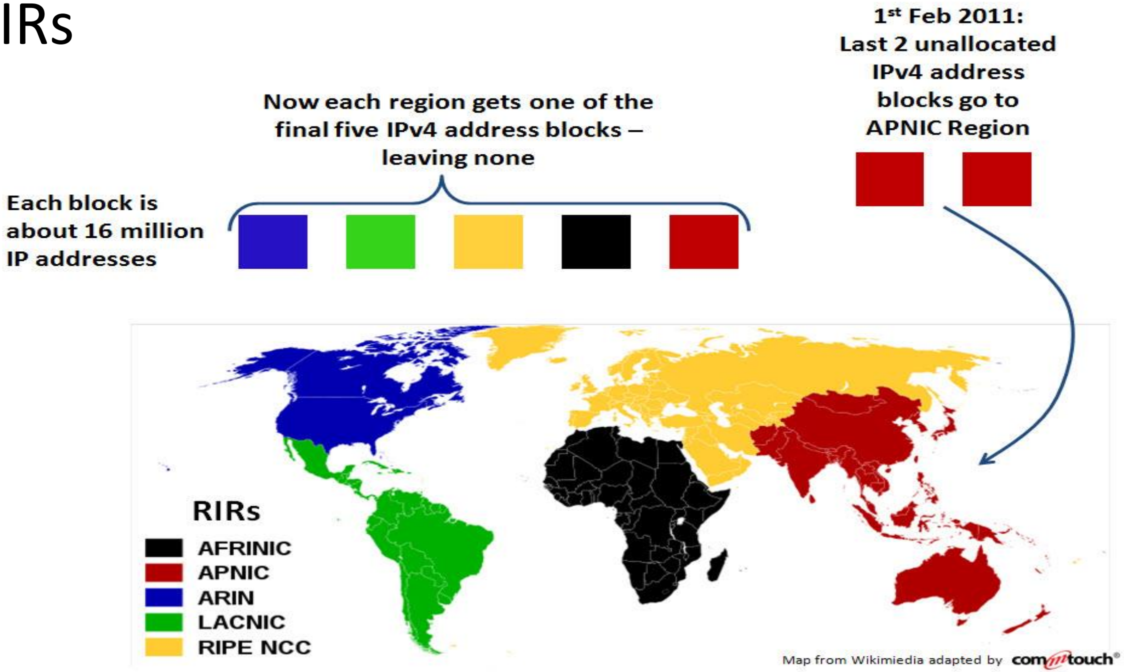
Networking Technologies and Applications

Exhaustion of IPv4 addresses

- No problem in the US
  - „Internet Heaven”
- Serious problem everywhere else
  - Many European or African countries received just a Class C address (254 addresses)
- Fast development of the Internet outside Northern America
  - Asia (2.5 billion people), Eastern Europe (250 million), Africa (800 million), South and Latin America (500 million)
- New communication devices need IP addresses
  - Mobile phones, PDAs, sensors, cars, etc.
- The exhaustion of IPv4 addresses was always projected for the next month/years (for more than 10 years)

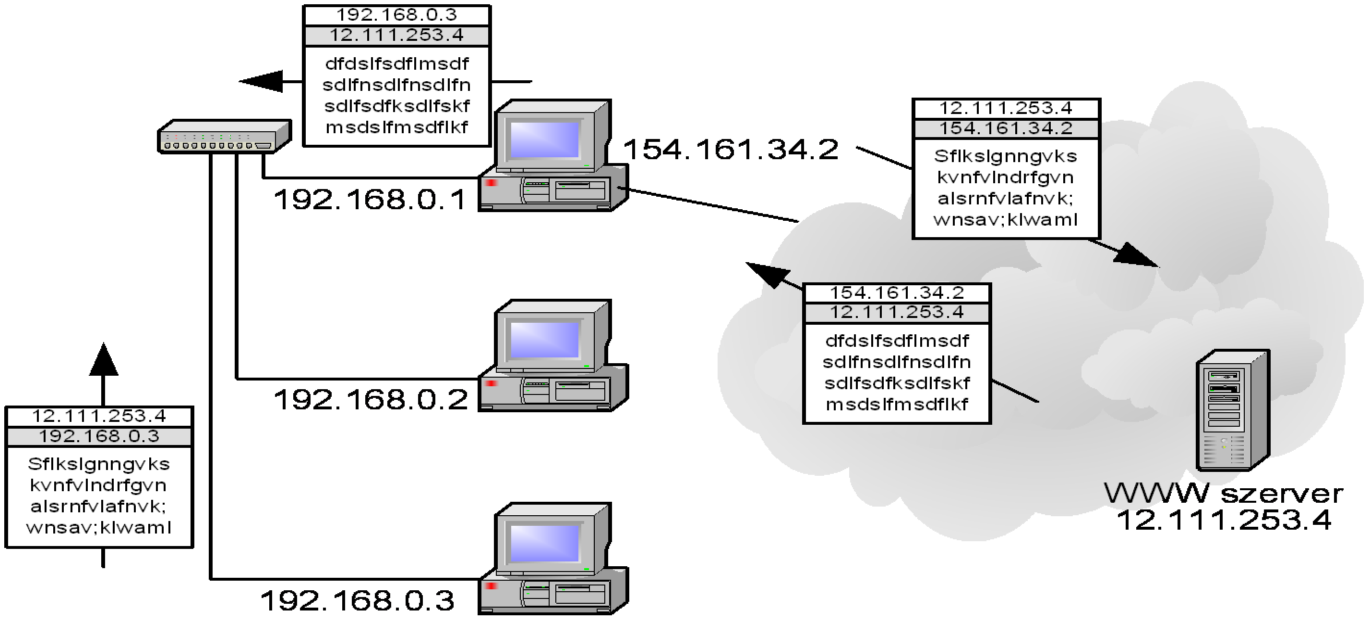
Is the Internet „full”?

