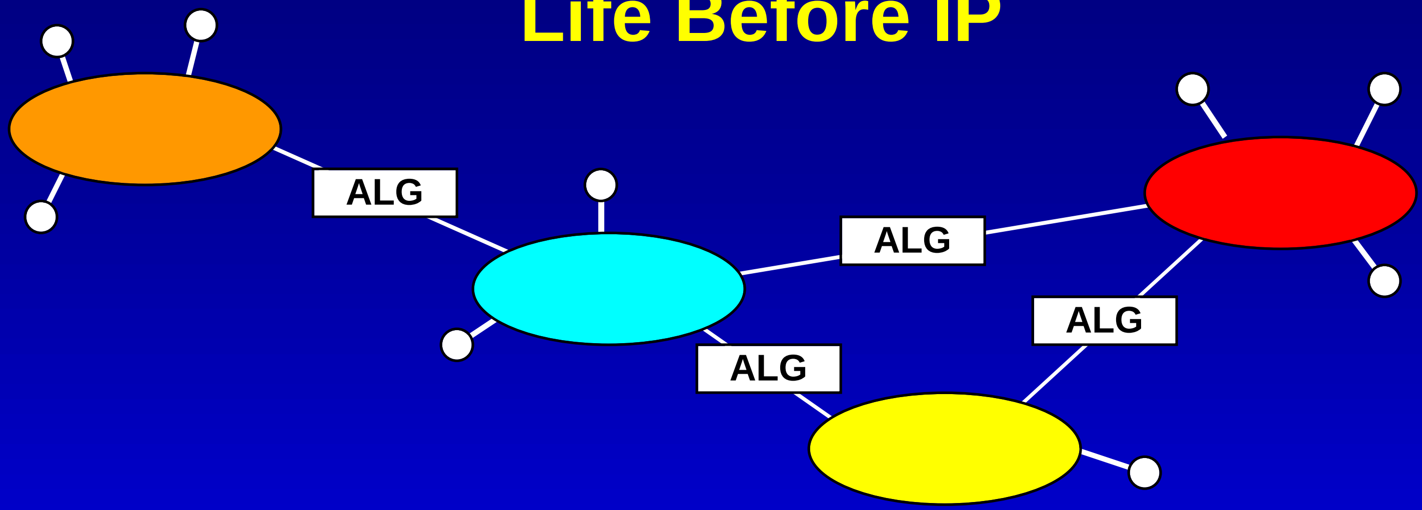
IPv6:
Why, What, When, How?

Steve Deering
deering@cisco.com

February 25, 2000

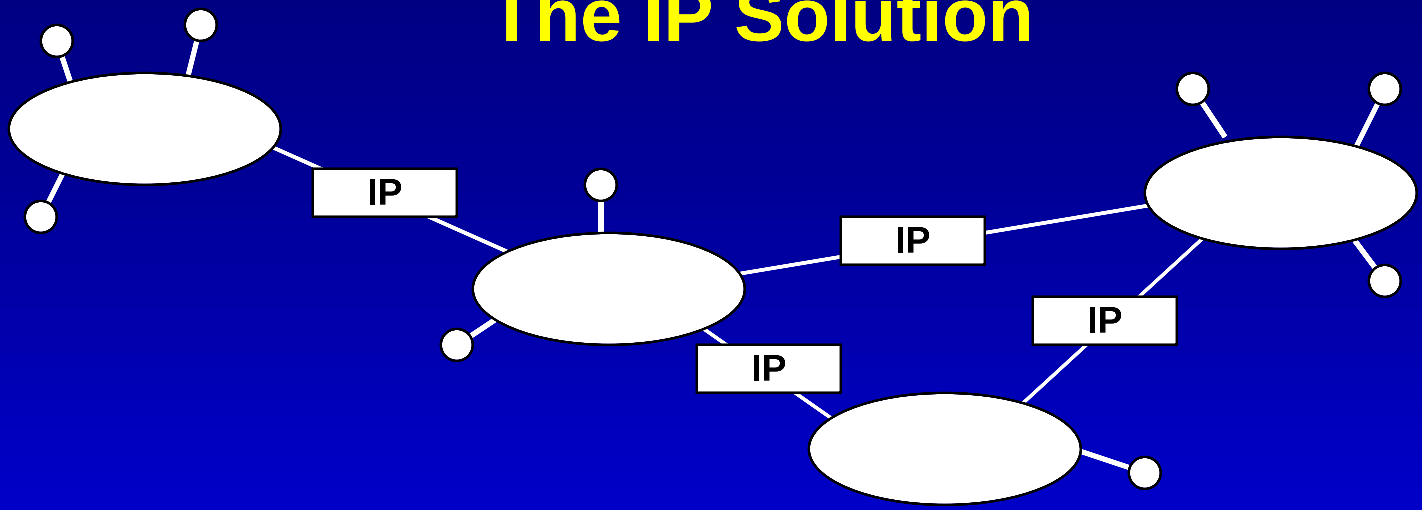
Why?

Life Before IP



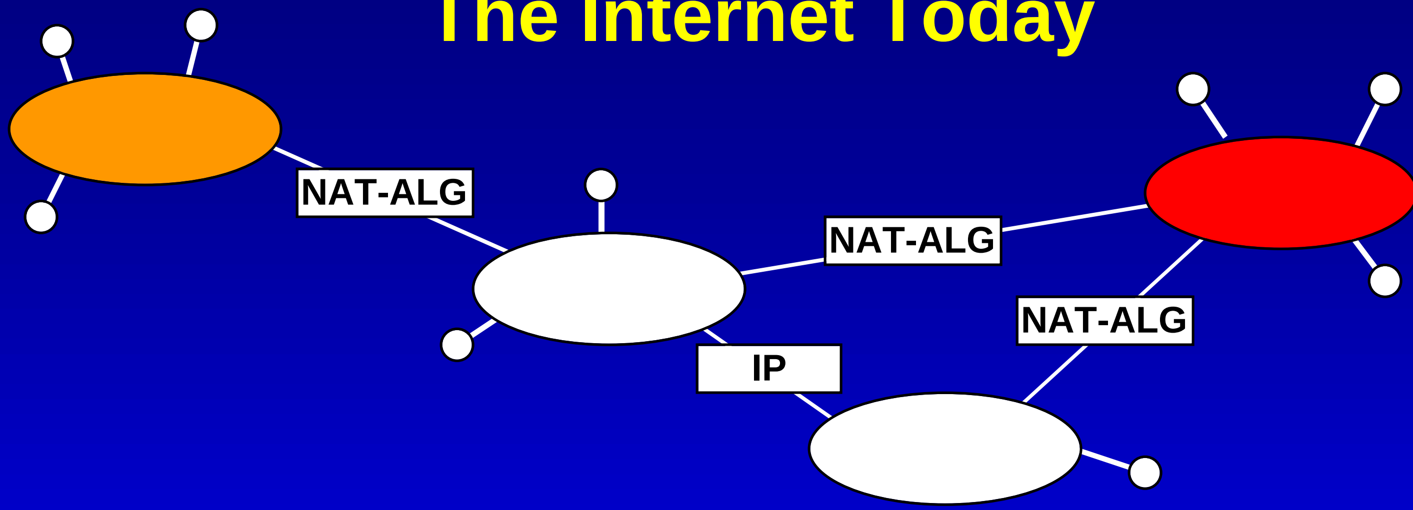
- application-layer gateways
  - inevitable loss of some semantics
  - difficult to deploy new internet-wide applications
  - hard to diagnose and remedy end-to-end problems
  - stateful gateways inhibited dynamic routing around failures
- no global addressability
  - ad-hoc, application-specific solutions

