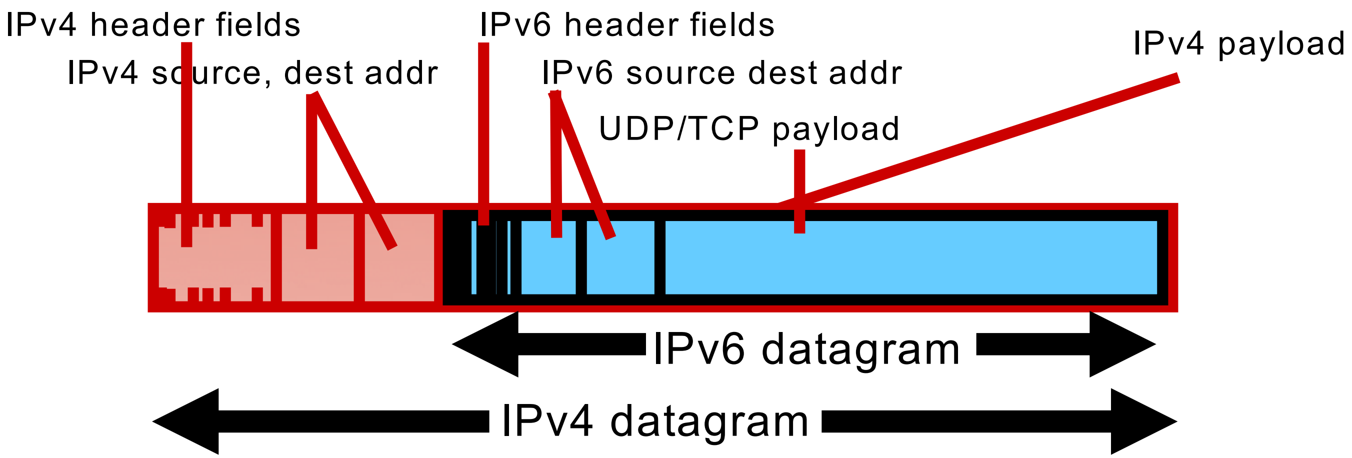
Q: What're the technical challenges to enable IPv6?

Transition from IPv4 to v6

- Not all hosts or routers can be upgraded simultaneously
  - No “flag days”
  - How will network operate with mixed IPv4 and IPv6 routers?
- Three categories of techniques in general
  - Tunneling
  - Translation
  - Dual-Stack

