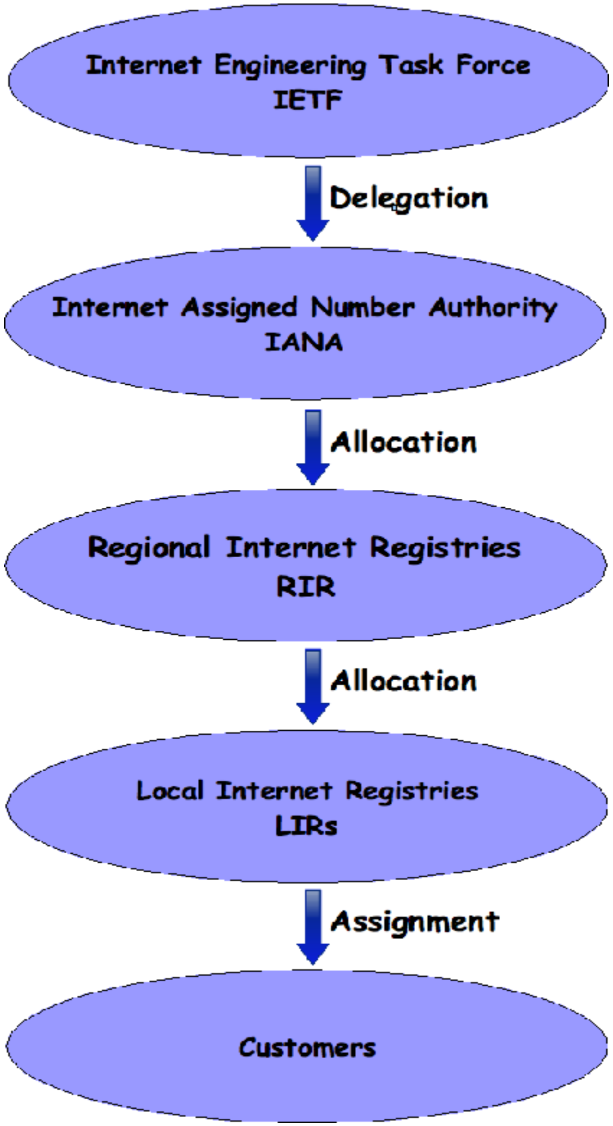
IPv6

What, Why, How

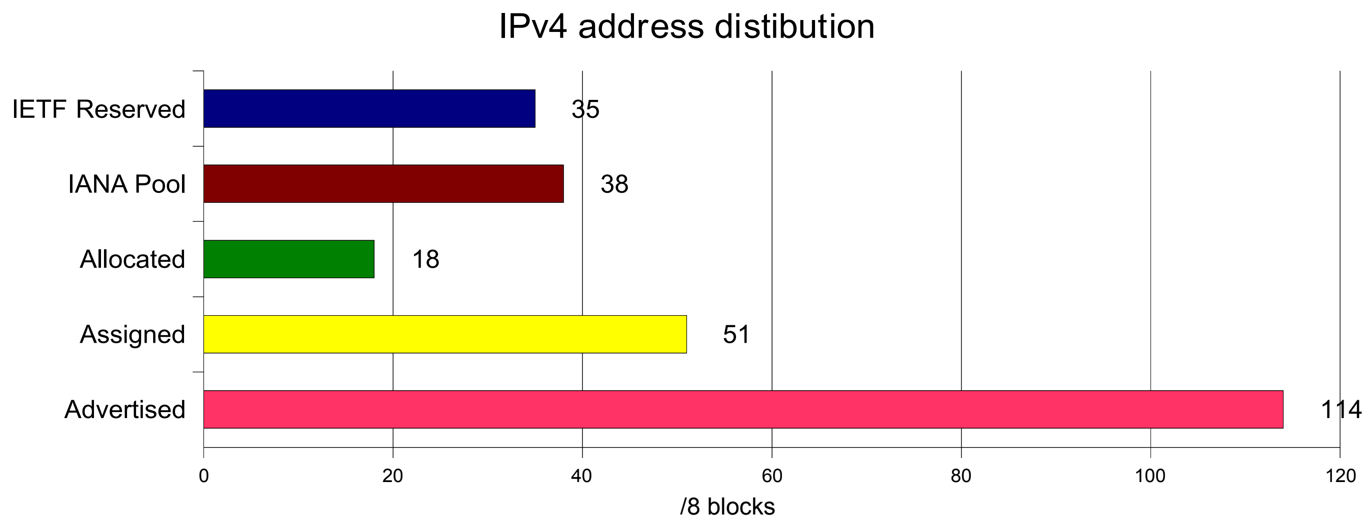
Jen Linkova aka Furry
furry - at -openwall.com
Openwall, Inc
http://www.openwall.com

