U.PORTO

U.PORTO FCUP FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

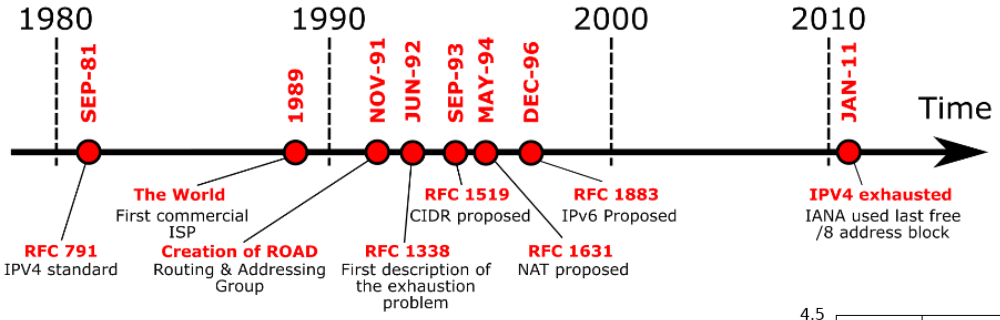
[dcc]

IPv6

Tópicos Avançados em Redes
2023/2024

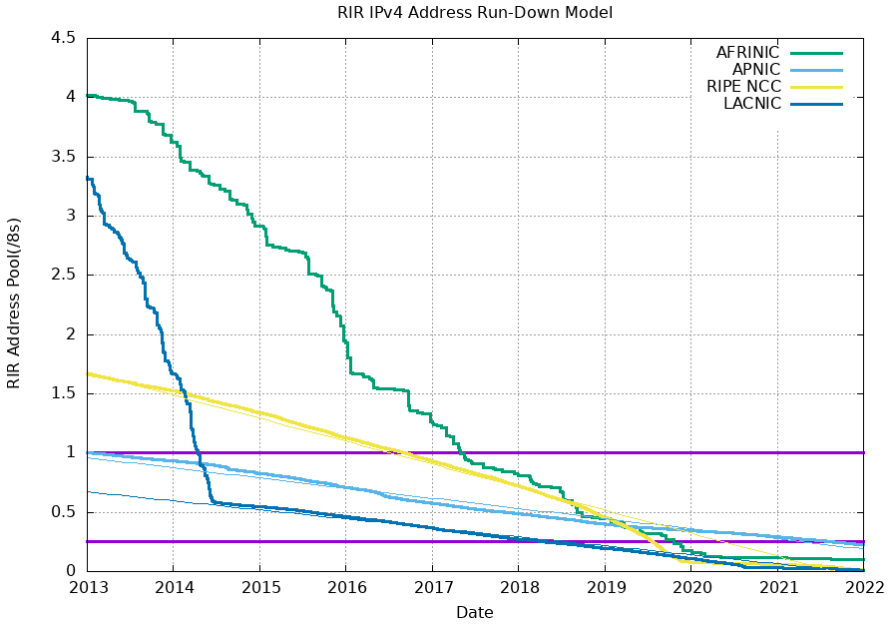
Why IPv6? Addresses!

- Lack of IPv4 addresses – most pressing reason



Source: Wikipedia

- NAT is a temporary band-aid

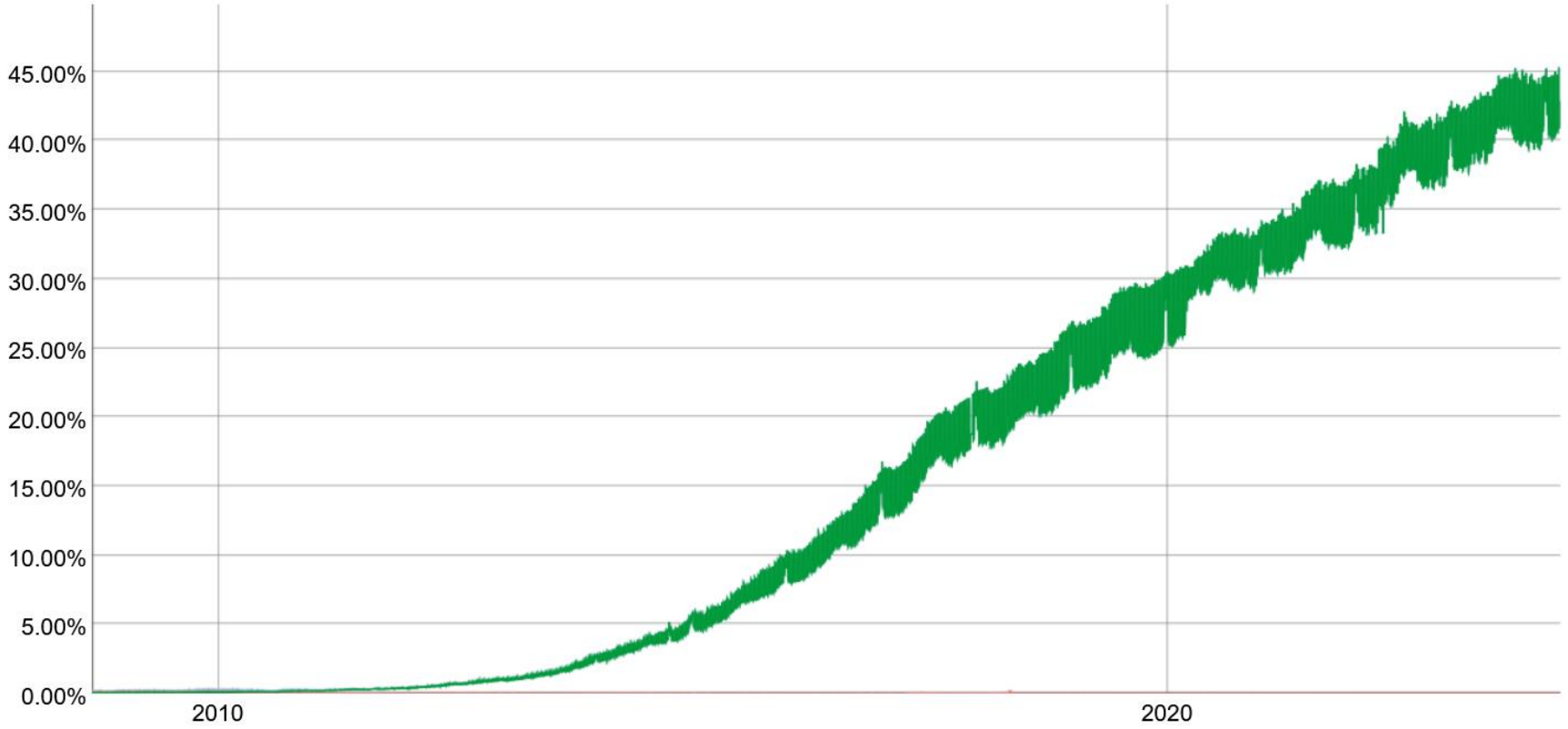


Source: https://www.potaroo.net/tools/ipv4/

Why IPv6? Other reasons

- Improve router performance
  - Simplify IP header
  - Align to 64 bits
  - Address hierarchy with more levels
  - Simplify routing tables
- Improve Mobile IP support
- RFC 8200 (July 2017) – Internet Protocol, Version 6 (IPv6) Specification
  - Previous specifications from 1998 and 1995

% IPv6 accesses (Google)



https://www.google.com/intl/en/ipv6/statistics.html in 2024-02-26

Headers

V6 (40 bytes)

Ver(4)	Class (8)	Flow Label (20)	
Payload Length (16)		Next Header (8)	Hop Limit (8)
Source Address (128)			
Destination Address (128)			

V4 (20+ bytes)

Ver(4)	IHL(4)	DSCP (8)	Total Length (16)	
Identification (16)			Flags(4)	Frag Offset (12)
TTL (8)		Protocol (8)	Header Checksum (16)	
Source Address (32)				
Destination Address (32)				
Options				Padding

Header: fields I

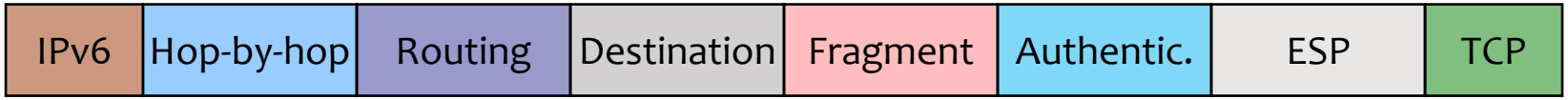
- Ver – protocol version (=6)
- Class – similar to IPv4's ToS / DSCP
- Flow Label – flow identification;
  - Unique for each flow with the same (src,dst) pair
  - Zero means "no flow label"
- Payload length – data field size
  - Limited to $2^{16} - 1 = 65\,535$ bytes, but there there's an option for Jumbograms up to $2^{32} - 1$ bytes
  - No need for total length since the header size is fixed

Header: fields II

- Next Header – type of next header
  - Hop-by-Hop Options
    - the only header examined by intermediate nodes
  - Routing header
  - Destination Options header
  - Fragment header
  - Authentication Header
  - Encrypted Security Payload header
  - Transport layer header
- Hop Limit – IPv4's TTL (but done properly)
- No checksum
  - Error checking performed by lower layers

Header: fields III

- Header order for defined extensions:
  - IPv6 header
  - Hop-by-Hop Options header
  - Destination Options header (processed by 1st dst and Routing Header addresses)
  - Routing header (deprecated)
  - Fragment header (only sender may fragment)
  - Authentication Header (AH)
  - Encapsulating Security Payload header (ESP)
  - Destination Options header (processed by final destination only)
  - Upper-layer (transport) header