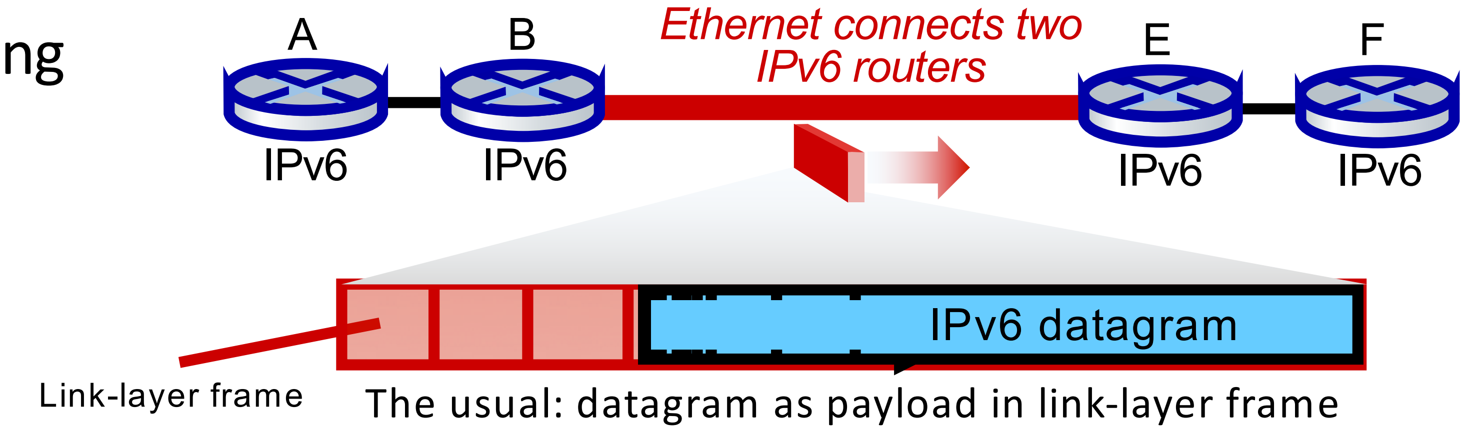
Tunneling for IPv6

- tunneling: IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers ("packet within a packet")
  - tunneling used extensively in other contexts (4G/5G)