Revision 1.0

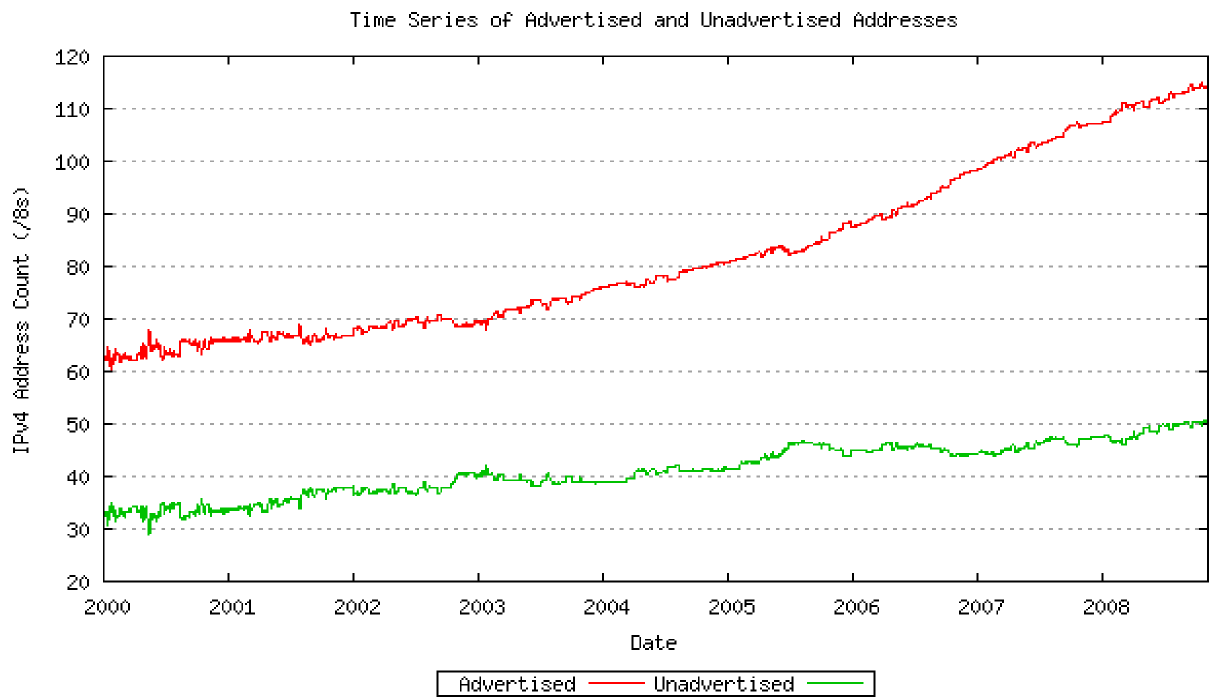
IPv4 Address Distibution

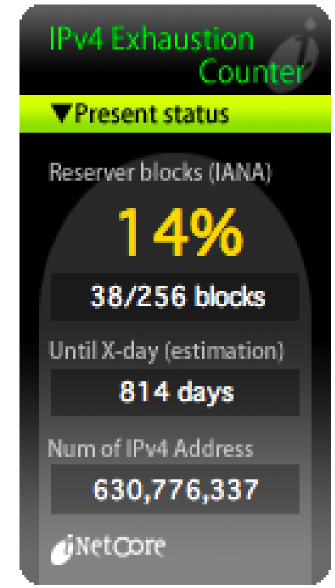


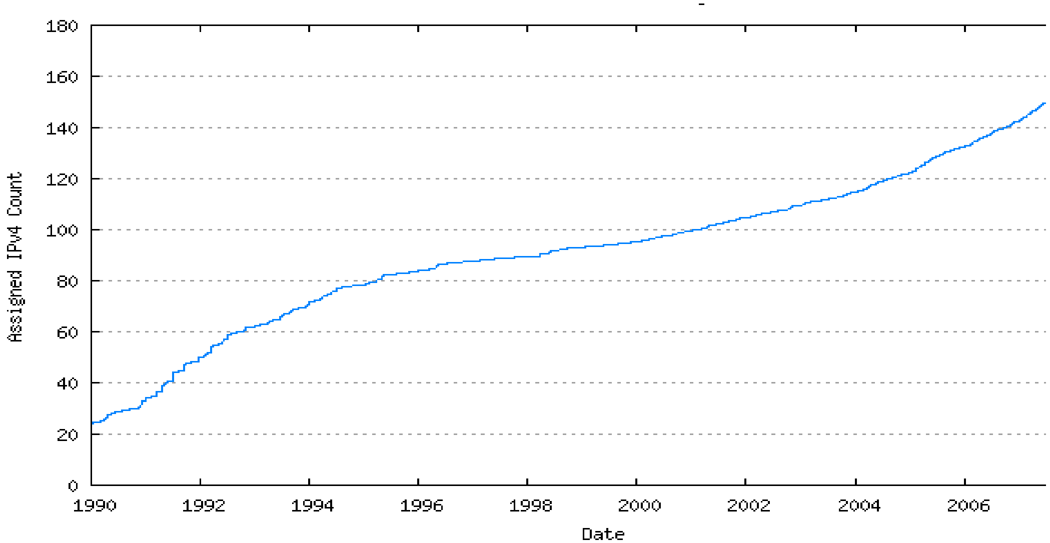
- 32-bit number
- 4 294 967 296 addresses
- 256 /8 network blocks