- On February 1st, 2011, the last /8 IPv4 address block allocated to the RIRs
- Last allocated blocks by RIRs
  - APNIC – April 2011
  - RIPE – Sept. 2012
  - LACNIC – June 2014
  - ARIN – Sept. 2015
  - AfriNIC – 2018?



Private addresses

- Using private addresses, that are not globally unique
  - Private addresses are not „visible" from the Internet
  - Need for network address translation



NAT problems

- Just an intermediate solution
  - Cannot establish a connection from outside with a host behind a NAT
    - Should be initiated from behind the NAT
  - More and more applications require public IP addresses
    - VoIP, videoconferencing, network games
  - Many protocols do not work on a network with NAT

IPv6 chronology

- **TUBA** (1992)
  - TCP and UDP over Bigger Addresses
  - Based on the OSI CLNP (Connection-Less Network Protocol)
  - abandoned
- **SIPP** (1993)
  - Simple IP Plus
  - 64 bit addresses
- **IPng**, based on an extended SIPP version (1994)
  - 128 bit addresses
  - From December 1995 officially called IPv6

IPv6 addressing scheme

- The IPv6 address pool is huge
  - 2^{128} = 340.282.366.920.938.463.463.374.607.431.768.211.456
  - 67 billion billion addresses for each cm^2 on the surface of the Earth
  - 10^{30} address for each person on the Earth
  - The address distribution and routing requires a hierarchical structure

IPv6 addressing scheme

- IPv6 addressing quite similar to IPv4
  - 128 bit long addresses, instead of 32 bits
- Three address types:
  - Unicast addresses
    - Identify a unique interface
  - Multicast addresses
    - Identify a group of interfaces, each of these will receive the message
    - Replace the broadcast addresses as well
  - Anycast addresses
    - Identify a group of interfaces, message will be delivered to one of these interfaces

Writing IPv6 addresses

- 128 bits = 16 bytes = 32 x 4bits = 32 hexadecimal digits grouped in 8 segments

  FECD:BA98:0000:0000:00CD:BA98:0000:3200
- The opening 0 digits in each segment can be neglected

  Instead of FECD:BA98:0000:0000:00CD:BA98:0000:3200

  we write FECD:BA98:0:0:CD:BA98:0:3200
- Adjacent 0 segments can be neglected, if there is only one such case in an address

  FECD:BA98::CD:BA98:0:3200
- Network prefix is encoded as in case of IPv4 CIDR

  entire IPv6 address/prefix length in bits

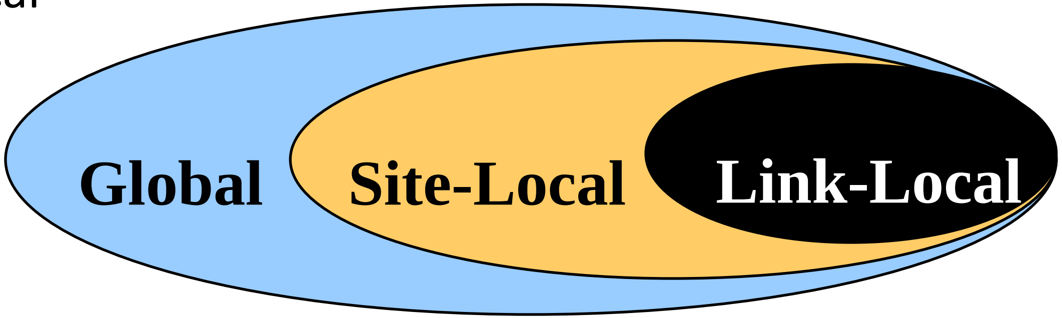
  12AB:0000:0000:CD30:FFFF:DEC8:0000:0000/60

  12AB:0:0:CD30:0:0:0:0/60

  12AB:0:0:CD30::/60

IPv6 addressing scheme

- One interface might have several addresses, with different scopes:
  - Link Local
  - Site Local
  - Global