Addresses

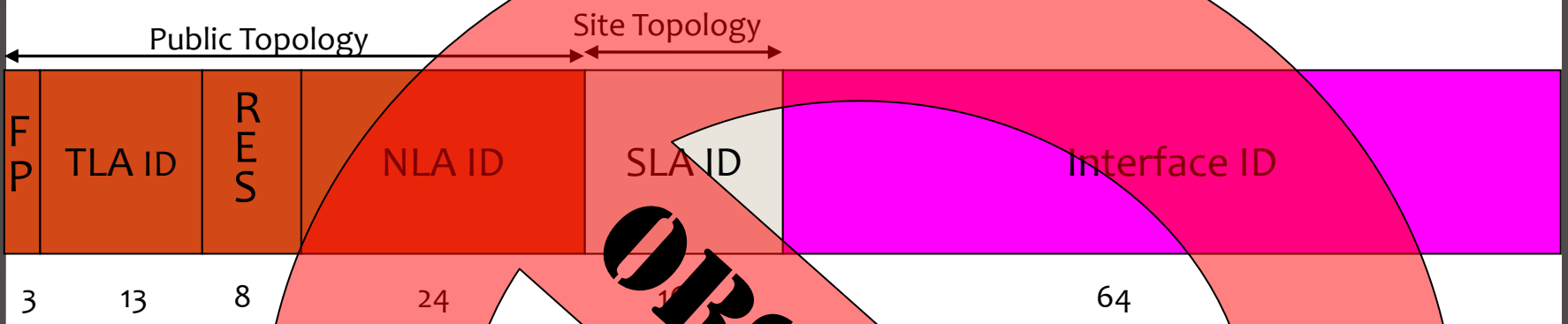
- 128 bits
  - 3.4×10^{38} possible addresses
  - 665 570 793 348 866 943 898 599 addresses per m^2 on earth!
- Represented in hexadecimal (8 groups of 4 hex digits)
  - E.g., 2528:8653:294c:0000:0000:90af:8900:7654
- Leading zeroes may be omitted
  - 2001:db8:0:0:1:0:0:1
- A sequence of zeroes may be abbreviated by ::
  - E.g., 2001:db8::aaaa:0:0:1
  - Only one :: in an address to avoid ambiguity
- Mask (prefix length) similar to IPv4 CIDR
  - E.g., Loopback address is ::1/128
- It is possible to embed IPv4 addresses in IPv6 addresses
  - ::ffff:5.6.7.8 (IPv4-mapped address)
  - ::5.6.7.8 (IPv4-compatible address – deprecated)
  - 2002:506:708::1 (6to4 address embedding the 5.6.7.8 IPv4 address)

See RFC5952 for text representation

Addresses: Unicast

- Loopback ::1 (/128)
- Link-local:
  - Only used on the local link (not routed)
    - FE80::/64 (last 64 bits are device identifier)
  - Device identifier defined in: RFC4291 – Appendix A
- Site-local (*deprecated — see* RFC 3879):
  - Used within the same site, and therefore only valid therein
    - FEC0:0000:0000:<Subnet(16)>:<Interface(64)>
- Global address:
  - Internet routable address
    - 2000::/3
    - But there can be routable IPv6 addresses outside this range (see RFC3587)

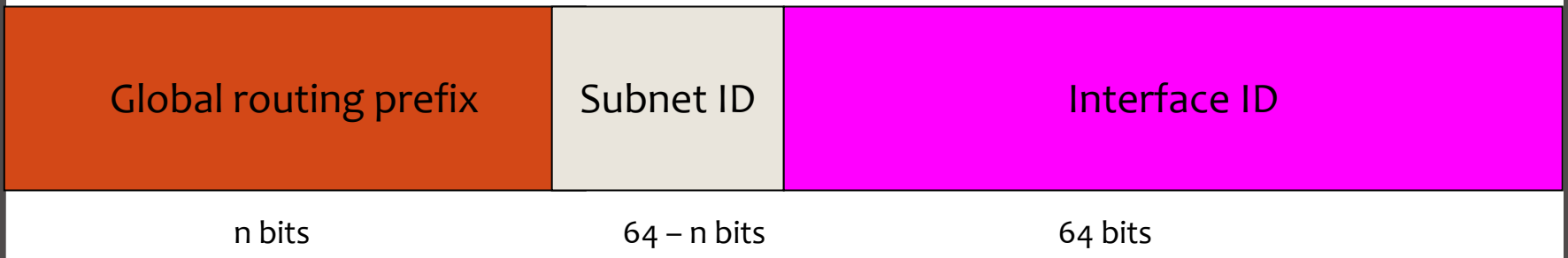
Addresses: Aggregation



OBSOLETE

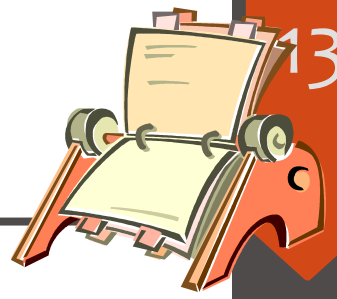
- RFC 2374 – An IPv6 Aggregatable Global Unicast Address Format
- FP – Format Prefix (001)
- RES – Reserved para future use
- TLA – Top Level Aggregator
- NLA – Next Level Aggregator
- SLA – Site Level Aggregator
- Interface ID

Addresses: Aggregation II

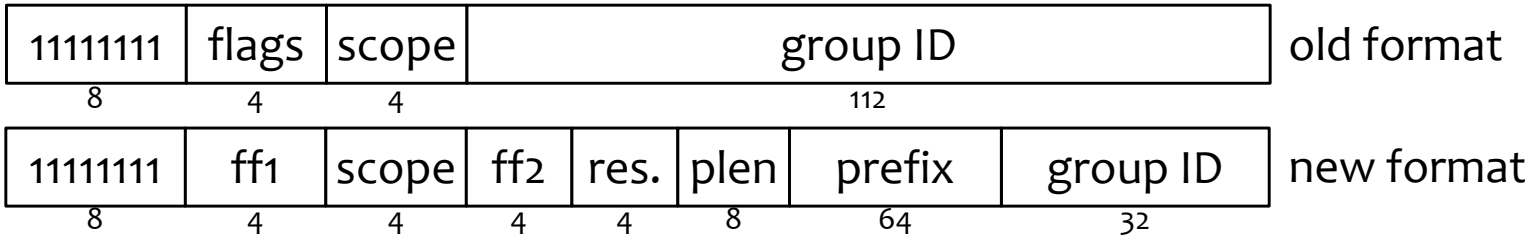


- RFC 3587 changed the aggregation
  - Address Policy left for Regional Internet Registries (RIRs)
  - Maintains the normal structure of RFC 4291

Addresses: multicast



- Multicast:
  - With different scopes



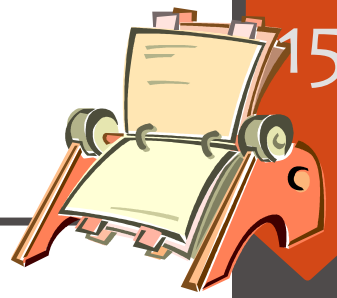
- 4-bit multicast scope field used to limit the scope of the multicast group

0 Reserved
1 Interface-Local scope
2 Link-Local scope
3 Realm-local scope
4 Admin-Local scope
5 Site-Local scope
6 (unassigned)
7 (unassigned)
8 Organization-Local scope
9 (unassigned)
A (unassigned)
B (unassigned)
C (unassigned)
D (unassigned)
E Global scope
F Reserved

Addresses: multicast examples

- All Nodes Addresses:
  - ff01:0:0:0:0:0:0:1
  - ff02:0:0:0:0:0:0:1
  - The above multicast addresses identify the group of all IPv6 nodes, within scope 1 (interface-local) or 2 (link-local).
    - Interface-local is used for loopback
- All Routers Addresses:
  - ff01:0:0:0:0:0:0:2
  - ff02:0:0:0:0:0:0:2
  - ff05:0:0:0:0:0:0:2
- All NTP servers on the Internet
  - ff0e:0:0:0:0:0:0:101

Addresses: other



- Anycast address
  - Defined on more than one interface, but delivered only to one
  - Same structure of unicast
    - A unicast address becomes anycast if routers are configured to recognize it as such
- Unspecified address
  - ::
  - Used only during configuration

Required addresses (RFC4291#2.8)

- Node:
  - Loopback
  - Link-local for each interface
  - Configured unicast (or anycast)
  - All-nodes multicast
  - Solicited-node multicast address for each of its unicast and anycast addresses
  - Multicast groups that the node belongs to
- Router: all the above plus
  - Subnet-Router Anycast (equal to the subnet prefix)
  - All-Routers multicast

ICMPv6

ICMPv6

- Internet control message protocol for IPv6 RFC 4443
- Type defines the type, code a subtype
  - Similar to ICMP (v4)
- Ex.:
  - Type = 1 (Error code for *destination unreachable*)
  - Code = 0 (*no route to destination*)
- Errors have type with high-order bit = 0 (0-126)
  - Informational are 128-254

Type(8)	Code(8)	Checksum (16)
Message content (length depends on type)		

ICMPv6 – Neighbour Discovery

- RFC 4861
- Replaces IPv4 ARP
- Adds auto-configuration helper functions
- Extensible: messages may have options for added information
- IP Hop limit: set to 255
  - so that routed packets (HL< 255) are identified as invalid (see RFC4861#7.1.1)