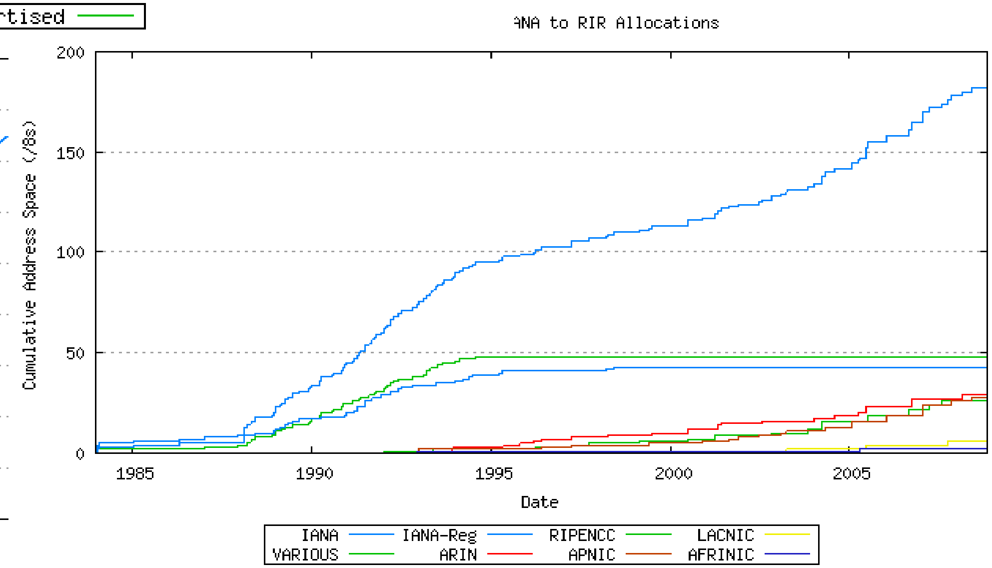


Lies, Damned Lies, Statistics









Source: «IPv4 Address Report», http://www.potaroo.net/tools/ipv4/

You have two choices: spend less...

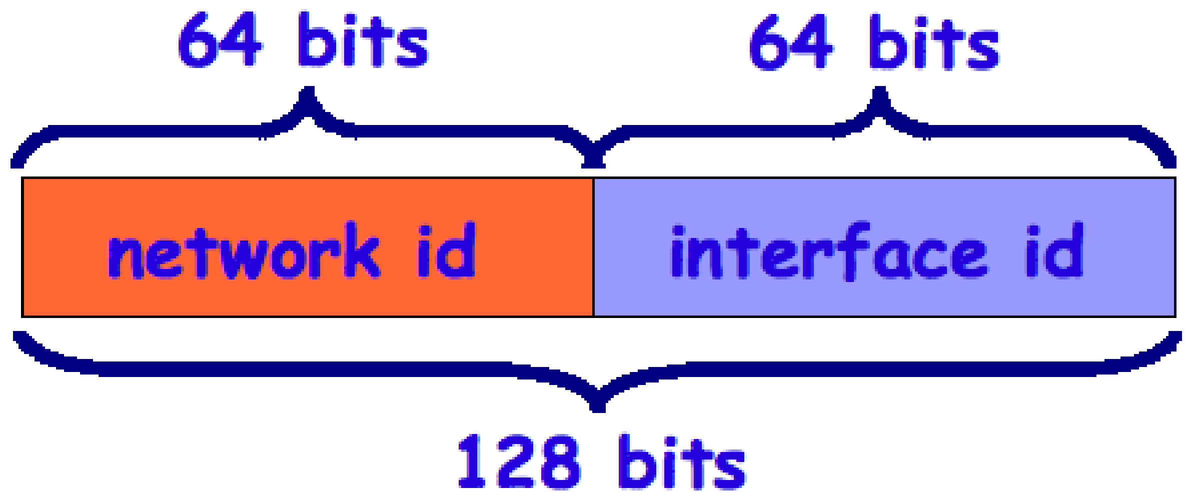
- Address allocation policy for LIR (/19, /20, /21...)
- Address translation (NAT, NAPT):
  - breaks end2end model
  - affects protocols/applications
  - provides a false sense of security

See also:

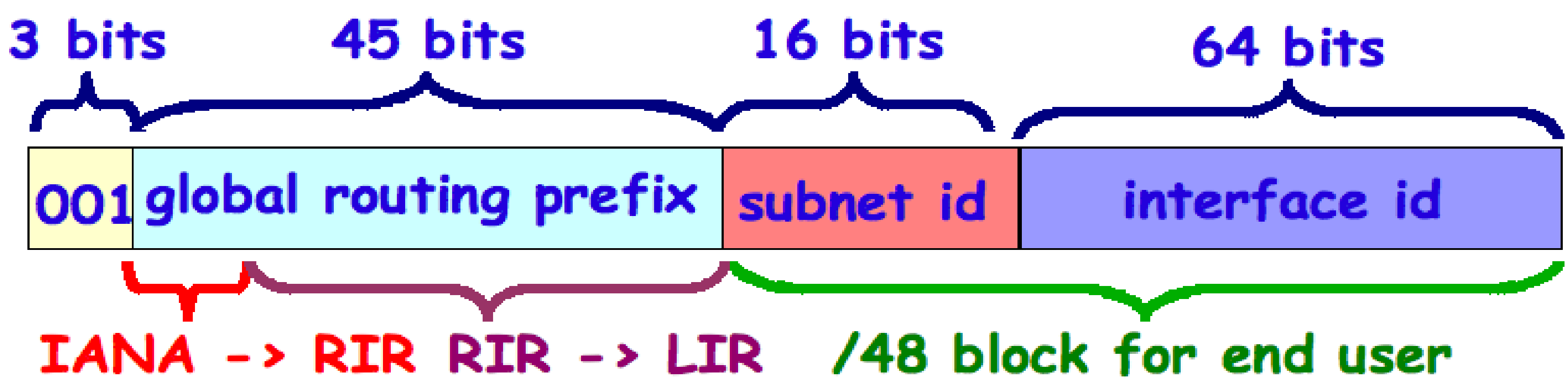
- *RFC 2775 «Internet Transparency»*
- *RFC 3027 «Protocol Complication with the IP Network Address Translator»*
- *RFC 2993 «Architectural Implications of NAT»*
- *Internet-Draft «Security implication of Network Address Translators»*