Unicast addesses

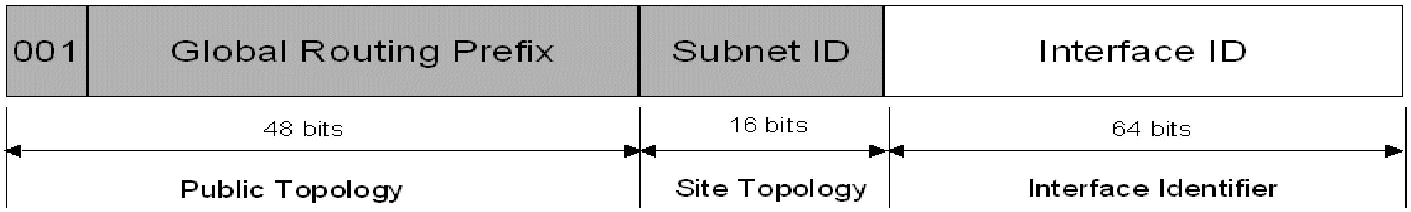
- Valid over a limited scope
  - Scoped address
  - Novelty in IPv6
- Scope = local link
  - For communication among nodes on the same link
  - Unique only on that link, cannot be used for communication outside the link
  - Automatically configured on each interface
  - Each IPv6 node has an initial address with which it can start communicating
    - Neighbor dicovery, router discovery
  - Format:
    - FE80:0:0:0:<interface identifier>
    - Interface ID – EUI (64) address
      - Extending the previous 48 bit MAC address

Unicast addresses

- Scope = site local
  - Used for communication inside the same site
  - Routers will not forward it outside the site (to the Internet)
  - Similar to IPv4 private addresses
  - Not automatically configured
  - Format:
    - FEC0:0:0:<subnet id>:<interface id>
    - Subnet id = 16 bit = 64K subnet
  - Allows the addressing of an entire organization (company, university)
    - E.g. allocate site-local addresses to the entire network
    - Renumbering when connecting to an IPv6 network
      - We change the first 48 bits (to a site ID)
    - Renumbering when connecting to a new service provider

Unicast addresses

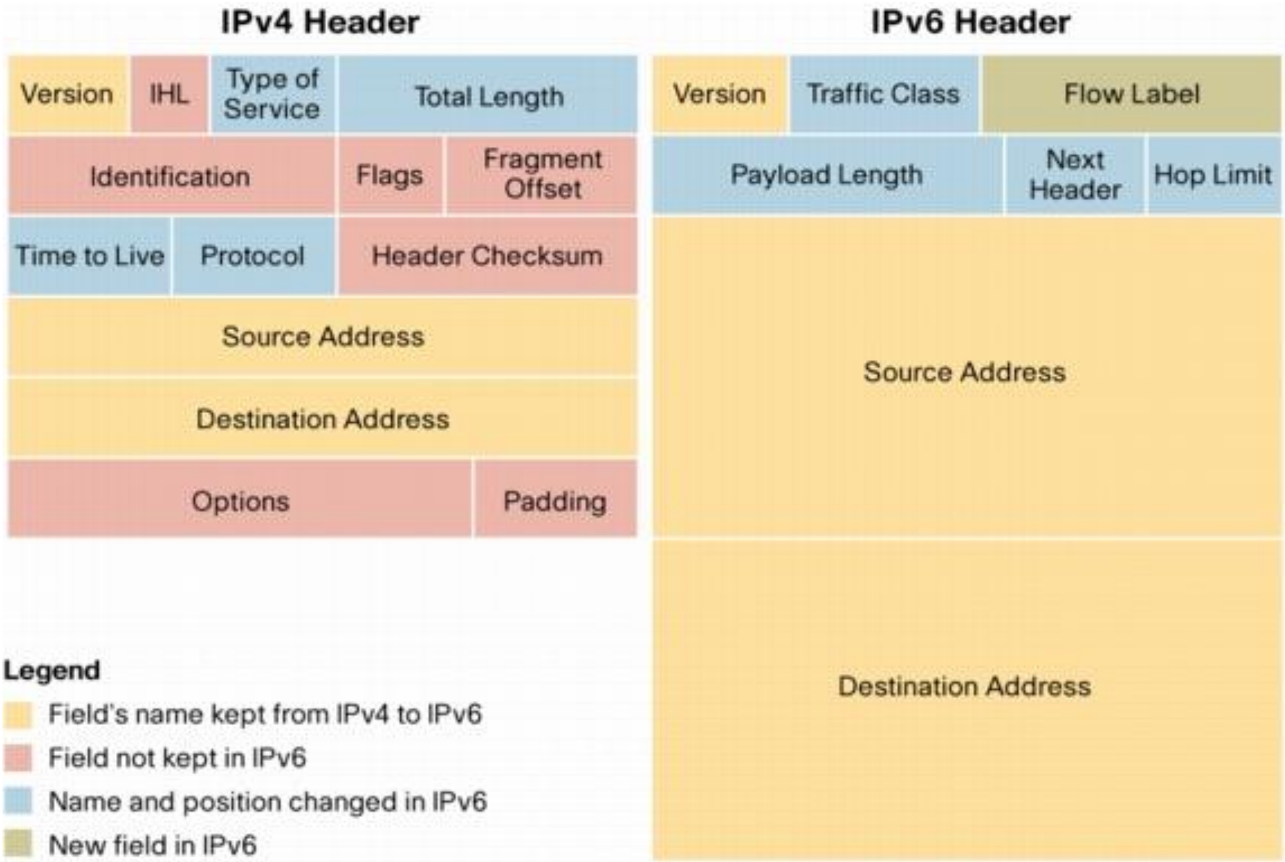
- Global Unicast Address
  - Used for global communication
  - Hierarchical global prefix
    - Structured by RIRs and ISPs
  - Subnet ID
    - Hierarchically structured by the network administrator
  - Interface ID