ICMPv6 – ND: Address Resolution

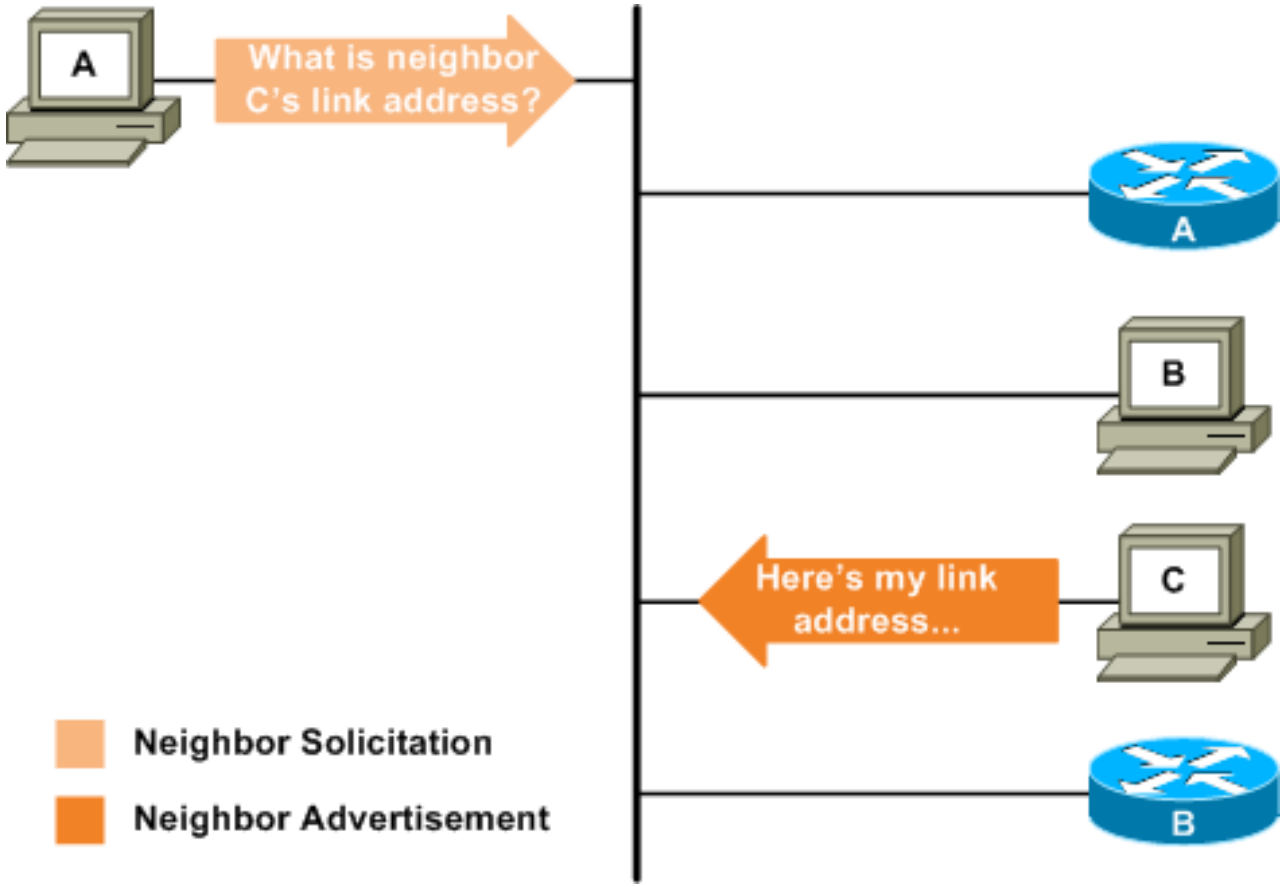


Image source: packetlife.net

ICMPv6 – Neighbour Discovery

- Neighbour solicitation
  - Sent by hosts or routers
  - Used for address resolution, neighbour unreachability detection (NUD), duplicate address detection (DAD)
  - Type=135, code = 0

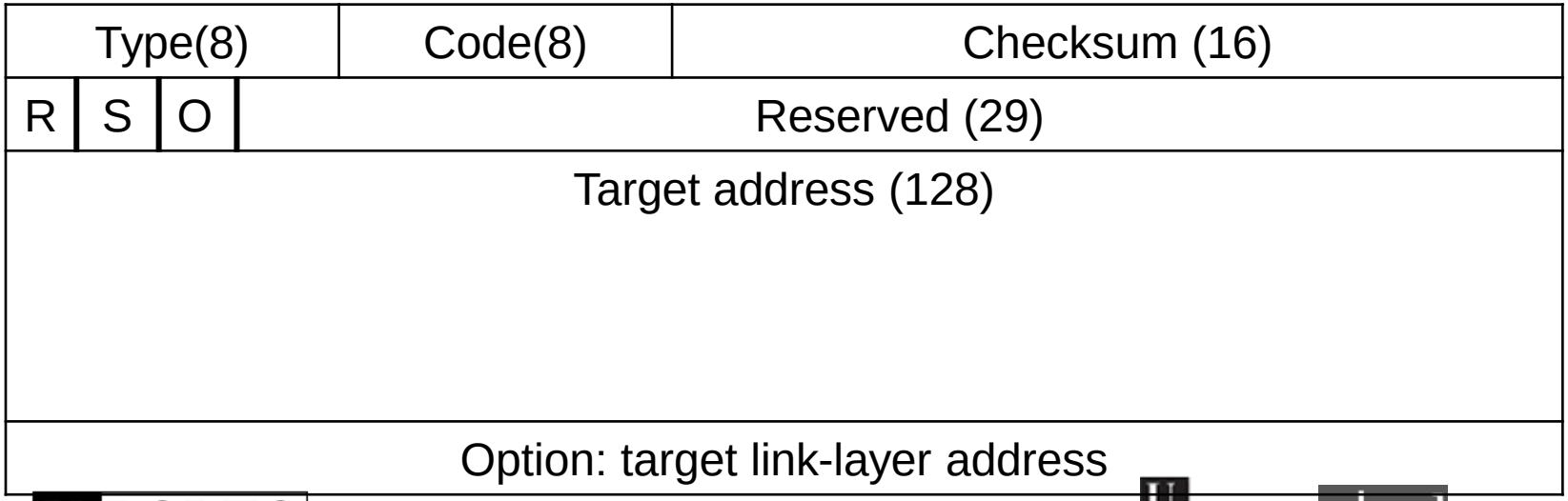
Type(8)	Code(8)	Checksum (16)
reserved		
Target address (128)		
Option: source link-layer address		

ICMPv6 – Neighbour Discovery

- Neighbour solicitation (cont.)
  - Target address
    - IP address of the solicitation target
  - IP destination address
    - solicited-node multicast address corresponding to the target address, or the target address itself (e.g., for NUD)
  - IP source address
    - address of the interface where the packet is sent, or unspecified address for DAD

ICMPv6 – Neighbour Discovery

- Neighbour advertisement
  - Sent by hosts or routers, in response to solicitations or unsolicited
  - Type=136, code = 0
  - Target address:
    - For solicited: same as in solicitation
    - For unsolicited: IP address that changed link-layer address
  - Option: source link-layer address of sender



Solicited-Node MC Address (RFC4291#2.7.1)

- Multicast address
- Prefix: ff02:0:0:0:0:1:ff00::/104
- Add the low-order 24 bits of an address (unicast or anycast)
- Example:
  - Unicast address: 4037::01:800:200e:8c6c
  - Solicited Address: ff02:0:0:0:0:1:ff0e:8c6c
- *Most probably, no two nodes in the same network map to the same multicast address ➔ much more efficient than the local broadcast used by ARP!*

ICMPv6 – ND: Router Discovery

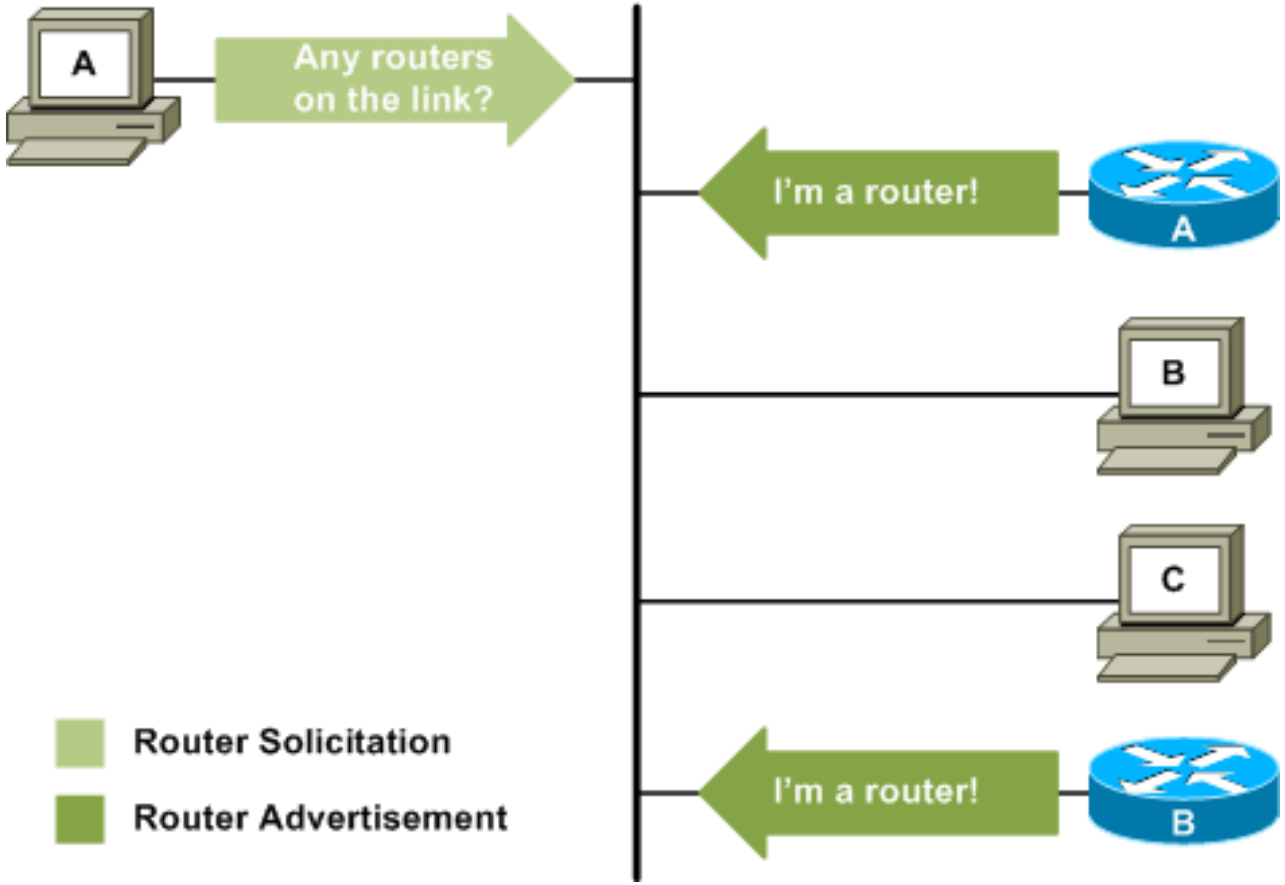


Image source: packetlife.net

ICMPv6 – ND: Router Discovery

- For discovering routers on the link
  - And, possibly, address information
- Router solicitations
  - Sent by hosts to the link-local *all-routers* multicast address
  - Type = 133, code = 0
- Router advertisements
  - Sent by routers
    - In response to solicitations — to unicast address of requester
    - Unsolicited — to *all-nodes* multicast address with link-local scope
  - Advertise: net prefix, default router(s), MTU
  - Type=134, code = 0