The IP Solution



- internet-layer gateways & global addresses
  - simple, application-independent, least-common-denominator network service: best-effort datagrams (i.e., packet switching)
  - stateless gateways could easily route around failures
  - with application-specific knowledge out of the gateways:
    - NSPs no longer had monopoly on providing new services
    - Internet became a platform for rapid, competitive innovation

The Internet Today



- NATs and application-layer gateways
  - inevitable loss of some semantics
  - difficult to deploy new internet-wide applications
  - hard to diagnose and remedy end-to-end problems
  - stateful gateways inhibit dynamic routing around failures
- no global addressability
  - ad-hoc, application-specific solutions (or worse!)

But Isn't There Still Lots of IPv4 Address Space Left?

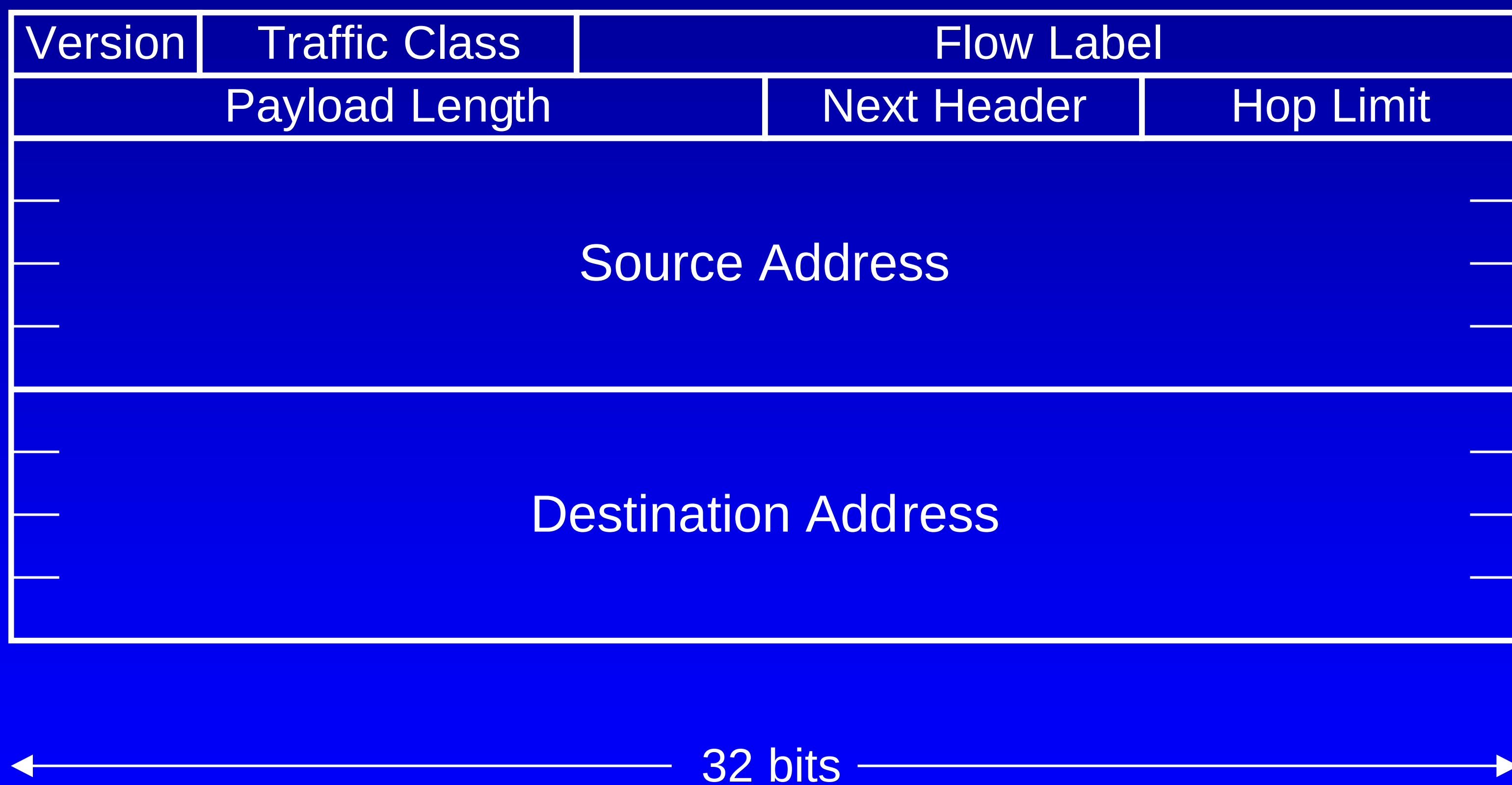
- approx. half the IPv4 space is unallocated today
  – how long does it take for the number of IP devices to double?
- IPv4 addresses are effectively being rationed
  => consumption statistics tell us nothing about the real demand for addresses, or the hardship created by wihholding them
  – the difficulty in obtaining addresses is why many (most?) of the NAT-ALGs exist
- new kinds of Internet devices will be much more numerous, and not adequately handled by NATs (e.g., mobile phones, cars, residential servers, ...)