Multicast addresses

- Instead of broadcast addresses
- Scoped addresses
  - Node, link, site, organisation, global
- Format
  - FF<flags><scope>::<multicast group>
- Flag:
  - 0 – permanent
  - 1 – dynamic
- Scope:
  - 1 – node
  - 2 – link
  - 5 – site
  - 8 – organisation
  - E – global
- E.g.
  - FF02::1 – all nodes on the local network
  - FF02::2 – all routers on the local network

Format of the mandatory fixed IPv6 header



IPv6 mandatory fixed header

- Version (4 bits)
  - IP version
- Class – priority class (8 bits)
  - Defines the priority of the packet
    - DSCP field in IPv4
  - Used for providing Quality of Service (discussed later)

IPv6 mandatory fixed header

- Flow Label
  - Associated to traffic with special QoS requirements
    - 20 bit long
  - Can be used in router caches to decrease the processing time
    - A packet first arrives to the router
    - The fow label is stored in the cache
    - When the next packet with the same flow label arrives…
      - No need to look up the destination address in the routing table
      - Packet immediately routed based on the flow label
- Payload Length (16 bits)
  - Length of the useful data, in bytes

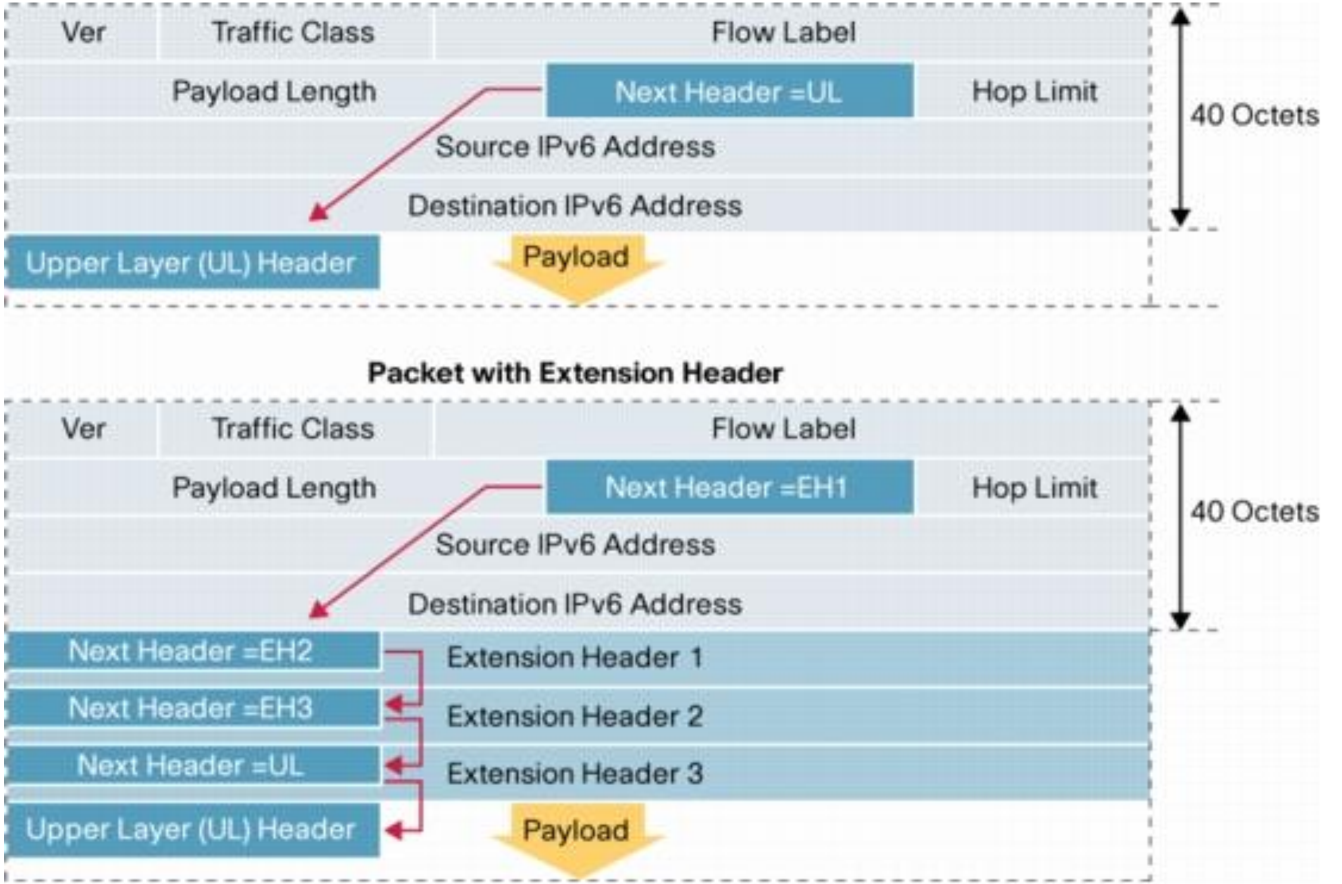
IPv6 mandatory fixed header

- Next-Header (8 bits)
  - Identifies the header directly following the fixed header
    - It might be an extension header, or a higher layer protocol header (e.g., TCP)
- Hop Limit (8 bits)
  - Provides how far a packet can travel
  - Same as the Time To Live (TTL) field in IPv4
- Source Address (128 bits)
  - Address of the original source of the packet
- Destination Address (128 bits)
  - Not necessarily the address of the last receiver, if the packet also contains a Routing Header