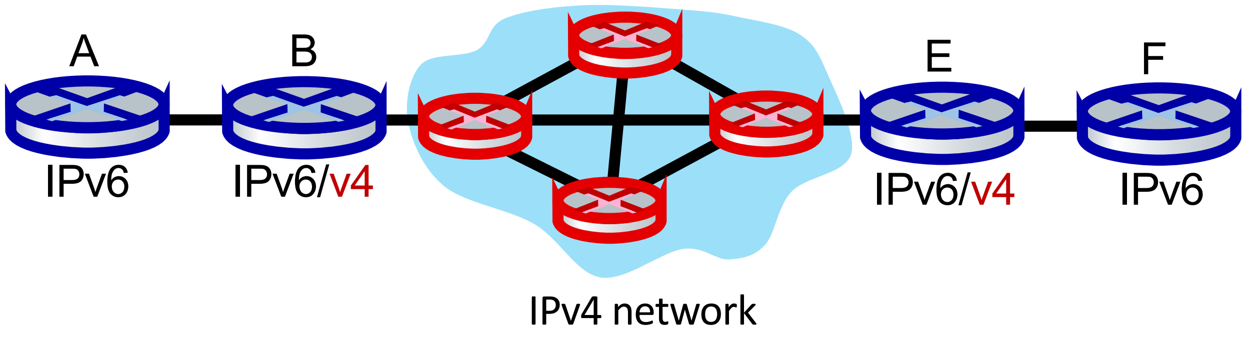


Tunneling and encapsulation

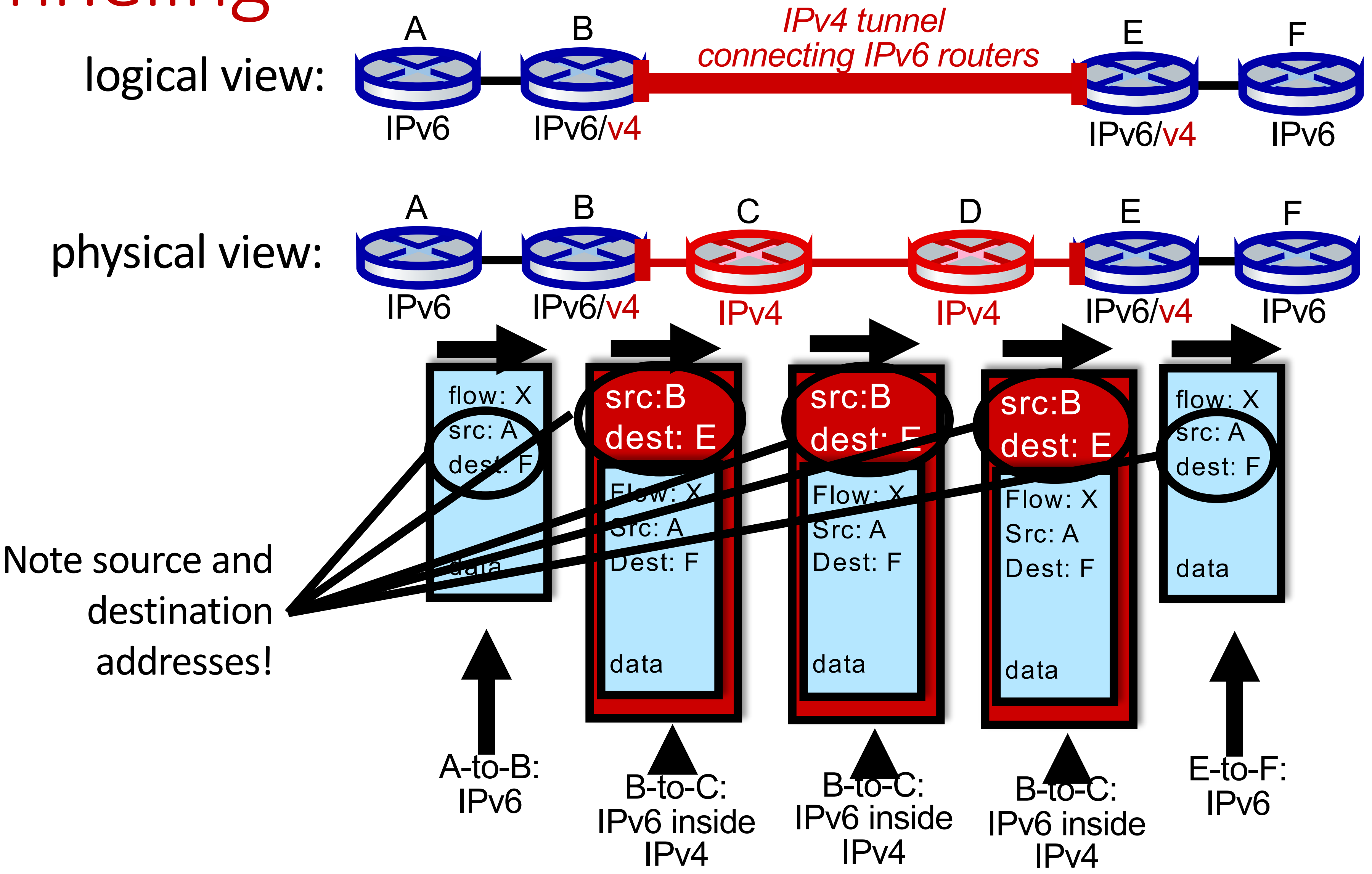
Ethernet connecting two IPv6 routers:



IPv4 network connecting two IPv6 routers

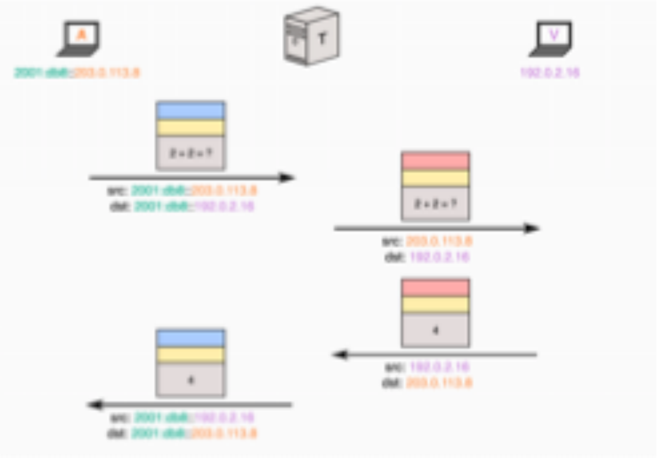


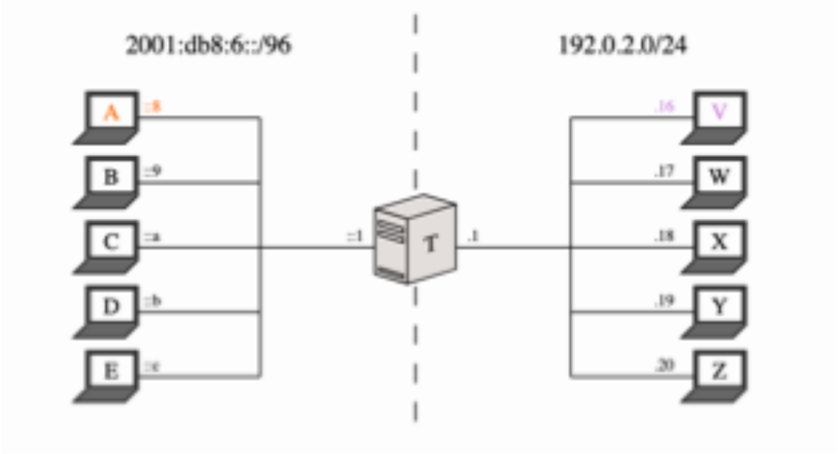
Tunneling


logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6/v4
IPv6/v4
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6/v4
IPv4
IPv4
IPv6/v4
IPv6
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
Note source and
destination
addresses!
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

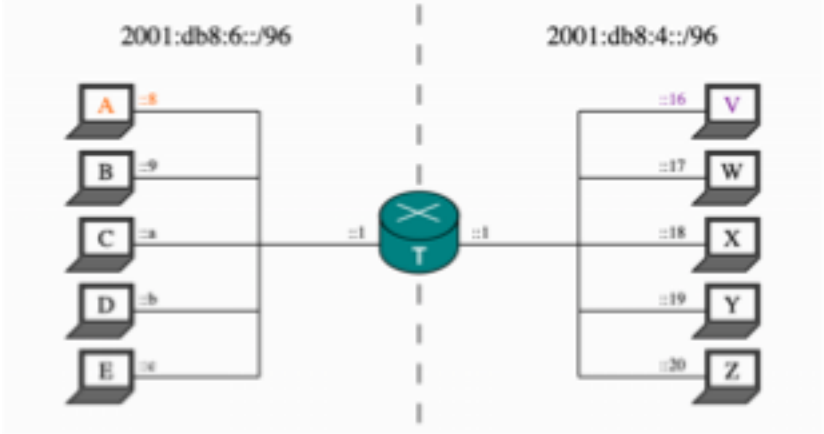
Translation: Stateless IP/ICMP Translation (SIIT)

- A translation algorithm maps v6 and v4 addresses
  - Traditionally, add/remove IPv6 header
  - Preconfigured static address translation mechanism
    - Explicit Address Mapping (EAM)
  - Often used in data centers



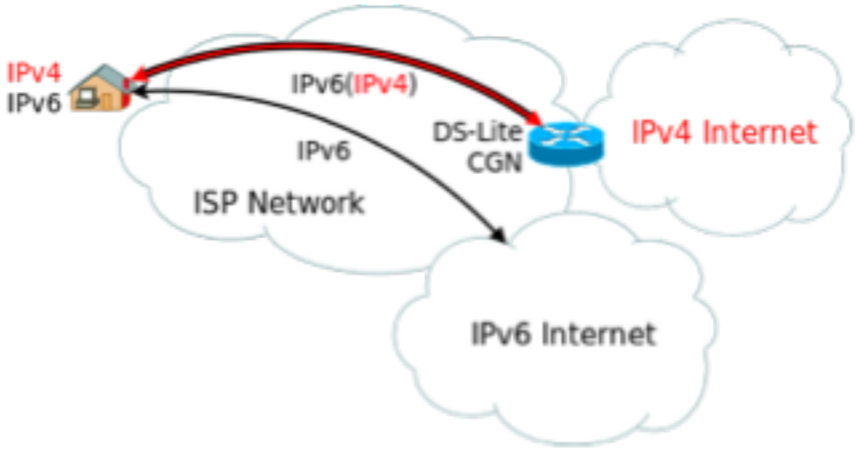


IPv6	IPv4
2001:db8:6::8	203.0.113.8
2001:db8:6::9	203.0.113.9
2001:db8:6::a	203.0.113.10
2001:db8:6::b	203.0.113.11
2001:db8:6::c	203.0.113.12
2001:db8:4::16	192.0.2.16
2001:db8:4::17	192.0.2.17
2001:db8:4::18	192.0.2.18
2001:db8:4::19	192.0.2.19
2001:db8:4::20	192.0.2.20