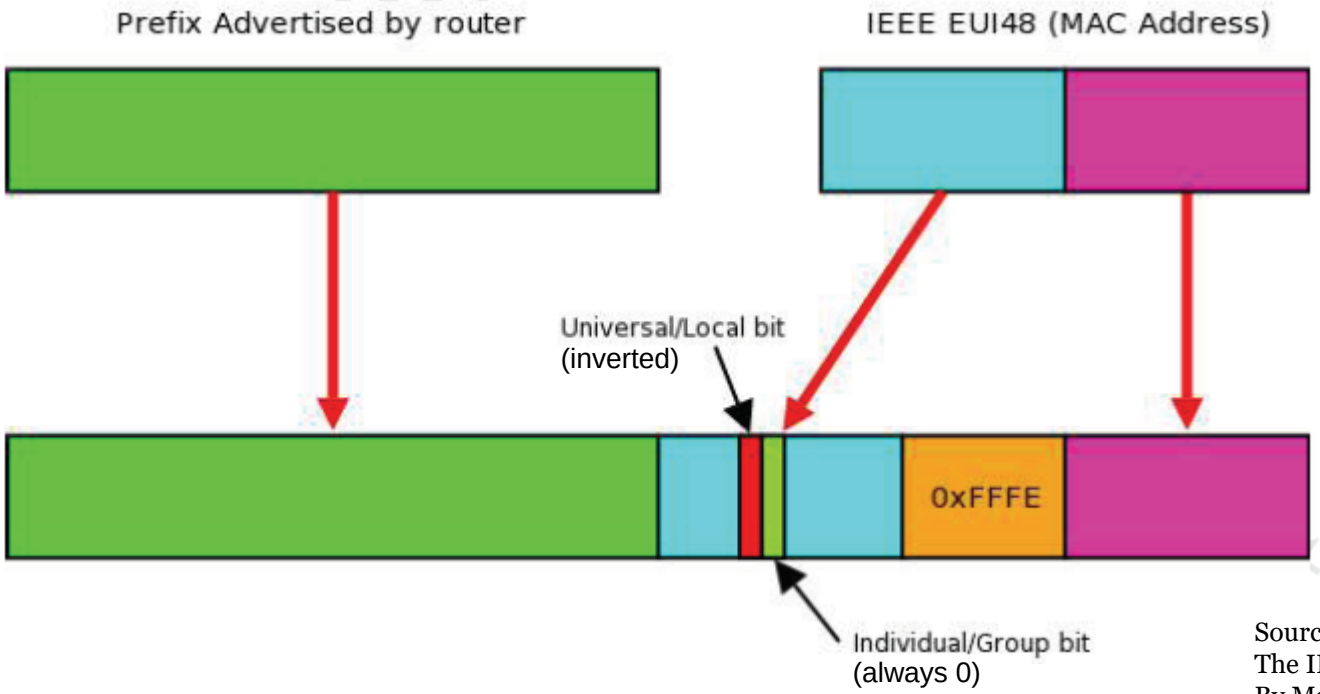
Auto Configuration — SLAAC

- **StateLess** Address AutoConfiguration (RFC4862)
  1. Generating link-local address
  2. Generating global address(es)
  3. Performing duplicate address detection (DAD)

- Based on *Router Advertisements*
- Uses advertised prefix and MAC address of interface
- Valid only for /64 networks

- *What if no router advertisements?*
- *Are link-local addresses enough? For what?*

Auto Configuration Address



Source:
The IPv6 Protocol,
By Mauro Tortonesi

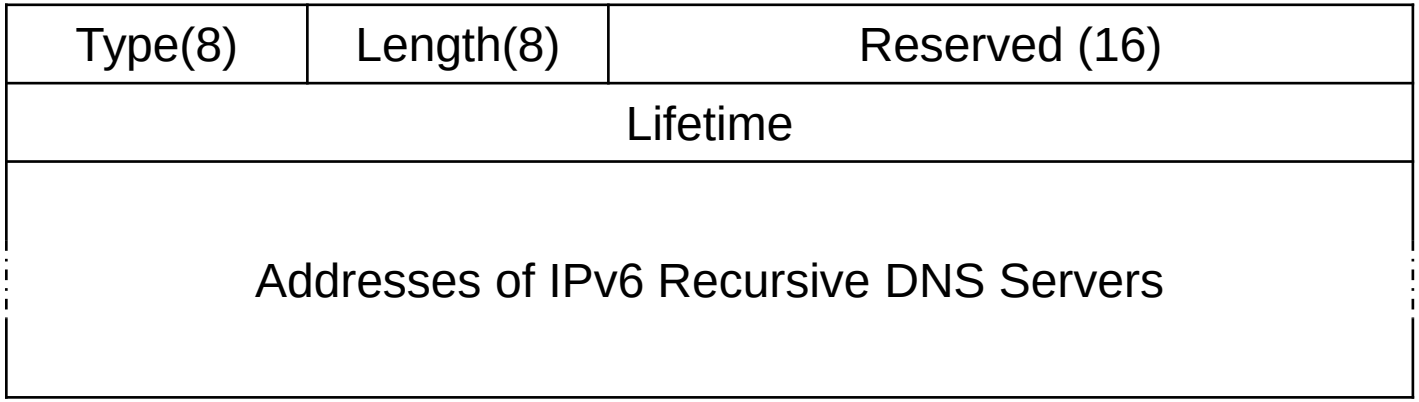
Example:

prefix 2000:0:0:1::/64 + MAC addr. 00:c0:df:08:d5:99 ➔ 2000::1:2c0:dfff:fe08:d599

NOTE: an alternative, more privacy-preserving method is defined in RFC7217 for generating opaque, yet stable addresses

What about DNS servers?

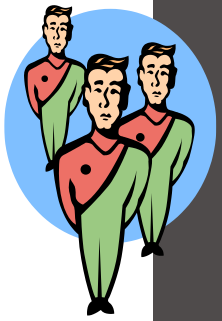
- RFC 8106: IPv6 Router Advertisement Options for DNS Configuration



- Type: 25
- Length: indicates the length of the option
  - Implicitly the number of advertised DNS servers

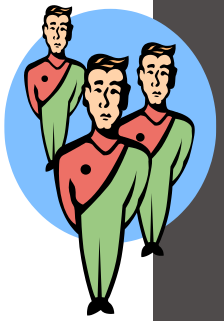
Duplicate Address Detection

- RFC 4862#5.4
- Used after address auto-configuration to check uniqueness of address
  - Not performed for anycast addresses (configured)
- Node sends neighbour solicitation with:
  - Src address: unspecified
  - Dst address: solicited-node multicast address of the tentative address
  - Target address: the tentative address



Duplicate Address Detection II

- Receiving Neighbour Solicitation
  - Target address is the tentative address
    - If src address is unicast (not DAD): ignore packet
    - If src address is unspecified: other node wants to use same address (DAD)
      - No node will use the target address (as extra precaution)
- Receiving Neighbour Advertisement
  - Target address is the tentative address
  - Address is not unique (cannot be used)



Path MTU Discovery

- IPv6 does not support fragmentation in routers
- Senders need to transmit packets ≤ smallest MTU along the path (i.e., Path MTU)
- Procedure for discovering the Path MTU
  - Source node assumes path MTU = 1st hop link MTU
  - If this is too large for a link
    - router drops the packet
    - returns an ICMPv6 Packet Too Big (type=2, code=0)
  - Source reduces MTU to the indicated size
  - When the MTU used by the source ≤ Path MTU, packets reach the destination

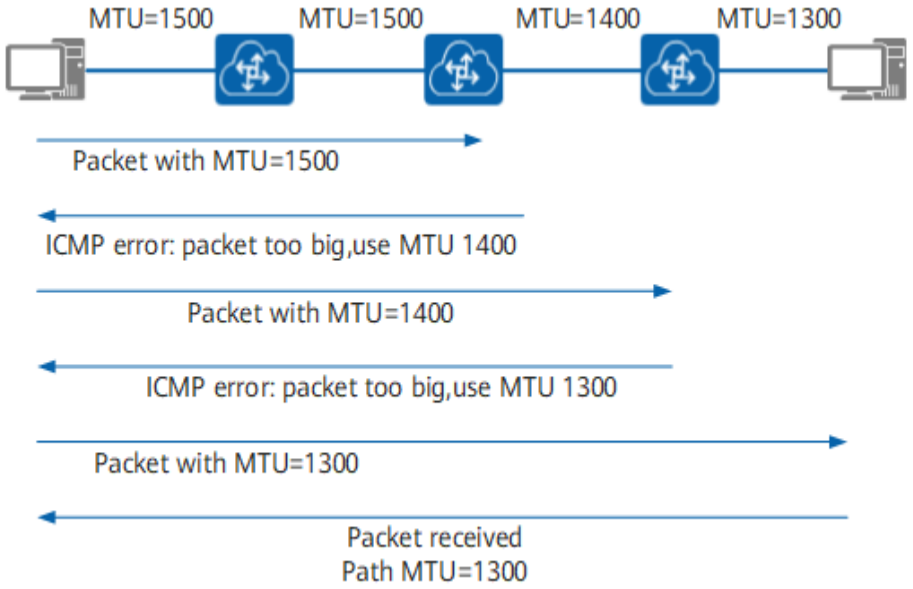


Image source: Huawei

Other layers

DHCPv6 – RFC8415

- Stateful configuration
- Uses UDP (like DHCPv4)
- DHCP clients use the link-local address
- Uses two multicast groups
  - All_DHCP_Relay_Agents_and_Servers (ff05::1:3)
    - Site-scoped
  - All_DHCP_Servers (ff02::1:2)
    - Link-scoped
    - Clients send requests to this address
- Requires DAD to ensure address uniqueness

DHCPv6

- Possible to obtain partial information
  - Information-Request message
  - E.g., to retrieve only DNS and NTP servers (no address configuration)
- DUID: DHCP Unique Identifier
  - Used by clients and servers to uniquely identify each other
  - Not all messages require this ID
- Optional authentication and encryption

DHCPv6 – Some messages

- SOLICIT (1): Sent by client to locate servers
- ADVERTISE (2): Sent by server in response to a Solicit message
- REQUEST (3): Sent by client to request a lease or delegated prefix
- REPLY (7): Sent by server
  - In response to Solicit, Request, Renew, Rebind messages received from a client
  - In response to an Information-request message
  - In response to a Confirm message, confirming or denying that the addresses assigned to the client are appropriate to the link to which the client is connected
  - To acknowledge receipt of a Release or Decline message
- INFORMATION-REQUEST (11): Sent by client to request configuration parameters without assignment of an IP addresses

DHCPv6 vs DHCP(v4) messages

DHCPv6 Message Type	DHCPv4 Message Type
Solicit (1)	DHCPDISCOVER
Advertise (2)	DHCPOFFER
Request (3), Renew (5), Rebind (6)	DHCPREQUEST
Reply (7)	DHCPACK / DHCPNAK
Release (8)	DHCPRELEASE
Information-Request (11)	DHCPINFORM
Decline (9)	DHCPDECLINE
Confirm (4)	none
Reconfigure (10)	DHCPFORCERENEW
Relay-Forw (12), Relay-Reply (13)	none