IPv6 extension headeres

- IPv6 packet – starts with a 40 byte long mandatory fixed header
- Extra information related to the intermediate network encoded in Extension Headers
- Most of the extension headers are not processed by the intermediate routers, only by the final destination
- Each extension header has a special value for the next header field
  - Many extension headers can be used in parallel
  - The value of the last next header field identifies the upper layer protocol
  - The header can be of any length

IPv6 chained extension headers



IPv6 extension headers

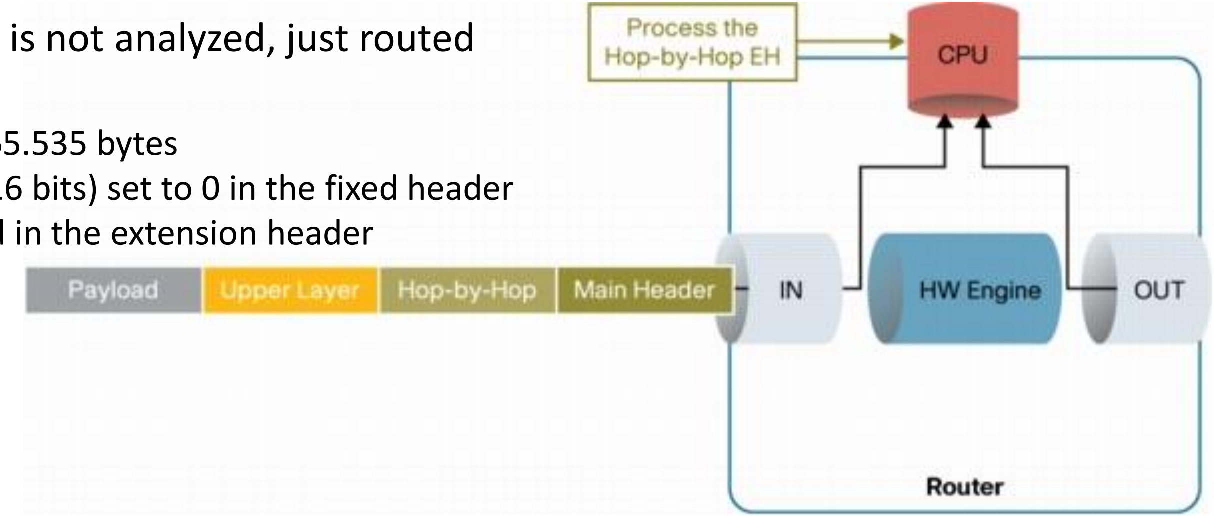
- Header order
  - Important: extension headers should respect the suggested order
    - Easier for routers to process the packet
    - In most cases the routers process only the hop-by-hop options and the routing header
  - Exception: destination option
    - Right before the higher layer header
    - If we want the intermediate routers to process the destination option header, we should put it right before the routing header, and they should be processed together.
    - A packet might contain a destination option headers in both locations

IPv6 extension headers

- The suggested header order:
  - IPv6 Header
  - Hop-by-hop Options Header (type = 0)
  - Destination Options Header (1)
  - Routing Header (type = 43)
  - Fragment Header (type = 44)
  - Authentication Header (type = 51)
  - Encapsulating Security Payload (ESP) (type = 50)
  - Destination Options Header (2) (type = 60)
  - Upper Layer Header (e.g. TCP or UDP)

IPv6 extension headers

- Hop-by-hop Options Header
  - Contains IP options for the intermediate routers
  - Each intermediate router should analyze and process the Hop-by-hop Header
  - Router Alert option alerts tranzit routers
    - If the packet packet contains information that should be processed by an intermediate router
    - Otherwise the packet is not analyzed, just routed
  - IPv6 jumbogram option
    - For packets larger than 65.535 bytes
    - The payload length (on 16 bits) set to 0 in the fixed header
    - The true length specified in the extension header



IPv6 extension headers

- Routing Header
  - In the normal case the source of the IP packet leaves the routing task to the network
  - In case of source routing, the source will specifiy a path with router addresses
    - The entire list in the Routing Header (e.g., A, B, C, D)
    - The destnation address is always the address of the next router specified in this field, except the last router
    - Each router modifies the destination address, before forwarding the packet

IPv6 extension headers

- Fragment Header
  - In IPv4 fragmentation and reassembly automatically, if explicitely not forbidden
    - **Don't Fragment flag**
  - In IPv6 packets are not fragmented by default
    - If the packet is too large for the transmission medium, it is dropped and an ICMP (Internet Control Message Protocol) error message is sent
    - The source discovers the path MTU-t
      - Maximum Transmission Unit
    - Tries to send packets with a lower size than the MTU
    - If we need fragmentation, that can be done using the Fragmentation Extension Header
- Authentication Header
  - Guarantees that …
    - The recieved packet is authentic
    - It was not altered on its path
    - Comes from the specified source
- Destination Option Header
  - Contains options to be processed by the destination node