But Can't We Just Make the NATs Better?

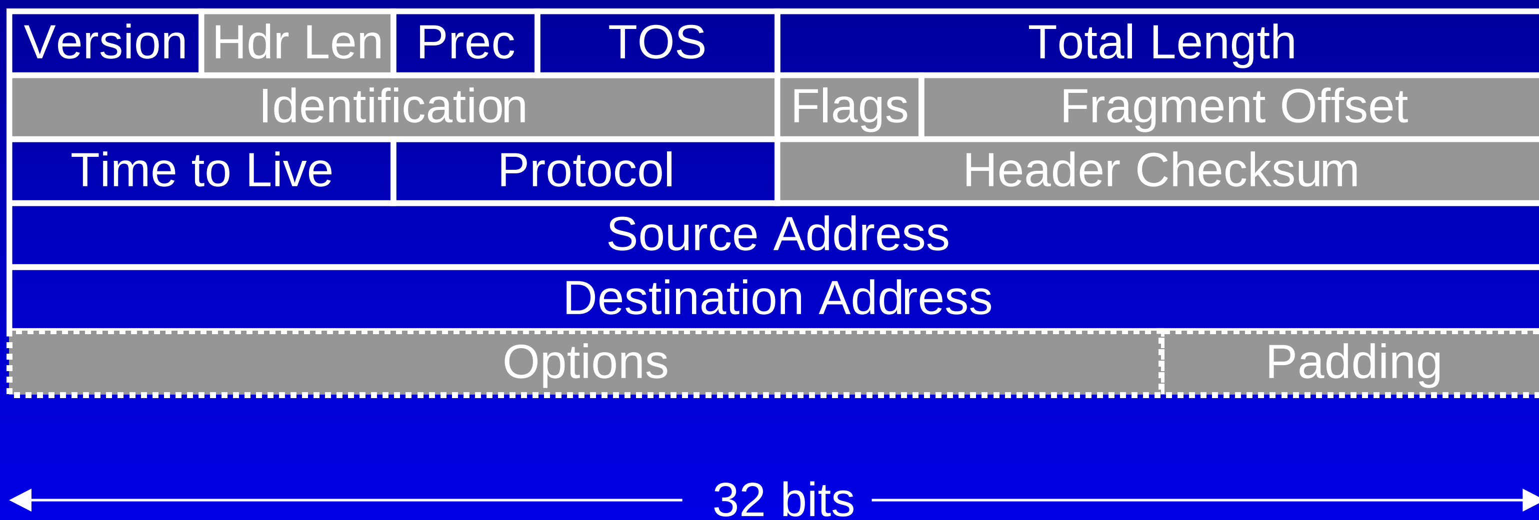
- we could keep adding more protocols and features to try to alleviate some of their shortcomings
  - might improve their functionality, but will increase their complexity, fragility, obscurity, unmanagability,...
  - new problems will arise when we start needing inter-ISP NAT
- doing one big thing (moving to IPv6) will avoid the need to continue doing many big and small things to keep the Internet working and growing
- (no, IPv6 is not the only possible solution, but the most mature, feasible, and widely agreed-upon one)

What?