...or earn more! ;-)

IPv6 address:



- *340 282 366 920 938 463 463 374 607 431 768 211 456* total addresses
- 2^64 nodes per subnet
- fixed subnet size



Is it still enough?

- **Assume...**:
  - RIRs request new block every 18 month
- **Then...**
  - The block currently assigned by IETF (1/8TH IPv6 space) is about to run out by 2158
  - More than 5/8TH IPv6 address space will be still available
  - (NB: 000/3 and 111/3 prefixes are reserved for special use)

Source: David Conrad, General Manager, IANA, 2007
http://www.iana.org/about/presentations/conrad-buenosaires-citel-060913.pdf

IPv6 Address Format

X:X:X:X:X:X:X:X
where X = 0000 ... FFFF (hex)

- 2001:0DB8:0000:0000:0008:8000:0000:417A
- 2001:DB8:0:0:8:8000:0:417A
- 2001:DB8::8:8000:0:417A
- 2001:DB8:0:0:8:8000::417A
- 2001:db8::8:8000:417A

Examples

- loopback address

  0:0:0:0:0:0:0:1 or ::1
- unspecified address

  0:0:0:0:0:0:0:0 or ::
- special exception: IPv4-mapped

  0:0:0:0:0:FFFF:192.0.2.1

  ::FFFF:192.0.2.1
-

Find the Mistake

2001:0DB8:0000:0000:FFFF:0CA0:0000:0000

1)2001:DB8::FFFF:CA0:0:0

2)2001:db8:0:0:FFFF:0CA0:0:0

3)2001:DB8::FFFF:CA0:0:0

4)2001:db8::FFFF:ca0::

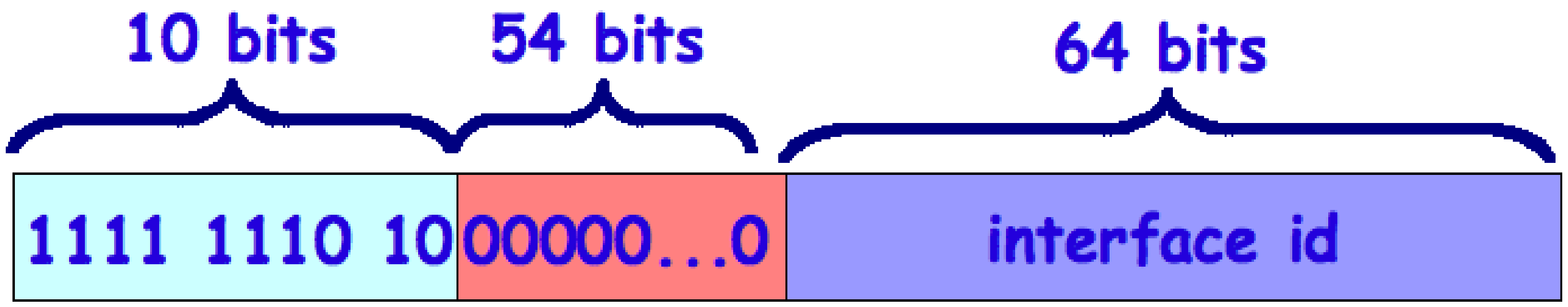
5)2001:db8:0:0:FFFF:CA0::

6)2001:db8::FFFF:CA:0:0

IPv6 Address Types

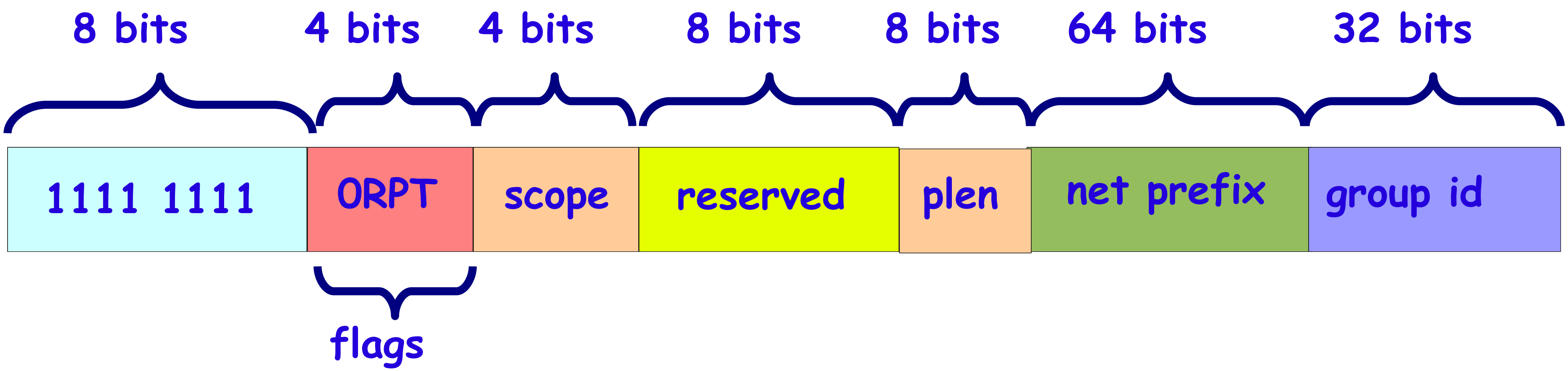
Address Type	Binary Prefix	Prefix
unspecified	000...0 (128 bits)	::/128
loopback	0000...01 (128 bits)	::1/128
link-local unicast	1111 1110 10	FE80::/10
multicast	1111 1111	FF00::/8
Global unicast	all other addresses	