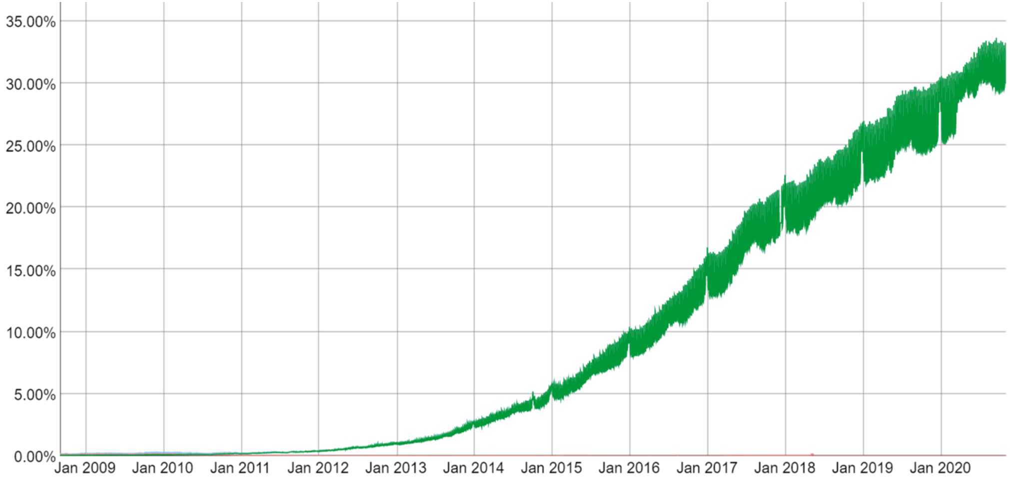
Transition to IPv6

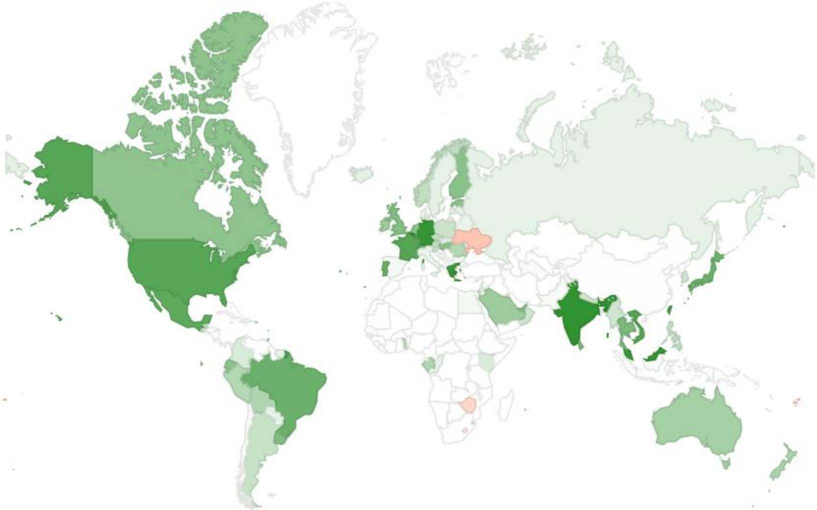
- Routing services built on IP
  - RIPv6(ng), OSPFv6 (v3), BGPv6
- Network and transport layer protocols built on IP
  - TCPv6, UDPv6, RSVPv6
- Applications
  - Each application that was using directly the IPv4 addresses is not independent from the lower layers, so IPv6 support should be implemented in it
- Gradual transition
  - No „D-day”
- Expectations regarding transition
  - No transition dependencies
    - The transition of a given node can be done independently from the others
    - The most important aspect is backward compatibility
  - It should be as easy as possible for the end user
  - The different transition solutions should be appliable independently of each other
    - At least at the level of the different domains

IETF paranthesis

- Internet Engineering Task Force (IETF)
  - Internet Drafts (valid for 6 months)
  - Request for Comments (RFCs)
    - No real comments requested
    - These are the actual standards
- Internet Research Task Force (IRTF)