The IPv6 Header



The IPv4 Header

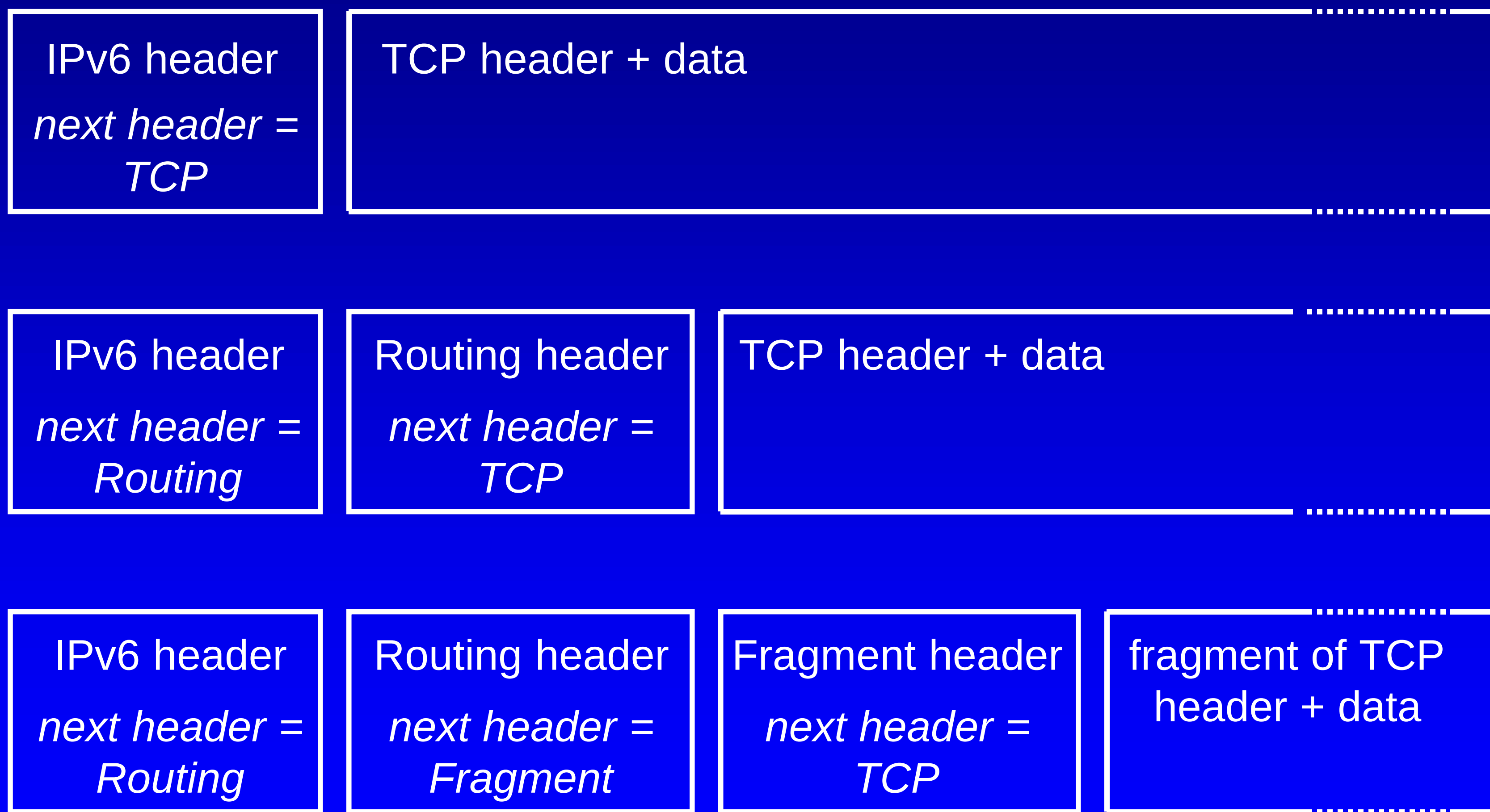


shaded fields are absent from IPv6 header

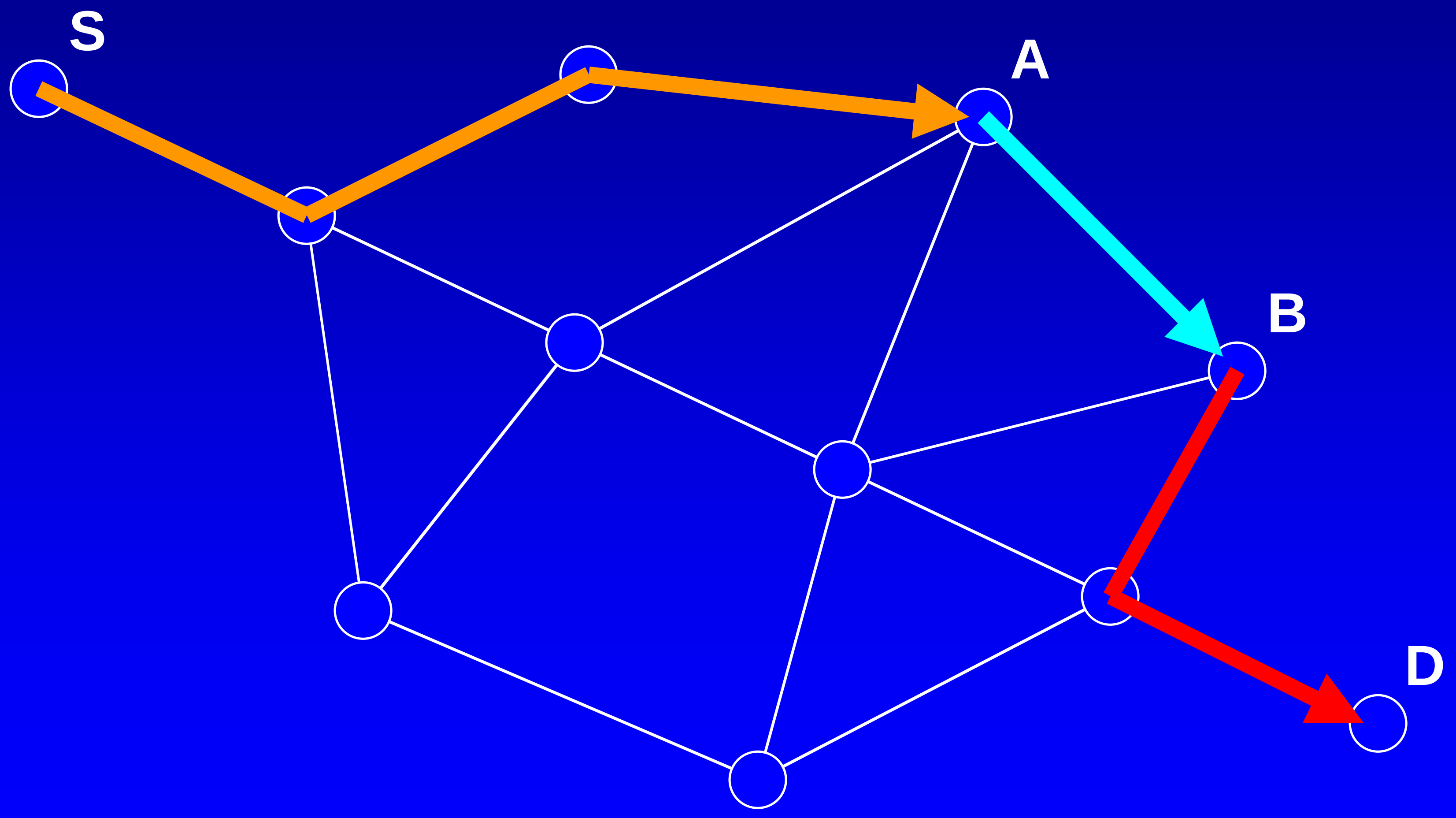
Header Changes from IPv4

- addresses increased from 32 to 128 bits
- fragmentation fields moved out of base header
- IP options moved out of base header
- Header Checksum eliminated
- Header Length field eliminated
- Flow Label field added
- Time to Live —> Hop Limit
- Protocol —> Next Header
- Precedence & TOS —> Traffic Class
- Length field excludes IPv6 header
- alignment changed from 32 to 64 bits

Extension Headers



Example of Using the Routing Header



Extension Headers (cont.)

- processed only by node identified in IPv6 Destination Address field => much lower overhead than IPv4 options
  - exception: Hop-by-Hop Options header
- eliminated IPv4's 40-octet limit on options
  - in IPv6, limit is total packet size, or Path MTU in some cases
- currently defined extension headers:
  - Hop-by-Hop Options, Routing, Fragment, Authentication, Encryption, Destination Options

MTU Issues

- minimum link MTU for IPv6 is 1280 octets (versus 68 octets for IPv4)
  - => on links with MTU < 1280, link-specific fragmentation and reassembly must be used
- implementations are expected to perform path MTU discovery to send packets bigger than 1280
- minimal implementation can omit PMTU discovery as long as all packets kept $\leq$ 1280 octets
- a Hop-by-Hop Option supports transmission of "jumbograms" with up to 2^{32} octets of payload

Fragment Header

Next Header	Next Header	Fragment Offset	0 0 M
Original Packet Identifier			

- though discouraged, can use IPv6 Fragment header to support upper layers that do not (yet) do path MTU discovery
- IPv6 frag. & reasm. is an **end-to-end** function; routers do **not** fragment packets en-route if too big—they send ICMP “packet too big” instead