Dual Stack

- A node could possess both IPv4 and IPv6 interfaces
  - Use DNS to decide whether an IPv4 or IPv6 packet should be sent
  - DNS AAAA Record -> v6, DNS A Record -> v4



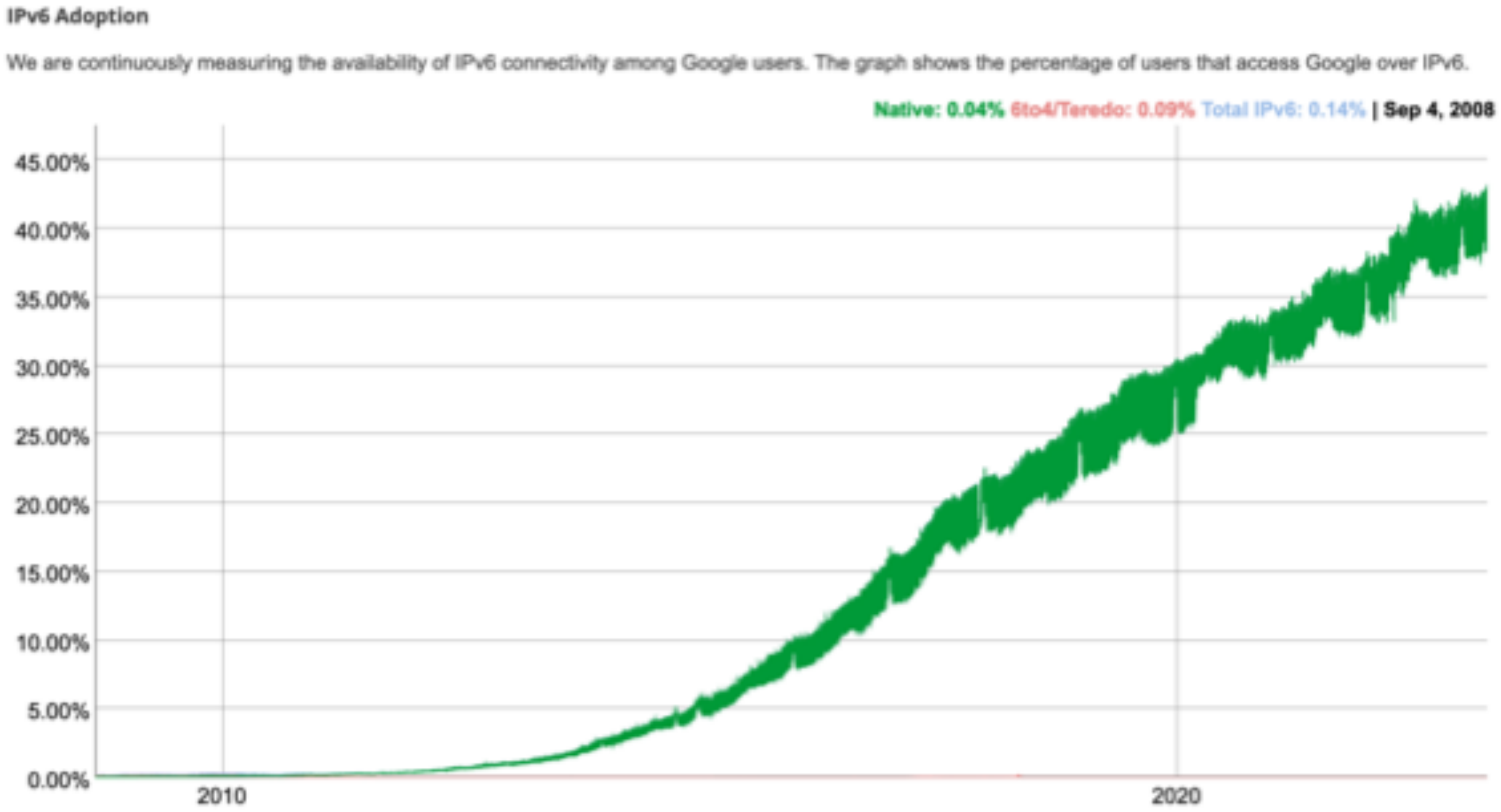
Outline

- From IPv4 to IPv6
- Transition
- Adoption

Q: How well has IPv6 been adopted in today's Internet?