Link-local Addresses



- FE80::/10 prefix
- Analogous to IPv4 169.254.0.0/16
- Automatically assigned to an interface
- Valid in the scope of the given link! Not to be routed!
- To be used for
  - auto-address configuration
  - neighbour discovery

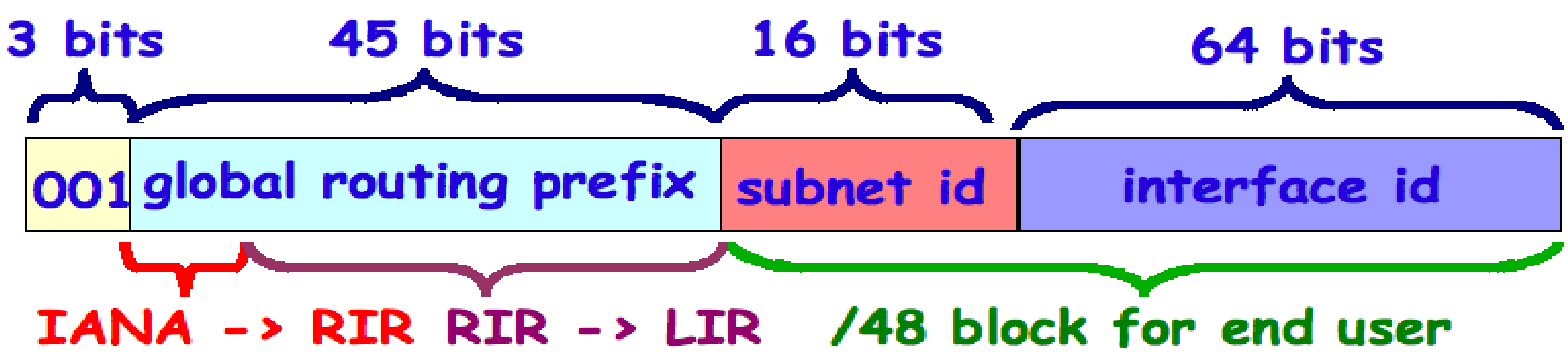
Multicast Addresses



- T=0 – permanently-assigned ("well-known") address, T=1 – non-permanently-assigned ("transient")
- Scope
  - 1 – node-local
  - 2 – link-local
  - 5 – site-local
  - 14 – global (Internet)
- Group ID identififes the mulicast group within the given scope. For example:
  - 1 – all nodes (scope = 1,2)
  - 2 – all routers (scope = 1,2,5)
  - 101 – all NTP servers
- Examples:
  - FF02::101 – all NTP-servers on the same link as a sender
  - FF02::2 – all routers on the same link as a sender
  - FF05::101 – all NTP-servers on the same site as a sender

Global Unicast

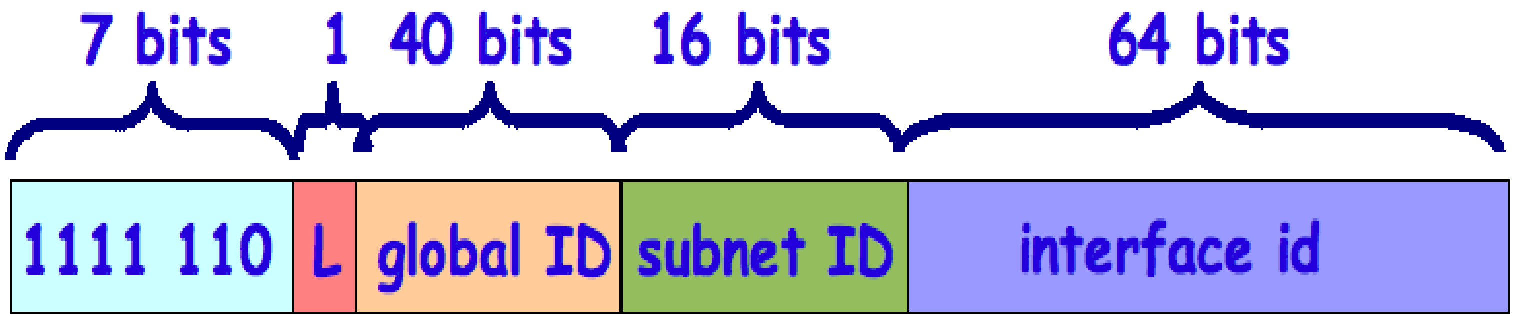
Address Type	Binary prefix	Prefix
unspecified	000...0 (128 bits)	::/128
loopback	0000...01 (128 bits)	::1/128
Ipv4-mapped	000...01111111111111111(96 bits)	::FFFF/96
ULA	1111 110	FC00::/7
Assigned to RIRs	001	2000::/3
Global unicast	all other addresses	