Text Representation of Addresses

“preferred” form:	1080:0:FF:0:8:800:200C:417A
compressed form:	FF01:0:0:0:0:0:0:43 becomes FF01::43
IPv4-compatible:	0:0:0:0:0:0:13.1.68.3 or ::13.1.68.3

Address Types

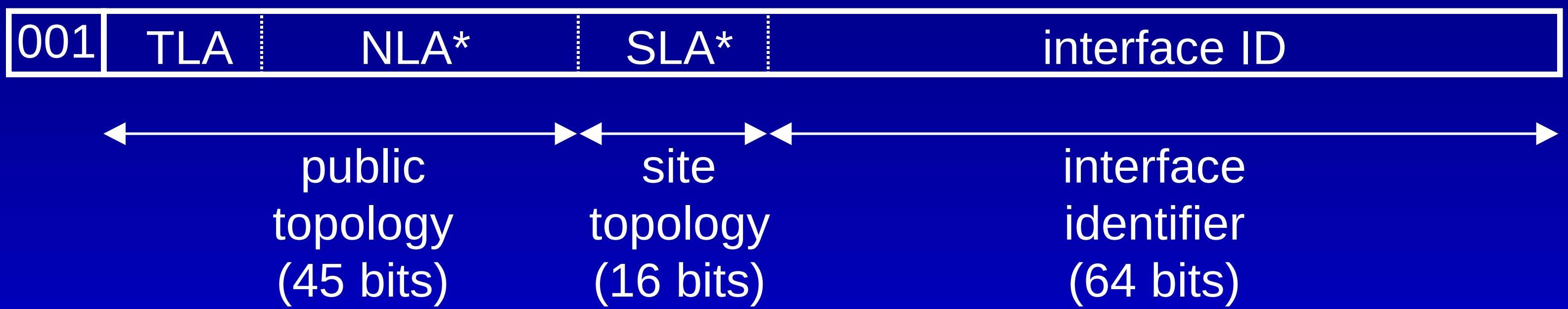
- unicast (one-to-one)
  - global
  - link-local
  - site-local
  - compatible (IPv4, IPX, NSAP)
- multicast (one-to-many)
- anycast (one-to-nearest)
- reserved

Address Type Prefixes

address type	**binary prefix**
IPv4-compatible	0000...0 (96 zero bits)
global unicast	001
link-local unicast	1111 1110 10
site-local unicast	1111 1110 11
multicast	1111 1111

- all other prefixes reserved (approx. 7/8ths of total)
- anycast addresses allocated from unicast prefixes

Global Unicast Addresses



- TLA = Top-Level Aggregator
  NLA* = Next-Level Aggregator(s)
  SLA* = Site-Level Aggregator(s)
- all subfields variable-length, non-self-encoding (like CIDR)
- TLAs may be assigned to providers or exchanges

Link-Local & Site-Local Unicast Addresses

link-local addresses for use during auto-configuration and when no routers are present:

1111111010	0	interface ID

site-local addresses for independence from changes of TLA / NLA*:

1111111011	0	SLA*	interface ID

Multicast Addresses

11111111	flags	scope	group ID
8	4	4	112 bits

- low-order flag indicates permanent / transient group; three other flags reserved
- scope field:
  1 - node local
  2 - link-local
  5 - site-local
  8 - organization-local
  B - community-local
  E - global

  (all other values reserved)

Routing

- uses same “longest-prefix match” routing as IPv4 CIDR
- straightforward changes to existing IPv4 routing protocols to handle bigger addresses

  unicast: OSPF, RIP-II, IS-IS, BGP4+, …

  multicast: MOSPF, PIM, …
- can use Routing header with anycast addresses to route packets through particular regions

  e.g., for provider selection, policy, performance, etc.

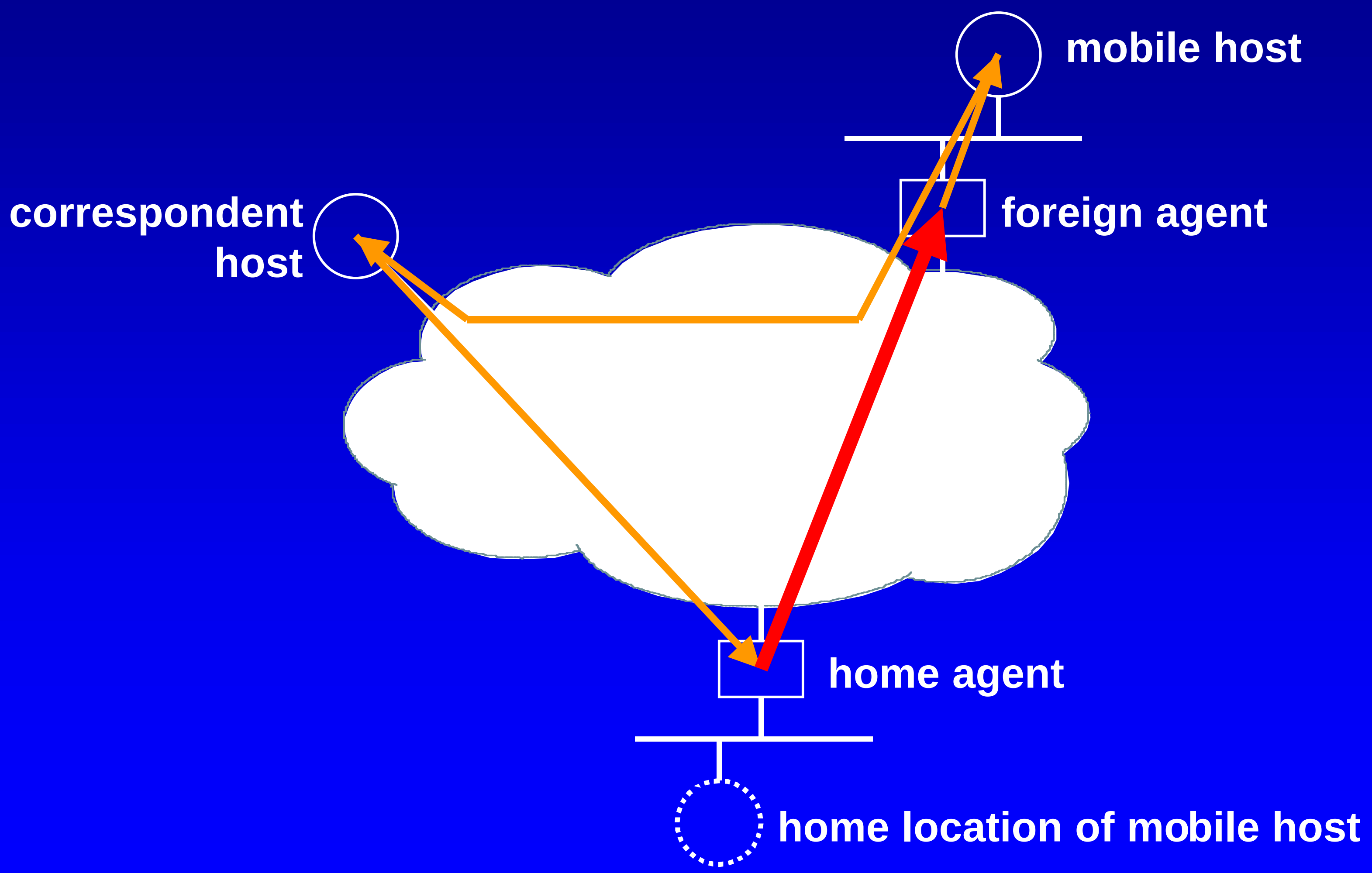
Serverless Autoconfiguration ("Plug-n-Play")