DHCPv6 – Information

- Carried in options
- Example: RFC3646: DNS Configuration options for DHCPv6

Option_DNS_Servers (16)	Option-len (16)
DNS-recursive-name-server (IPv6 address)	
…	

- `Option_DNS_Servers`: 23
- `Option-Len`: multiple of 16

Recursive DNS server configuration

- Approaches for DNS configuration in clients (RFC4339)
  - Router Advertisements
  - DHCPv6
  - Well-known anycast address

DNS – changes for IPv6 (RFC3596)

- AAAA new record type
  - IPv6 address

- AAAA query
  - Returns all IPv6 addresses associated with domain name
- IP6.ARPA domain (similar to IPv4's in-addr.arpa)
  - For reverse (PTR) queries
    - Resolve from IPv6 address to host name
- Existing query types
  - **NS, SRV, MX**: re-defined to also return AAAA entries

DNS in IPv4 and IPv6

- BCP91/RFC3901
- Maintain IPv4 and IPv6 accessible DNS recursive server
  - IPv4 only or dual stack
  - At least one IPv4 reachable server per DNS zone

Transition and interoperation mechanisms

IPv6 in an IPv4 world

- Problem is incompatibility
  - For hosts and routers: header of IPv6 vs. IPv4
  - For applications: Sockets API for IPv6 is different (RFC 3493)
    - Compatibility maintained for IPv4 addresses
    - Use of the IPv4-mapped address:
      - ::ffff:<IPv4-address>
  - Also for applications: RFC6535 Bump-In-the-Host
    - "BIH hides IPv6 and makes the IPv4-only applications think they are talking with IPv4 peers by local synthesis of IPv4 addresses"

IPv6 in an IPv4 world: solutions

- Dual stack nodes (RFC 4213)
  - Support both IPv4 and IPv6
  - Routers must maintain both routing tables, protocols, etc.
- Configured Tunneling (RFC 4213)
  - Aka 6in4
  - The tunnel endpoints encapsulate the original IPv6 in an IPv4 packet
    - Using their own addresses as source/destination
  - Tunnel endpoints (routes) are manually configured
  - Does not scale well
    - Requires a different tunnel for each pair of routers ➔ $O(n^2)$

IPv6 in an IPv4 world: tunnelling

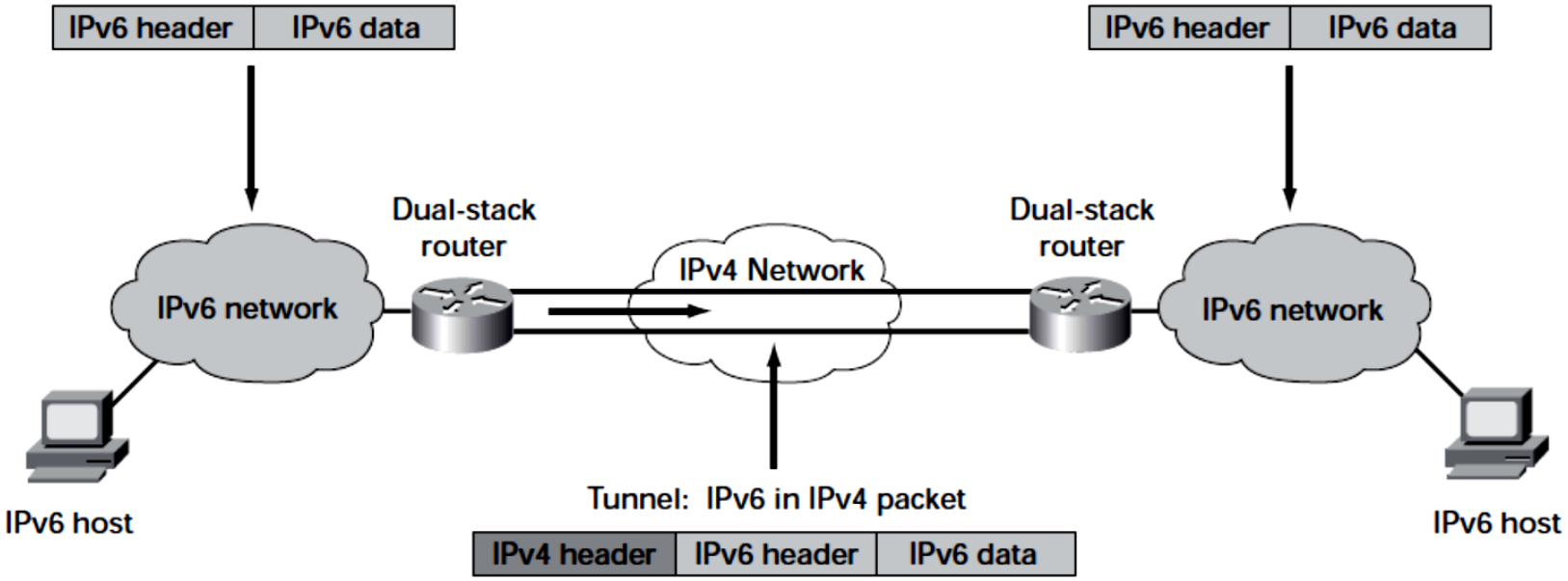
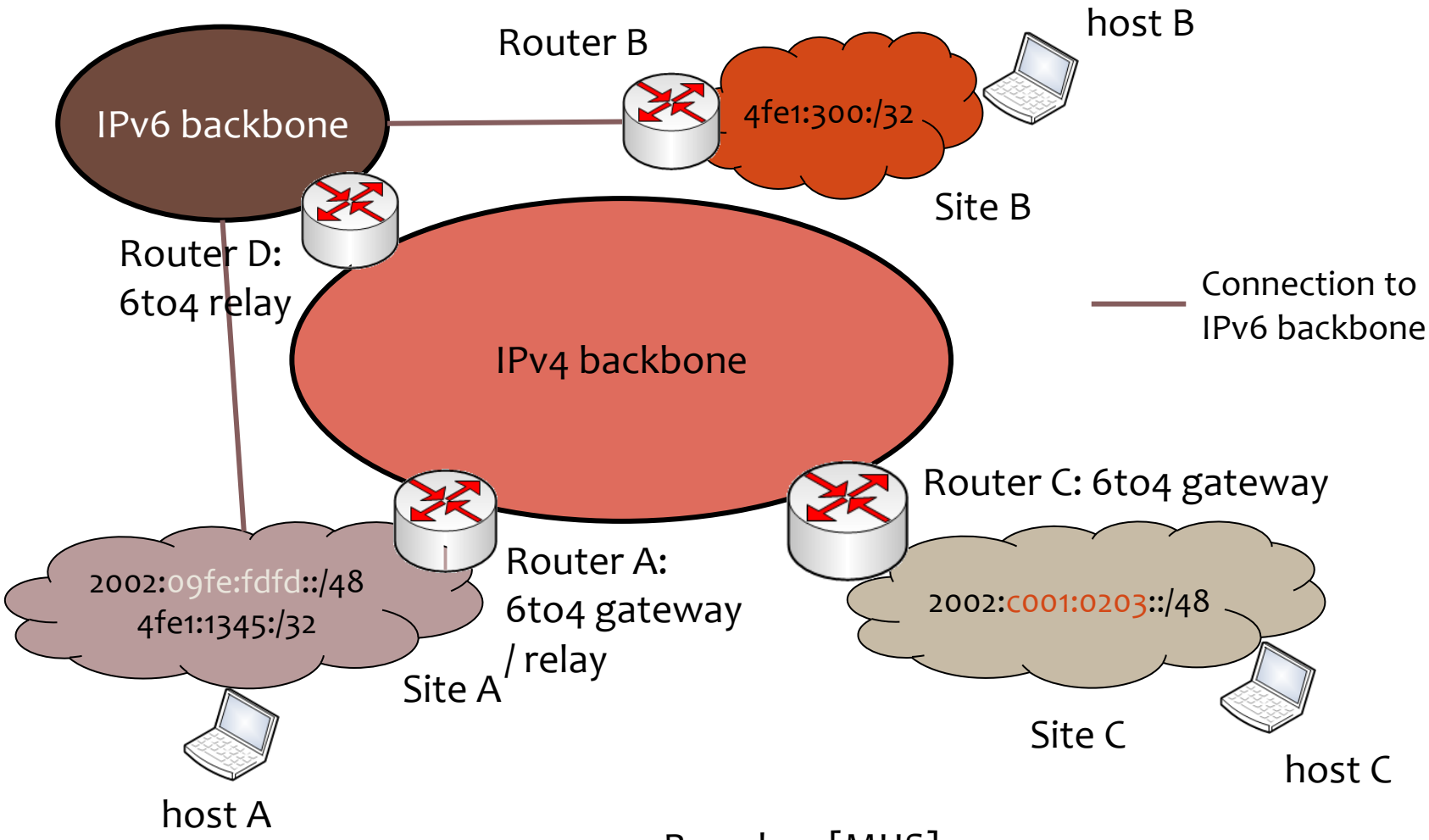


Image from [CiscoIPv6ABC]

IPv6 in an IPv4 world: 6to4

- 6to4 Tunnel (without explicit setup) (RFC 3056)
- Use of 2002:*V4ADDR*::/48 networks
  - V4ADDR is the IPv4 address of the entry router
  - Allows for many /64 prefixes in the IPv6 island
  - Routers advertise these prefixes on local IPv6 networks
    - 2002::/16
- Sites use 6to4 relay to communicate through the tunnels
  - These relay routers can have an IPv4 anycast address of 192.88.99.1 (RFC3068)
  - Sites may also have connection to IPv6-only networks