IPv6: adoption

- Google[1]: ~ 40% of clients access services via IPv6 (2023)
- NIST: 1/3 of all US government domains are IPv6 capable



IPv6: adoption

- Google[1]: ~ 40% of clients access services via IPv6 (2023)
- NIST: 1/3 of all US government domains are IPv6 capable
- Long (long!) time for deployment, use
  - 25 years and counting!
  - think of application-level changes in last 25 years: WWW, social media, streaming media, gaming, telepresence, …
  - *Why?*

[1] https://www.google.com/intl/en/ipv6/statistics.html

IPv6: Topology Discovery

- Understanding IPv6 topology is important to
  - Optimize the content distribution and traffic optimization
  - Better address anonymization and reputation
  - Enhance network security
- However, there are two major challenges
  - What to probe: Massive address space that is sparsely populated
  - How to send probes? Mandated ICMPv6 rate limiting

What to probe

- Conventional approaches: Mimic IPv4 probing techniques
  - For each IPv6 prefix in global BGP table, sequentially traceroute to: ::1 in prefix random address in prefix
- Issue: Miss subnetting and other topological structure
  - Breadth, no depth!
- Insights from the “hitlists” (collections of known IPv6 hosts)
  - Targets in some hitlists concentrated in small number of prefixes / Ases
  - Need new approach to find out the structure

Target Generation with Seeding


```
2607:5300::1029                    2607:5300::/64                2607:5300:::1234:5678
2607:5300::109f
2607:5300::102a
2a07:18e8:4005:80b:e3ae::200e      2a07:18e8:4005:80b::/64   →   2a07:18e8:4005:80b::1234:5678
2a07:18e8:4005:80b:87e8::400a
```

- Begin with seeds: hitlist addresses
- zn aggregation: Group addresses into prefixes of length n
- Targets are synthesized with interface identifier

How to Probe

- Existing probing methods
  - “Sequential” (i.e. TTL=1,2,...)
  - Limited parallelism (i.e. waiting for responses, window of destinations)
  - Probing faster can be self-defeating: triggers more rate-limiting
- How to probe in IPv6 to minimize effect of rate-limiting, while maintaining complete probing?

Probe using Yarrp

- Yarrp: “Yelling at Random Routers Progressively” (IMC2016)
  - Uses a block cipher to randomly permute the hIP, TTLi domain
  - Is stateless, recovering necessary information from replies
  - By randomly spreading probes in time/space, permits fast Internet-scale active topology probing
- Yarrp6
  - Add IPv6-specific enhancements
  - Hypothesis: Yarrp-mapping of the IPv6 Internet will suffer less rate-limiting, even at higher probing rates

Some issues with Yarrp

- Yarrp is stateless
  - Must select TTL range (maxTTL) (potentially missing hops)
  - Don’t know when to stop probing (potentially wasting probes)

- Solution:
  - For response to probe with TTL=h, immediately probe with TTL=h + 1 if h >= maxTTL

Results

- Settings
  - Single runs: May 14, 2018
  - 3 vantage points: 2 US Universities; 1 EU Network
  - 18 different target sets
  - Yarrp6 w/ TTL=16 and fillmode
  - ICMPv6 probes 2kpps
- Results
  - 45.8M traces to 12.5M destinations (in less than a day)
  - Discover 1.4M IPv6 router addresses
  - Order of magnitude more than prior efforts

Findings

Unanticipated Result

- EUI64 embeds a device's H/W MAC into its IPv6 address
- For privacy reasons, most OSes use ephemeral random addresses instead
- Surprisingly, across 45.8M traces, discover 651.4k EUI64 addresses (45% of all addresses!)

Implications to Security and Privacy (RFC7721)

- Primarily at the end of the path (CPE!)
- Concentrated among providers and manufacturers
- Working with community to address
- (E.g., next week at IETF maprg WG)