- hosts can construct their own addresses:
  - subnet prefix(es) learned from periodic multicast advertisements from neighboring router(s)
  - interface IDs generated locally, e.g., using MAC addresses
- other IP-layer parameters also learned from router adverts (e.g., router addresses, recommended hop limit, etc.)
- higher-layer info (e.g., DNS server and NTP server addresses) discovered by multicast / anycast-based service-location protocol [details still to be decided]
- DHCP also available for those who want more control

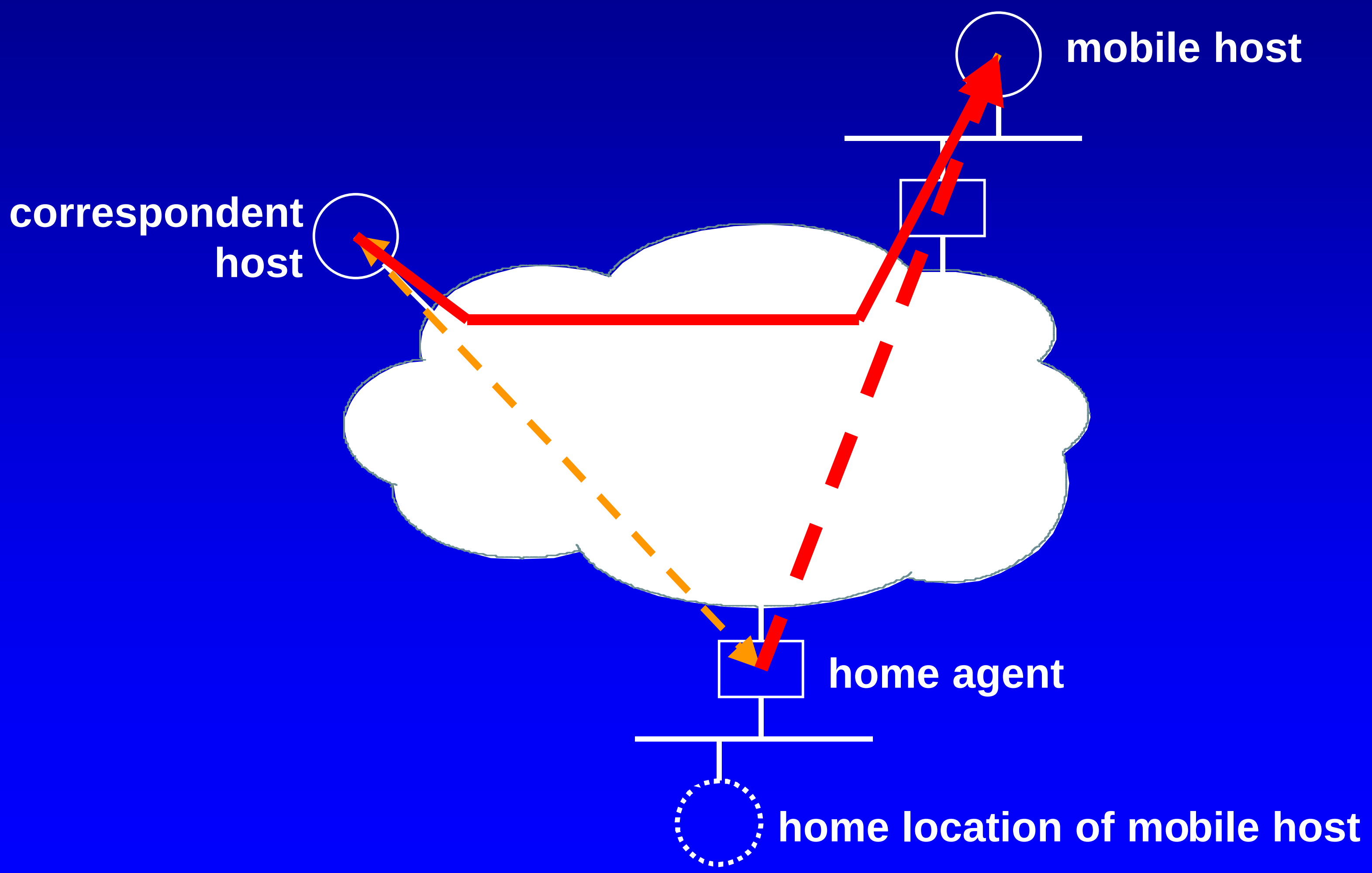
Auto-Reconfiguration ("Renumbering")

- new address prefixes can be introduced, and old ones withdrawn
  - we assume some overlap period between old and new, i.e., no "flash cut-over"
  - hosts learn prefix lifetimes and preferability from router advertisements
  - old TCP connections can survive until end of overlap; new TCP connections can survive beyond overlap
- router renumbering protocol, to allow domain-interior routers to learn of prefix introduction / withdrawal
- new DNS structure to facilitate prefix changes

Mobile IP (v4 version)



Mobile IP (v6 version)



Other Features of IPv6

- flow label for more efficient flow identification (avoids having to parse the transport-layer port numbers)
- neighbor unreachability detection protocol for hosts to detect and recover from first-hop router failure
- more general header compression (handles more than just IP+TCP)
- IPsec & diff-serv features — same as IPv4

How?