IPv6 deployment as seen by Google



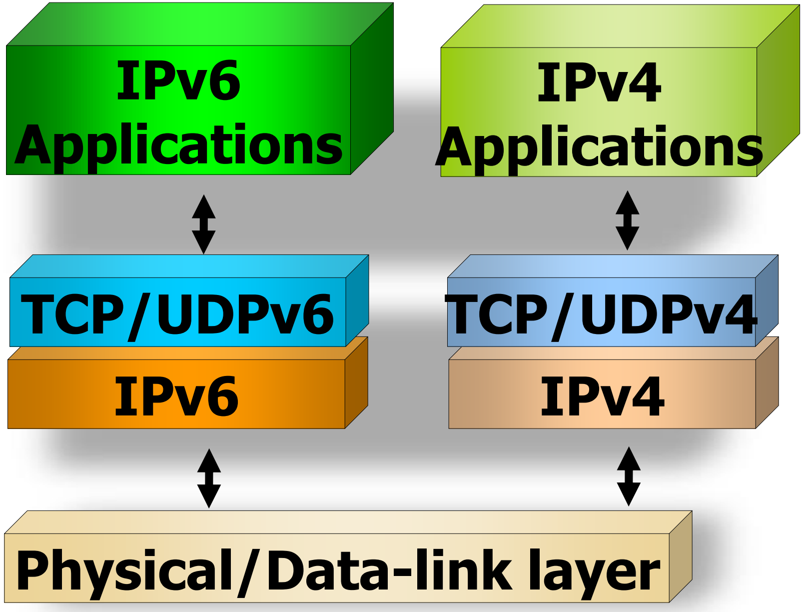


Transition solutions

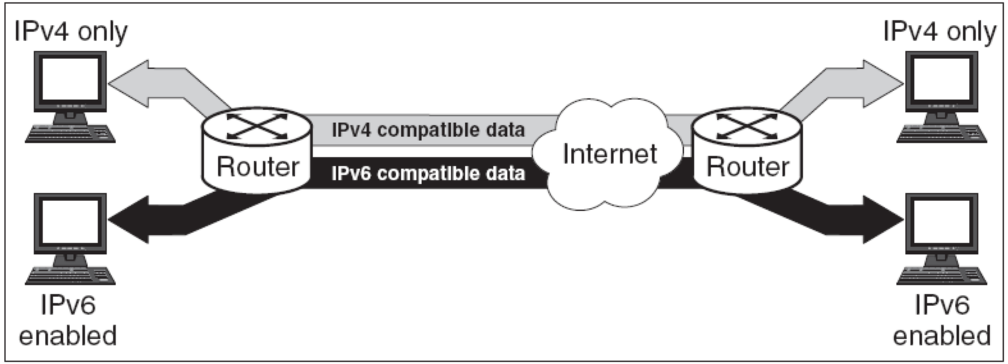
- Dual Stack
  - Both IPv4 and IPv6 stack on the same device
- Tunnels
  - Initially tunneling IPv6 packets in IPv4 domains
  - Later, tunneling IPv4 packets in IPv6 domains
- Protocol translation
  - Headers containing protocol information should be translated into different protocol headers, based on certain translation rules
  - IPv6 <-> IPv4

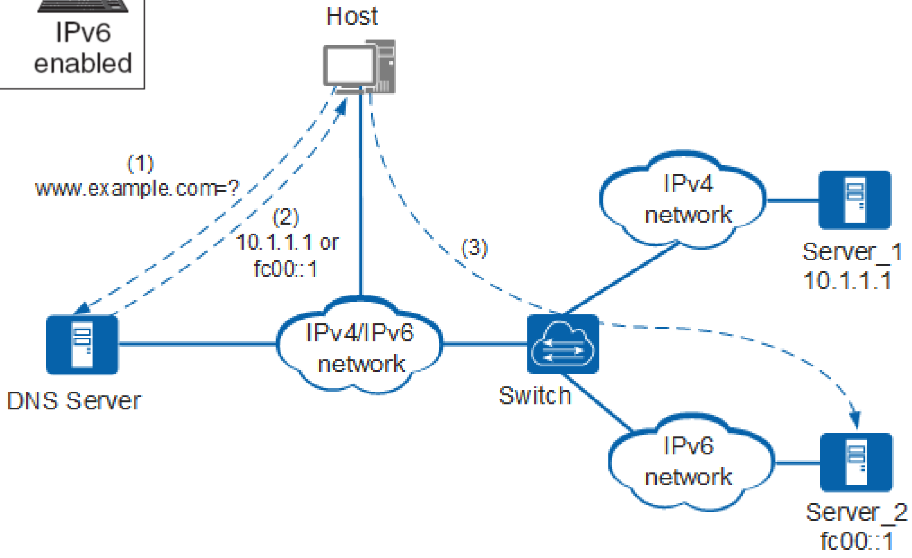
Dual Stack

- The first step towards deploying IPv6 is deploying some nodes that support IPv6 as well, next to IPv4
  - They have a double stack strategy
    - Use IPv6 to communicate with other IPv6 systems
    - Can switch back to IPv4 mode to talk to IPv4 systems



Dual stack





The host might send an IPv4 (class-A) or an IPv6 (class-AAAA) DNS query to the DNS server

- The server will answer with either one or both IP addresses
- By default trying first the IPv6 address