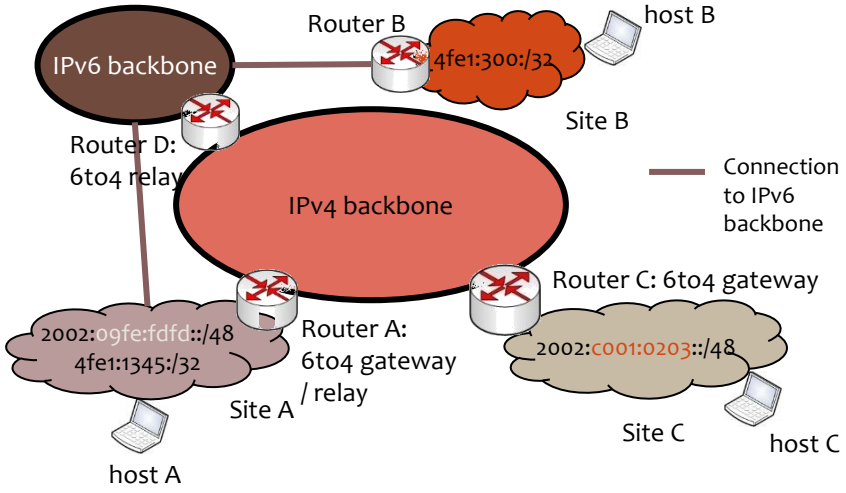
6to4 components



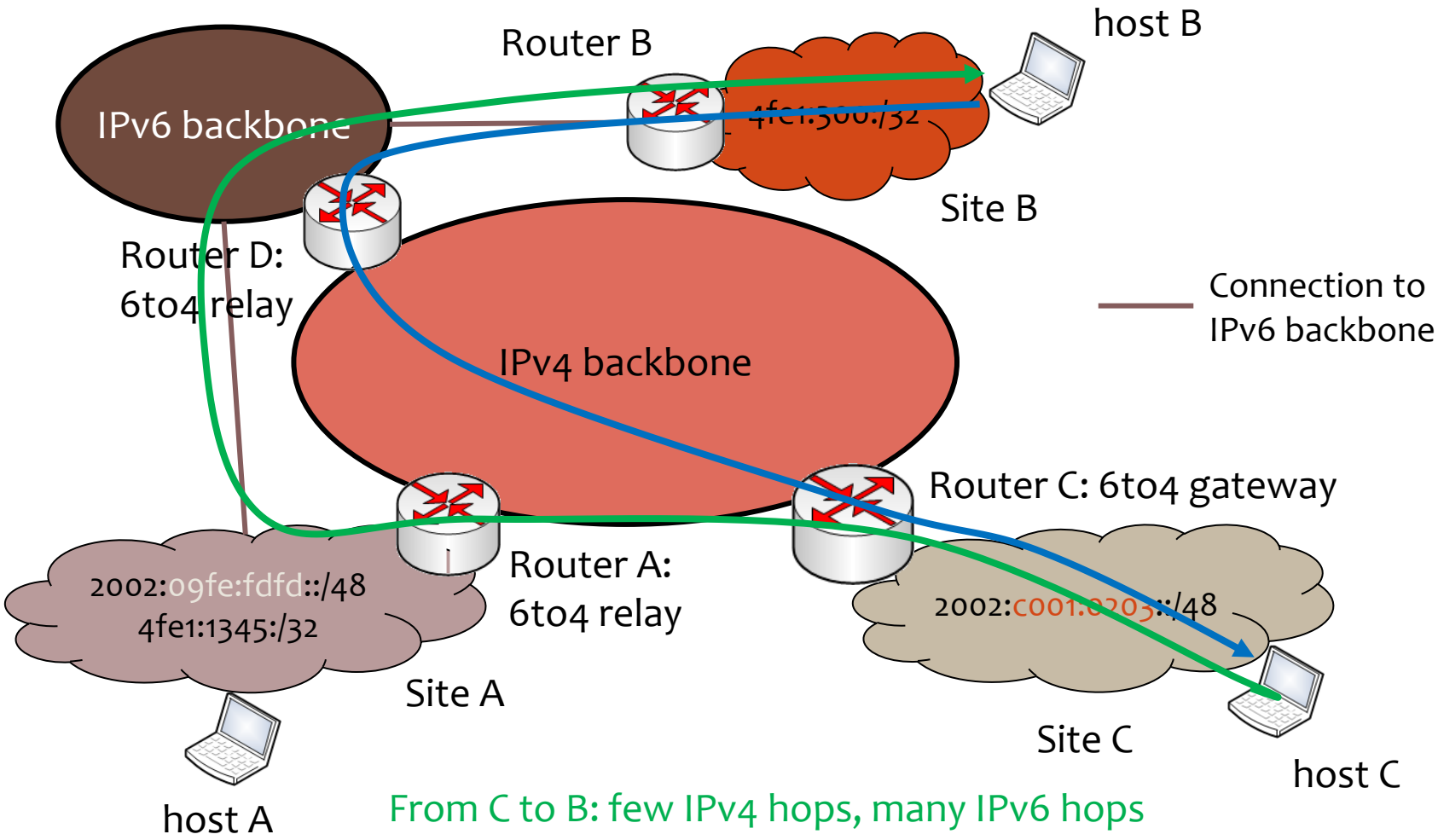
Based on [MHS]

6to4 Components: II

- Router B
  - Connects to the IPv6 backbone
  - Has a native IPv6 address (not 2002::/16)
  - Does not handle IPv4
- 6to4 Router C
  - Connects to the IPv4 backbone
  - Has a 6to4 IPv6 address
  - Has a native IPv4 address
  - Does not relay packets to the IPv6 backbone
- 6to4 relay Router A
  - Connects to the IPv4 backbone
  - Connects to the IPv6 backbone
  - Has a 6to4 IPv6 address
  - Has a native IPv6 address (not 2002::/16)
  - Has a native IPv4 address



6to4 causes asymmetric routing



From C to B: few IPv4 hops, many IPv6 hops

From B to C: many IPv4 hops, few IPv6 hops

Other 6to4 issues

- 6to4 relays may receive traffic from anywhere
  - An ISP deploying a 6to4 relay may receive traffic from extraneous (non-customer) users
    - E.g., owner of router D in previous slide relays traffic from non-customers B and C
  - Traffic may become unpredictable
  - Performance may suffer
- 6rd is a slight modification that solves this issue

6rd

- IPv6 Rapid Deployment on IPv4 Infrastructures (RFC5569)
- Similar to 6to4 but using ISP-specific prefix instead of 2002::/16

ISP-specific Prefix	IPv4 address	(remaining bits)

  - Allows use of private IPv4 addresses
  - ISP-specific prefix length not fixed
- The ISP
  - Operates one or more gateways at the IPv4/IPv6 border
  - Deploys relays with specific anycast addresses (only for its customers)
  - Has more control over traffic flowing through its network

IPv6 in an IPv4 world: ISATAP

- Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) (RFC 5214)
- Connects dual-stack nodes over IPv4 networks
- Uses entire IPv4 network as a link layer (NBMA)
- Defines locators (mappings) on interfaces that are used to route packets
- Potential router list (PRL) obtained by doing a DNS lookup for isatap.<domain> (or some other means)
  - Necessary for sending Router Solicitations and communicating with the public IPv6 Internet
    - Not on the same physical link
- ISATAP IPv6 addresses:
  - <prefix>:0000:5efe:<IPv4 addr>

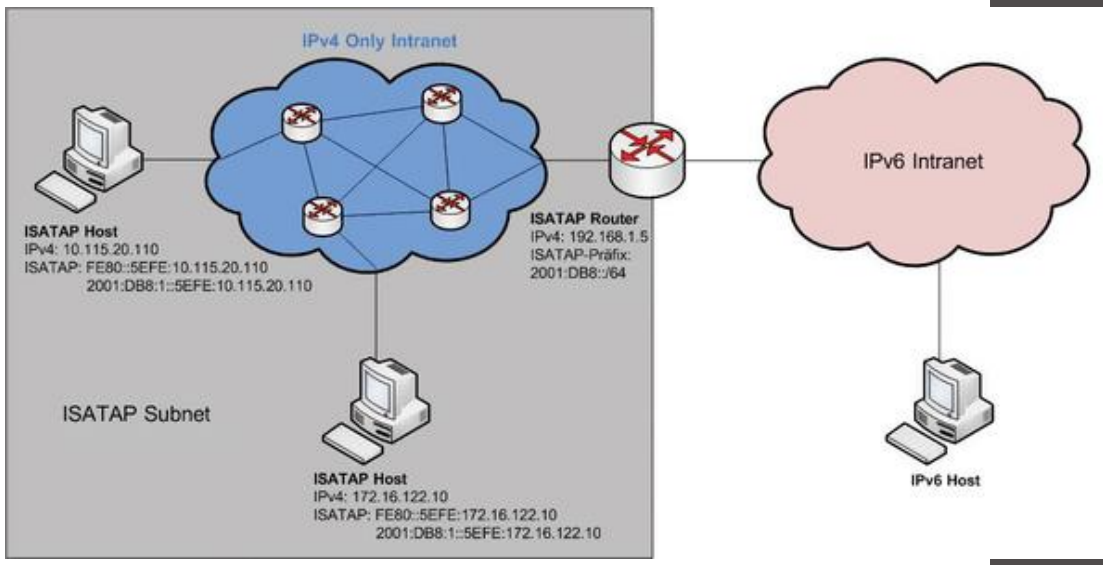


Image from Admin Magazine IPv6

IPv6 in an IPv4 world: Teredo (RFC 4380)

- Teredo allows automatic IPv6 tunneling between hosts that are located across one or more IPv4 NATs
- Encapsulates IPv6 packets in UDP
  - More overhead than other techniques ➔ use only when they are infeasible
- Use of 2001:0000::/32 for Teredo clients
  - 2001:0000:<teredo server IP address>:<flags (16)>:<obscured UDP port>:<obscured public IPv4 address>
- Discovers and maintains NAT mappings to the client
- Components:
  - Client: has IPv4 connectivity, wants IPv6
  - Server: to discover external address and type of NAT
  - Relay: forwards traffic using the Teredo encapsulation on IPv4 and to the IPv6