Unique Local Unicast Addresses (ULA)

- FC00::/7 prefix (RFC4193)
- For local communications (site or limited set of sites)
- High probability of uniqueness
- Not expected to be routable on Internet
- Well-know prefixes => Easy filtering
- If leaked outside – no conflicts with other addresses

ULA Address Format



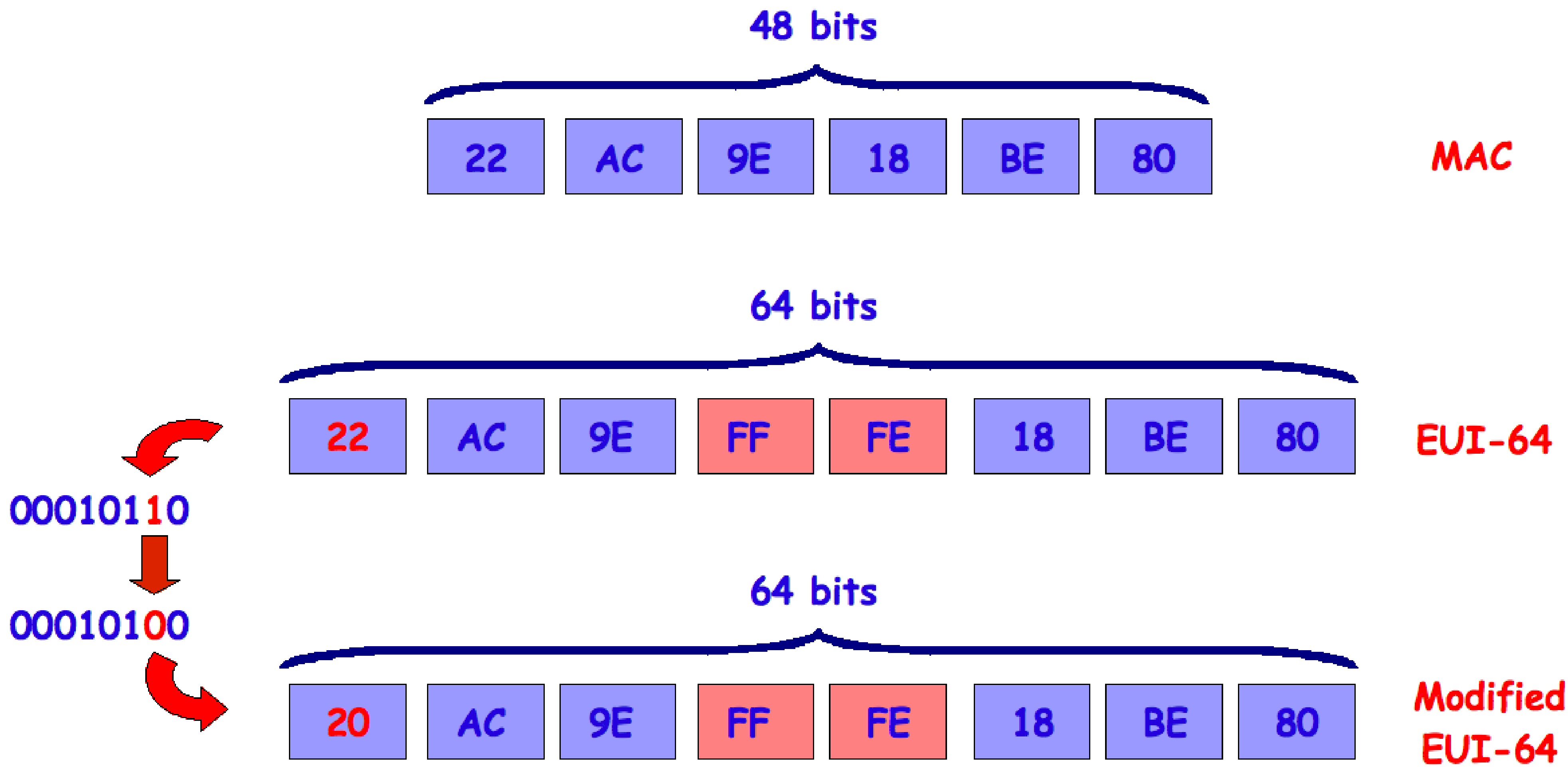
L = 1 the prefix is locally assigned
L = 0 for future use
Global ID a globally unique prefix identifier
Subnet ID the identifier of a subnet within a site

Pseudo-Random Global ID Algorithm:
1) Obtain the current time of day in 64-bit NTP format
2) Obtain EUI-64 identifier (from MAC for example) or any suitably unique ID
3) Concatenate the time (1) with the system ID (2)
4) Compute SHA-1 digest of (3) and use the least significant 40 bits as Global ID