IPv4-IPv6 Co-Existence / Transition

a wide range of techniques have been identified and implemented, basically falling into three categories:

(1) dual-stack techniques, to allow IPv4 and IPv6 to co-exist in the same devices and networks

(2) tunneling techniques, to avoid order dependencies when upgrading hosts, routers, or regions

(3) translation techniques, to allow IPv6-only devices to communicate with IPv4-only devices

expect all of these to be used, in combination

Dual-Stack Approach

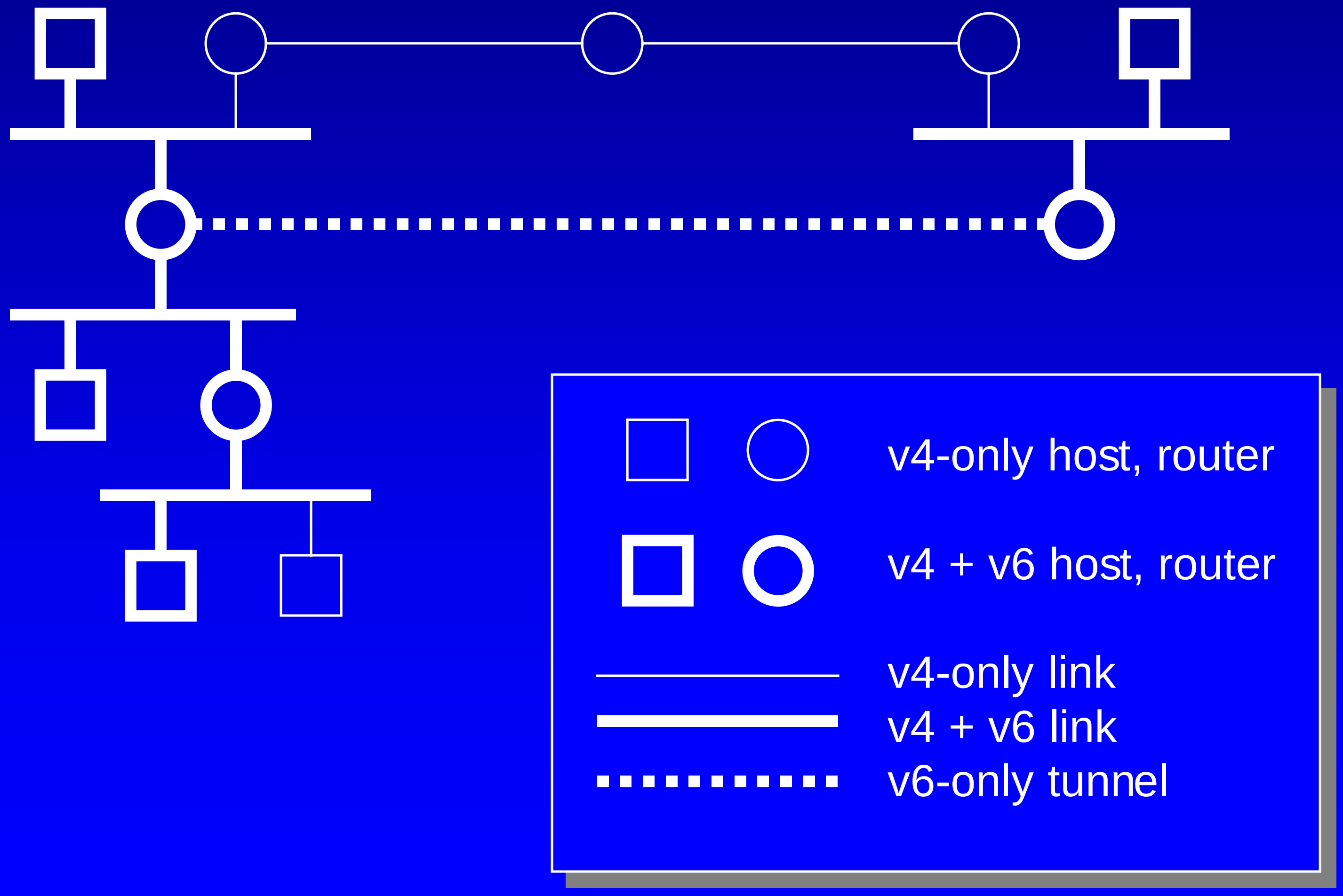
- when adding IPv6 to a system, do not delete IPv4
  - this multi-protocol approach is familiar and well-understood (e.g., for AppleTalk, IPX, etc.)
  - note: in most cases, IPv6 will be bundled with new OS releases, not an extra-cost add-on
- applications (or libraries) choose IP version to use
  - when initiating, based on DNS response:

    if (dest has AAAA or A6 record) use IPv6, else use IPv4
  - when responding, based on version of initiating packet
- this allows indefinite co-existence of IPv4 and IPv6, and gradual, app-by-app upgrades to IPv6 usage

Tunnels to Get Through IPv6-Ignorant Routers / Switches

- encapsulate IPv6 packets inside IPv4 packets (or MPLS frames)
- many methods exist for establishing tunnels:
  - manual configuration
  - "tunnel brokers" (using web-based service to create a tunnel)
  - "6-over-4" (intra-domain, using IPv4 multicast as virtual LAN)
  - "6-to-4" (inter-domain, using IPv4 addr as IPv6 site prefix)
- can view this as:
  - IPv6 using IPv4 as a virtual link-layer, or
  - an IPv6 VPN (virtual public network), over the IPv4 Internet (becoming "less virtual" over time, we hope)