IPv6 in an IPv4 world: Teredo

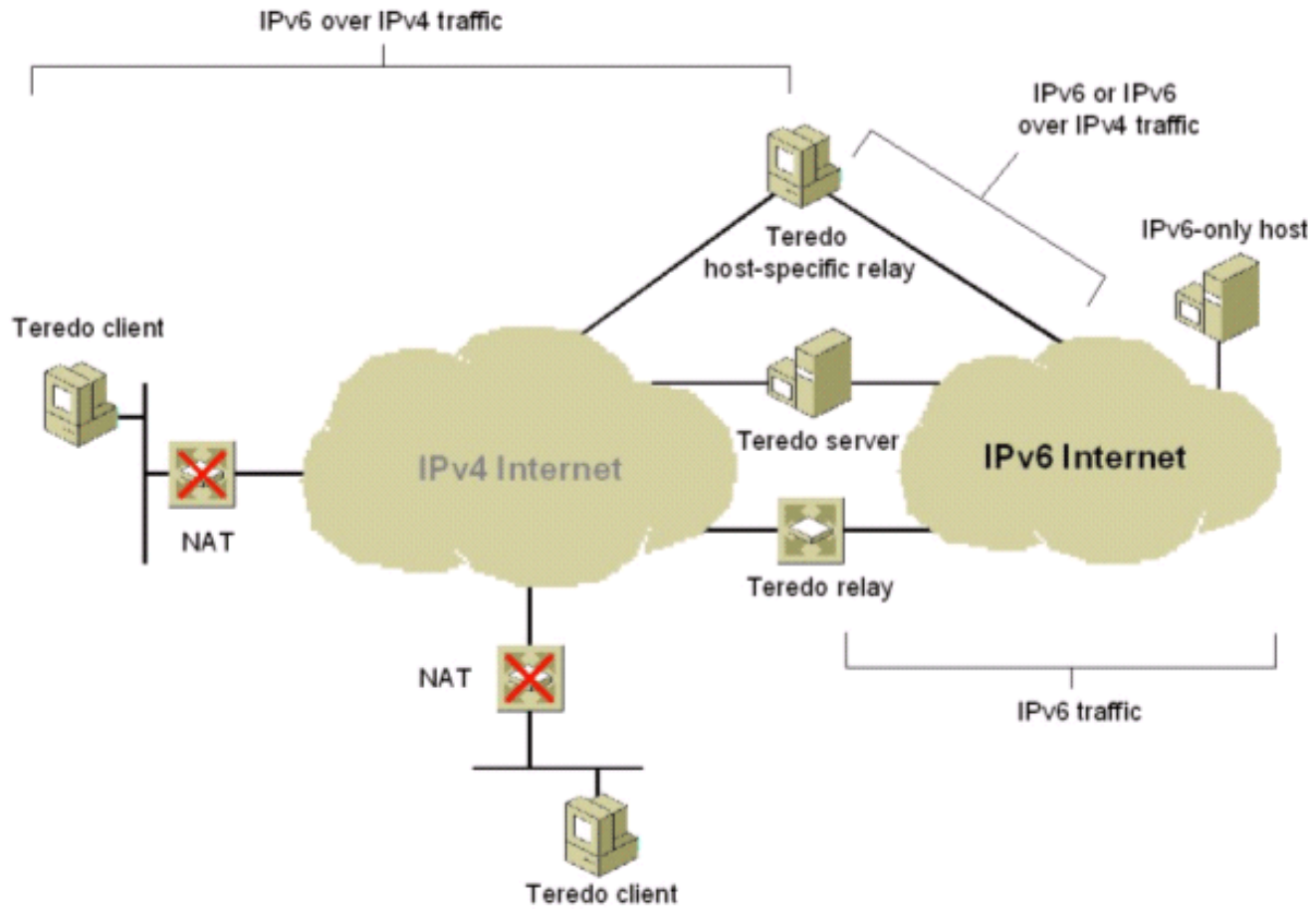


Image from Microsoft Technet article

IPv4 in an IPv6 world

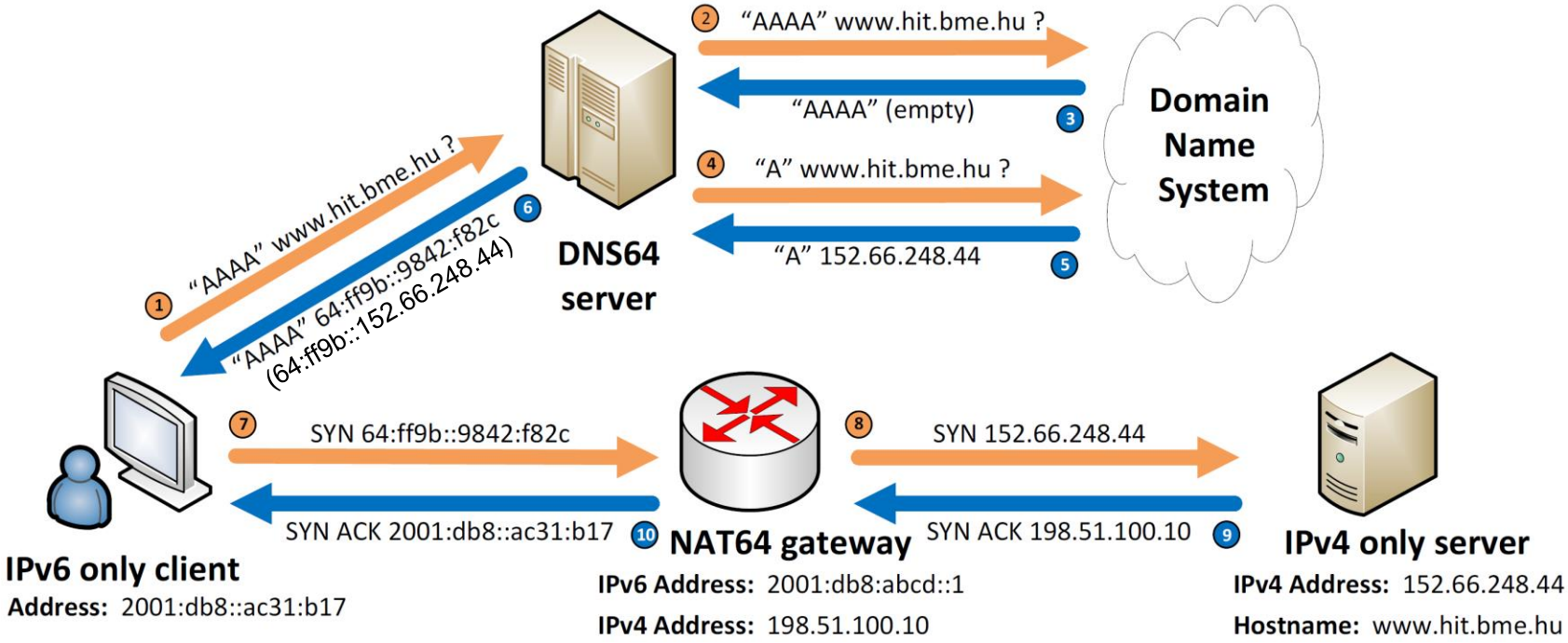
- Increasing deployment of IPv6 ➔ need for
  - Nodes in IPv6-only networks to reach IPv4-only services on the Internet
  - Traversal of IPv6-only operator networks to reach IPv4-only services on the Internet
- Larger IPv6 addresses make it easier to use translation / NAT instead of tunneling
  - IPv6 addresses can embed IPv4 addresses

IPv4 in an IPv6 world: NAT64/DNS64

- Use case: node in IPv6-only network needs to access IPv4-only services on the Internet
- NAT64 (RFC 6146) works like IPv4 NAPT, but using IPv6 instead of IPv4 with private addresses
  - Stateful NAT
  - May use a single public IPv4 address
- NAT64 IPv6 prefix may be
  - Provider-specific
  - Well-known — 64:ff9b::/96
- DNS64 (RFC 6147) synthesizes AAAA records for services providing only A records
  - Using the chosen prefix
- Services provided on the IPv6 Internet are used directly

IPv4 in an IPv6 world: NAT64/DNS64

- Example:



NOTE: The IPv6 address 64:ff9b::9842:f82c is the same as 64:ff9b::152.66.248.44

Image source: http://doi.org/10.11601/ijates.v5i2.129

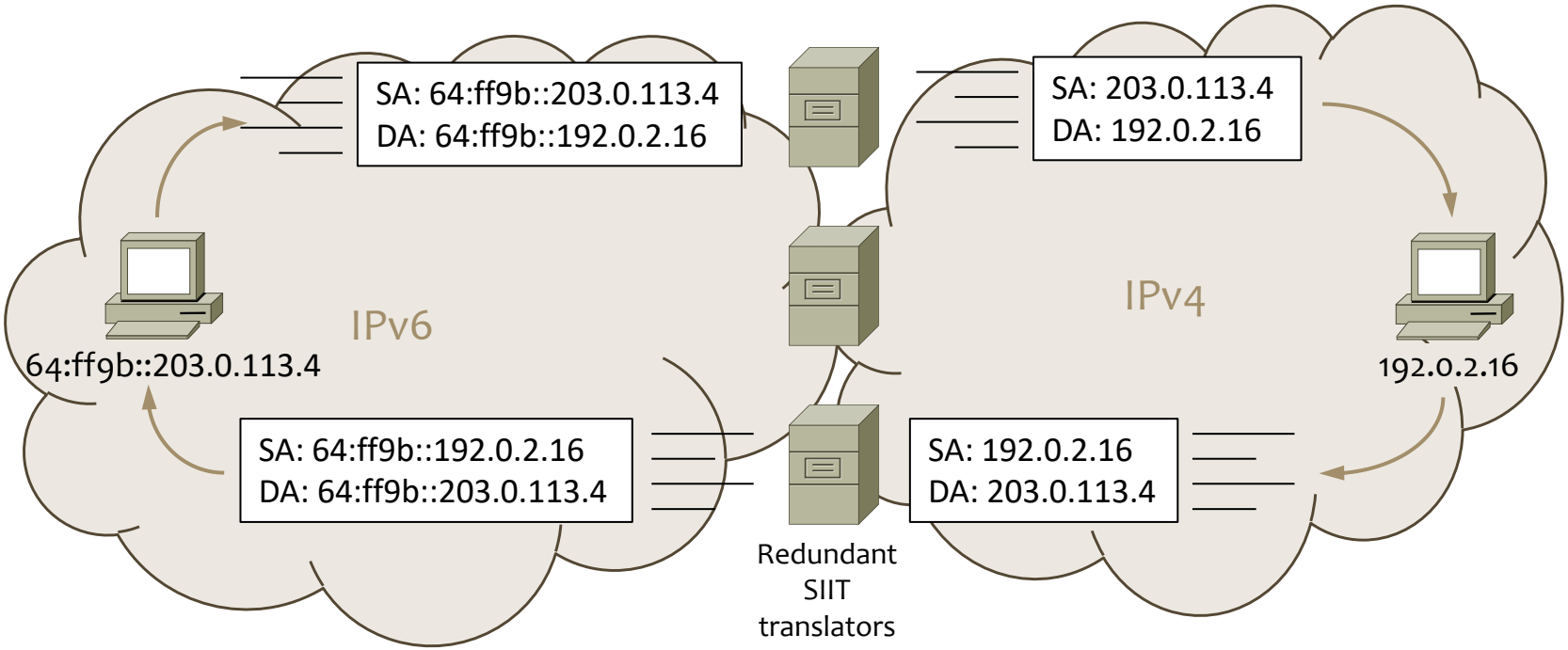
IPv4 in an IPv6 world: NAT64/DNS64

- Situations where NAT64/DNS64 cannot help:
  - IPv4-only software
    - Especially older, no longer maintained software
  - IPv4 literals (no DNS lookup is performed)
    - Badly written HTML:
      `<a href="http://193.136.39.12/~rprior/">My homepage</a>`
    - Applications conveying IPv4 addresses in application layer messages (e.g., FTP)
  - IPv4 networks behind IPv6-only ISP networks
- These cases require
  - IPv4-capable clients (perhaps dual-stack)
  - Additional components
    - 464XLAT combines NAT64 and SIIT to address these cases

SIIT

- Stateless IP/ICMP Translation (RFC 7195)
- Translates headers of
  - IP packets
  - IP packet fragments inside ICMP messages (for transparency)
- Comes in two flavors:
  - "Traditional" SIIT (entire IPv4 address embedded in IPv6; RFC 6052)
  - SIIT with Explicit Address Mappings (EAM; RFC 7757)
- Advantages of stateless translation:
  - No need to maintain per-flow state
  - Easier load distribution — translator for outgoing and incoming packets needs not be the same
- Disadvantage: the 1:1 mapping between IPv4 and IPv6 addresses wastes scarce IPv4 addresses

SIIT: Example ("traditional" SIIT)



The translator may be different for each packet, they just need to be identically configured.