Interface Identifier
How to configure

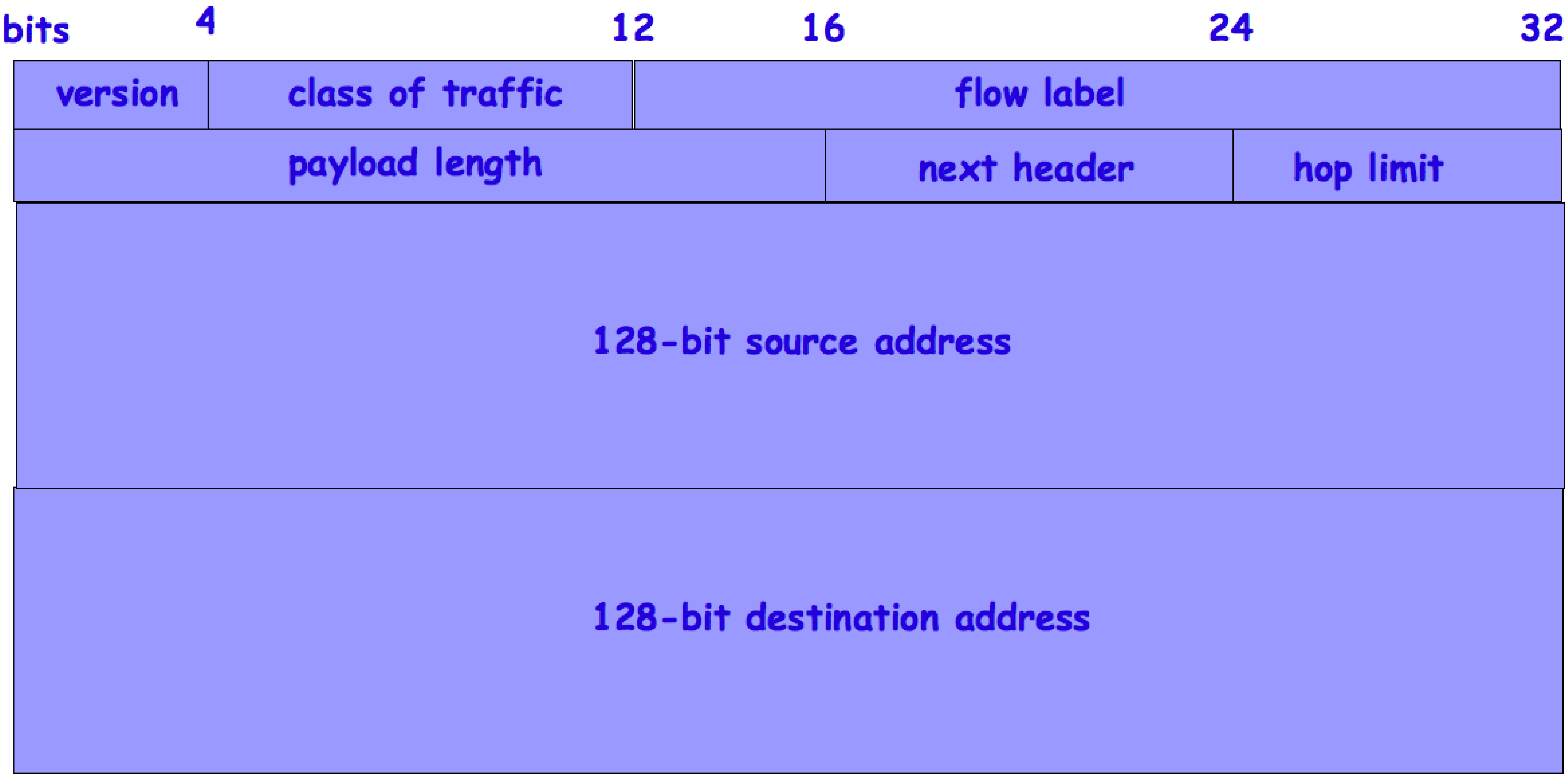
- Manual configuration
- Autoconfiguration (EUI-64-based interface ID)
- DHCPv6
- Pseudo-random interface ID
- Cryptographically generated ID

Extended Unique Identifier
EUI-64



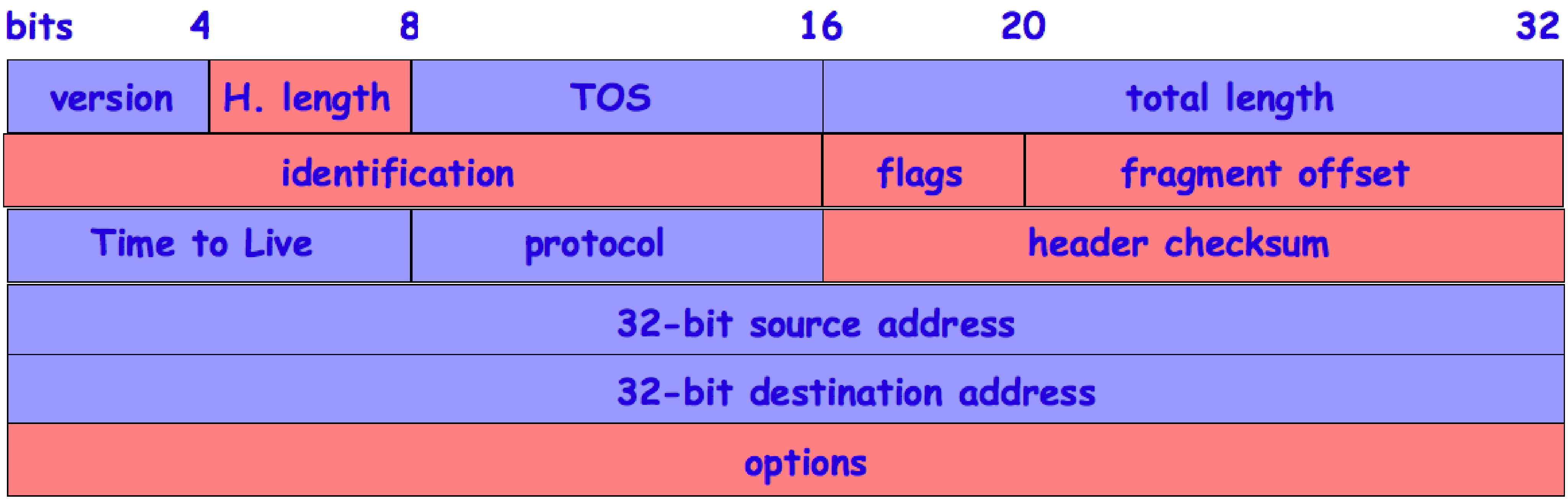
Therefore: ::1 – globally assigned EUI-64, but locally assigned MEUI-64