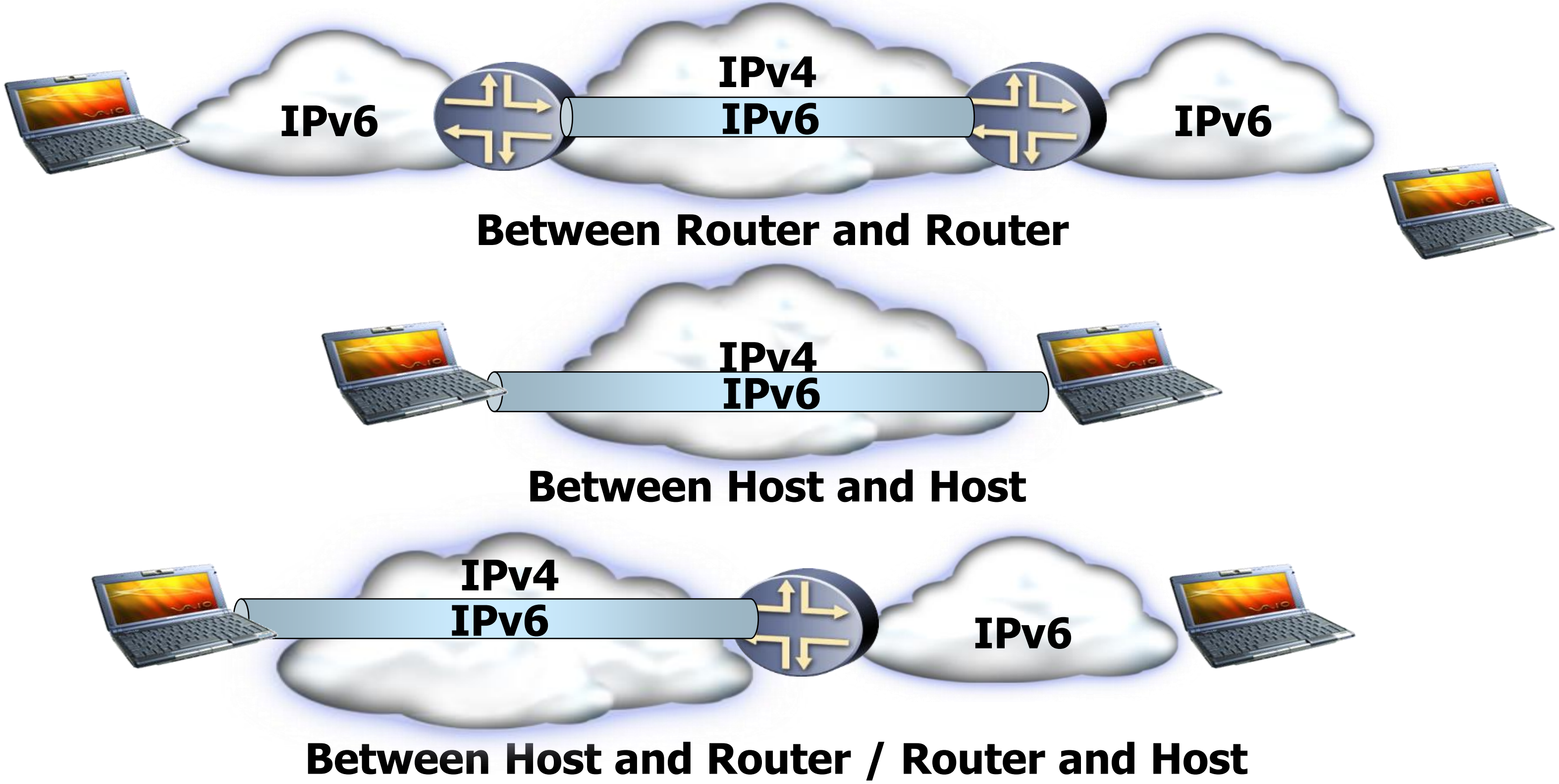
Dual stack

- Advantages
  - Easy to install, configure, maintain
  - The entire functionality of IPv6 can be exploited
  - Any two nodes can communicate exclusively with IPv4 or IPv6 packets
  - Transparent transition for the end users
- Drawbacks
  - Not scalable: each node should have an IPv6 and an IPv4 address, the limitation of the IPv4 address domain obstructs its spreading
  - The size of the routing tables is increased in the routers
  - Not flexible: no communication possibility between nodes speaking just IPv4 and just IPv6

Tunneling

- IPv6 packet encapsulated inside an IPv4 packet
- The tunnel endpoints manage the encapsulation
- The process transparent to the intermediate nodes
- Configured tunnels
  - The tunnel endpoints are explicitely configured
  - They are dual stack nodes
- Automatic tunnels
  - The tunnel endpoint are automatically discovered by the network
  - Tunnel Brokers (RFC3053)
  - 6to4 (RFC3056)
  - ISATAP (Intra-Site Automatic Tunnel Addressing Protocol)
  - 6over4 (RFC2529)
  - Teredo: support tunnels through IPv4 NAT

Tunneling


IPv6
IPv4
IPv6
IPv6
Between Router and Router
IPv4
IPv6
Between Host and Host
IPv4
IPv6
IPv6
Between Host and Router / Router and Host

Translators

- Network layer translators
  - SITT (Stateless IP/ICMP Translator Algorithms) (RFC2765)
  - NAT-PT (Network Address Translator-Protocol Translator) (RFC2766)
  - BIS (Bump int the Stack) (RFC2767)
- Transport layer translator
  - TRT (Transport Relay Translator) (RFC3142)
- Application layer translators
  - BIA (Bump in the API) (RFC3338)
  - SOCKS64 (RFC3089)
  - ALG (Application Level Gateway)

Network layer translators

- The IPv4 messages are translated into IPv6 messages, and vice-versa (especially the headers)