SIIT

- "Traditional" SIIT embeds the entire IPv4 address in an IPv6 address with given prefix. Examples:
  - 192.0.2.16 ↔ 64:ff9b::192.0.2.16 (prefix = 64:ff9b::/96)
  - Prefix lengths shorter than /96 are possible:
    192.0.2.16 ↔ 2001:db8:c000:216:: (prefix = 2001:db8::/32)
- SIIT-EAM uses configured host-specific or block mappings. Examples:

IPv4	IPv6	
192.0.2.1	2001:db8:aaaa::	Host-specific mappings
192.0.2.2/32	2001:db8:bbbb::b/128	
192.0.2.16/28	2001:db8:cccc::/124	Block mappings
192.0.2.128/26	2001:db8:dddd::/64	
192.0.2.192/29	2001:db8:eeee:8::/62	
192.0.2.224/31	64:ff9b::/127	

IPv4 in an IPv6 world: 464XLAT

- 464XLAT is an architecture combining Stateful NAT64 with an additional, stateless translator (SIIT)
  - CLAT: Client-side transLATor (SIIT)
  - PLAT: Provider-side transLATor (Stateful NAT64)
- Addresses the use cases not covered by NAT64/DNS64

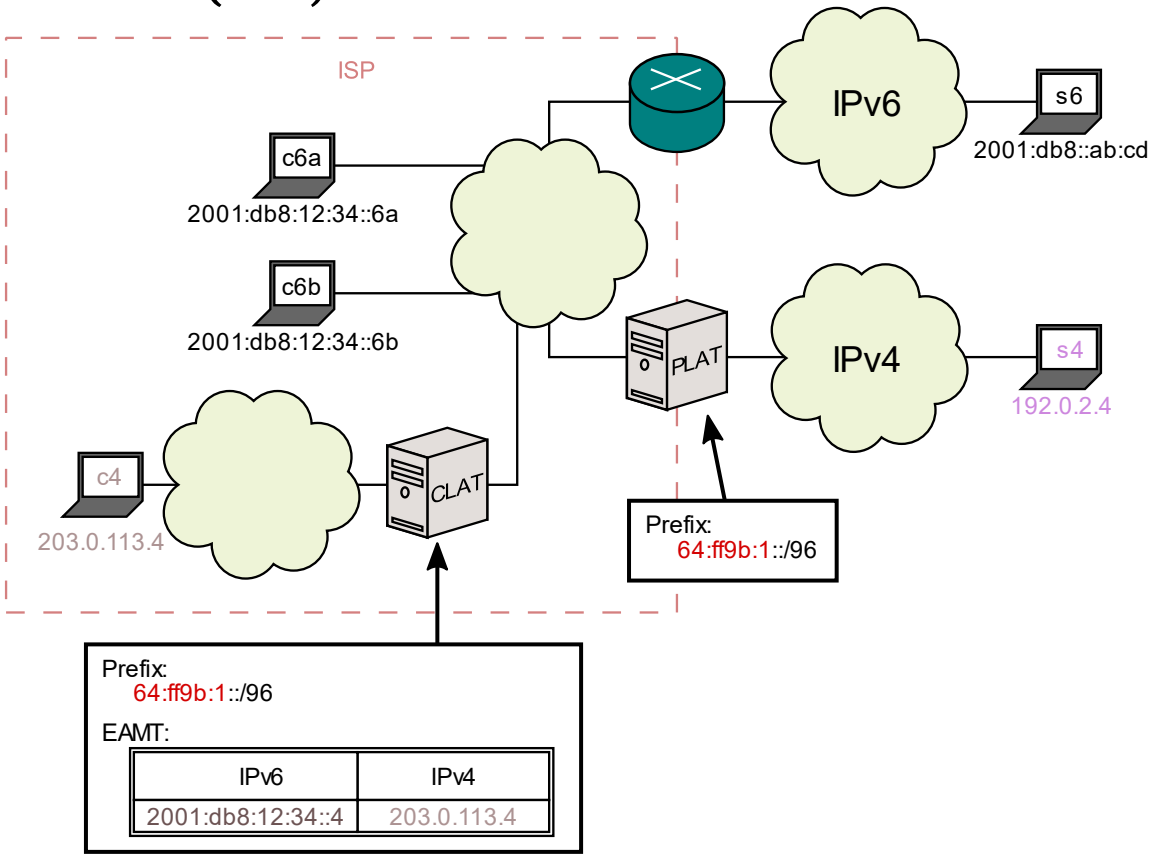


Image adapted from
https://www.jool.mx/en/intro-xlat.html

464XLAT Example: Translations

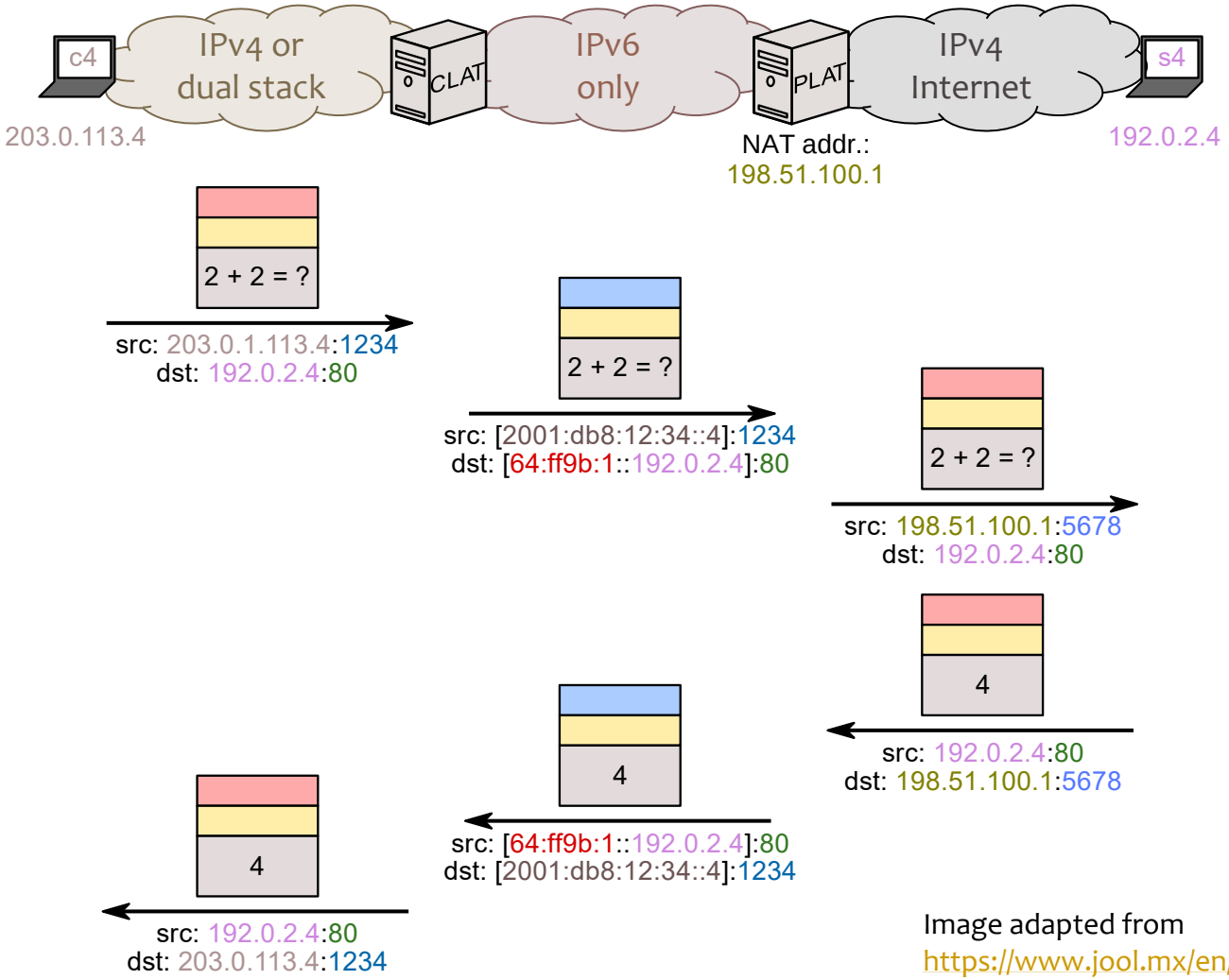
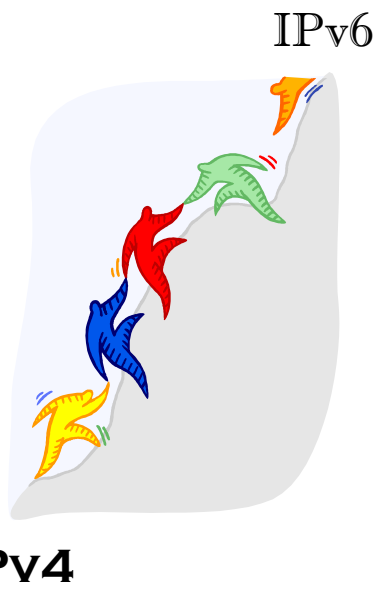


Image adapted from
https://www.jool.mx/en/intro-xlat.html

IPv4→IPv6 transition mechanisms

- Several other mechanisms were defined
  - See, e.g., Wikipedia IPv6 transition mechanisms
- VPN solutions may also be used to provide tunneling
- Eventually, everything will converge to IPv6



The end