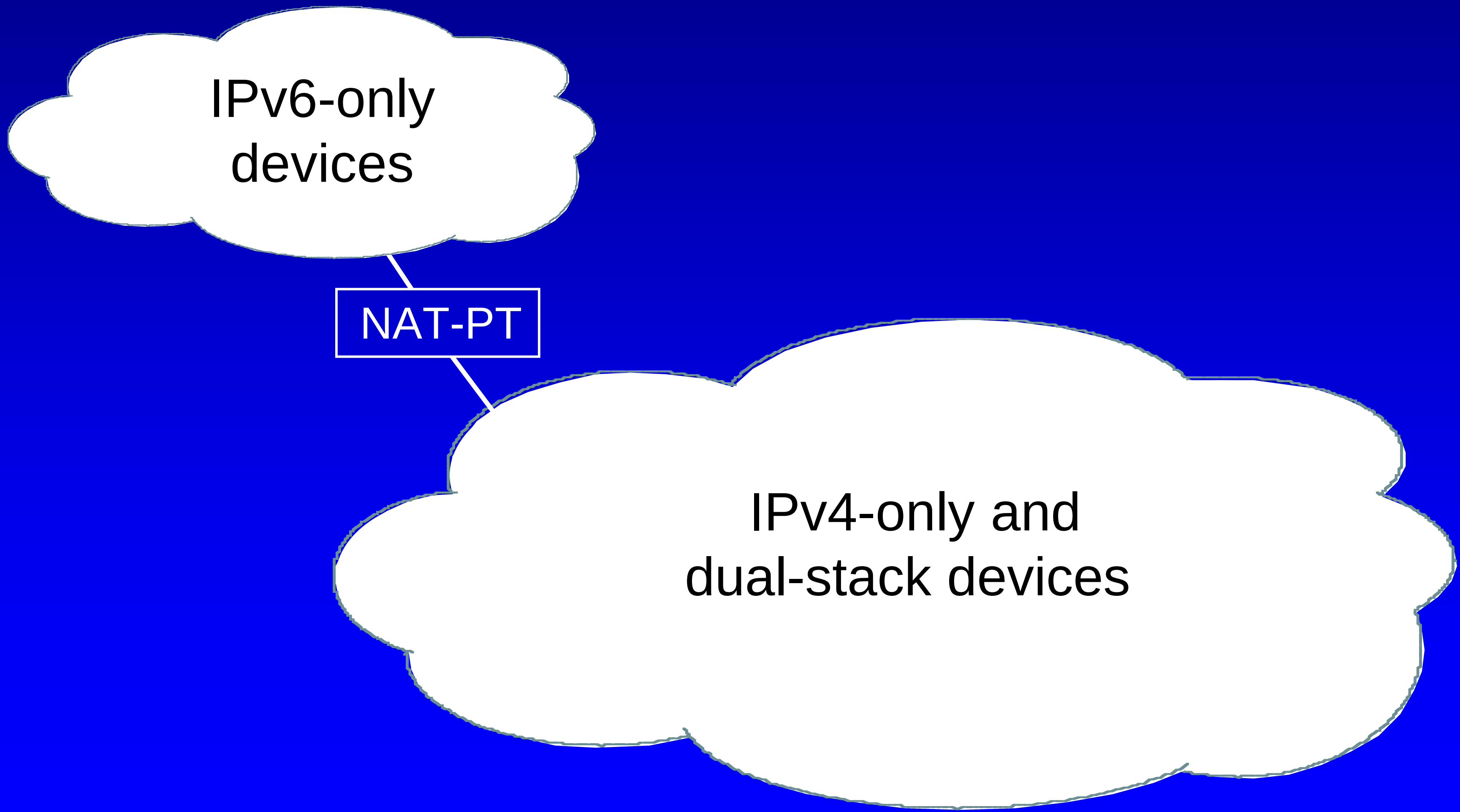
Tunneling Example



Translation

- may prefer to use IPv6-IPv4 protocol translation for:
  - new kinds of Internet devices (e.g., cell phones, cars, appliances)
  - benefits of shedding IPv4 stack (e.g., serverless autoconfig)
- this is a simple extension to NAT techniques, to translate header format as well as addresses
  - IPv6 nodes behind a translator get full IPv6 functionality when talking to other IPv6 nodes located anywhere
  - they get the normal (i.e., degraded) NAT functionality when talking to IPv4 devices
  - methods used to improve NAT functionality (e.g, ALGs, RSIP) can be used equally to improve IPv6-IPv4 functionality

Network Address Translation and Protocol Translation (NAT-PT)



When?

Specifications

- core IPv6 specifications are IETF Draft Standards => well-tested & stable
  - IPv6 base spec, ICMPv6, Neighbor Discovery, Multicast Listener Discovery, PMTU Discovery, IPv6-over-Ethernet,...
- other important specs are further behind on the standards track, but in good shape
  - mobile IPv6, header compression, A6 DNS support, IPv6-over-NBMA,...
- for up-to-date status: http://playground.sun.com/ipng

Implementations

- most IP stack vendors have an implementation at some stage of completeness
  - some are shipping supported product today, e.g., 3Com, Epilogue, Ericsson, IBM, Hitachi, KAME, Nortel, Sun, Trumpet
  - others have beta releases now, supported products “soon”, e.g., Cisco, Compaq, HP, Linux community, Microsoft
  - others known to be implementing, but status unkown (to me), e.g., Apple, Bull, BSDI, FreeBSD, Mentat, Novell, SGI

  (see playground.sun.com/ipng for most recent status reports)
- good attendance at frequent testing events

Deployment

- experimental infrastructure: the 6bone
  - for testing and debugging IPv6 protocols and operations
  - mostly IPv6-over-IPv4 tunnels
  - > 200 sites in 39 countries; mostly universities, network research labs and IP vendors
  - anyone may join: www.6bone.net

Deployment (cont.)

- production infrastructure in support of education and research: the 6ren
  - CAIRN, Canarie, CERNET, Chunahwa Telecom, Dante, ESnet, Internet 2, IPFNET, NTT, Renater, Singren, Sprint, SURFnet, vBNS, WIDE
  - a mixture of native and tunneled paths
  - see www.6ren.net, www.6tap.net
- commercial infrastructure
  - only a few ISPs (IIJ, NTT,SURFnet, Trumpet) have announced intent to provide commercial IPv6 service, so far

Deployment (cont.)

- IPv6 address allocation
  - 6bone procedure for test address space
  - regional IP address registries (APNIC, ARIN, RIPE-NCC) for production address space
- deployment assistance
  - ipv6.org: contributed FAQs and other info
- deployment advocacy (a.k.a. marketing)
  - IPv6 Forum: www.ipv6forum.com

Other Sources of Information

books:

IPv6, The New Internet Protocol
by Christian Huitema (Prentice Hall)

Internetworking IPv6 with Cisco Routers
by Silvano Gai (McGraw-Hill)

and many more... (14 hits at Amazon.com)

video:

IPv6: the New Internet Protocol
by Steve Deering & Craig Mudge
(University Video Communications, www.uvc.com)