IPv6 Header Format



Total length: 40 bytes

IPv4 Header Format



Total length: 20 bytes + options

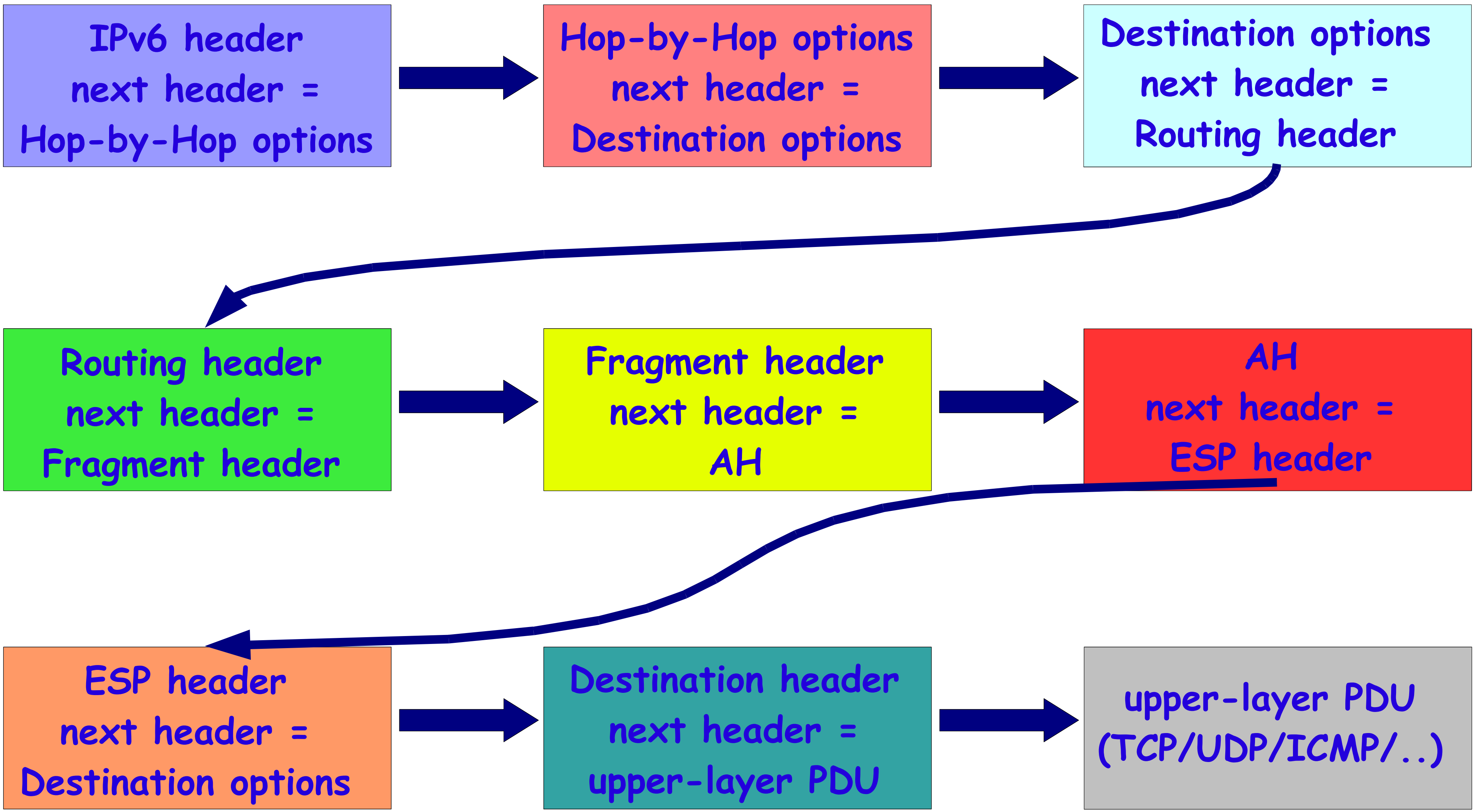
modified

deleted

IPv6 Header

- Fixed length
- All optional/additional info is encoded in **Extension Header(s)**
- Is not protected by checksum
- **Payload** Length instead of **Total** Length
- "Time To Live" field is replaced by "Hop Limit" one to better reflect its functions

Extension Headers

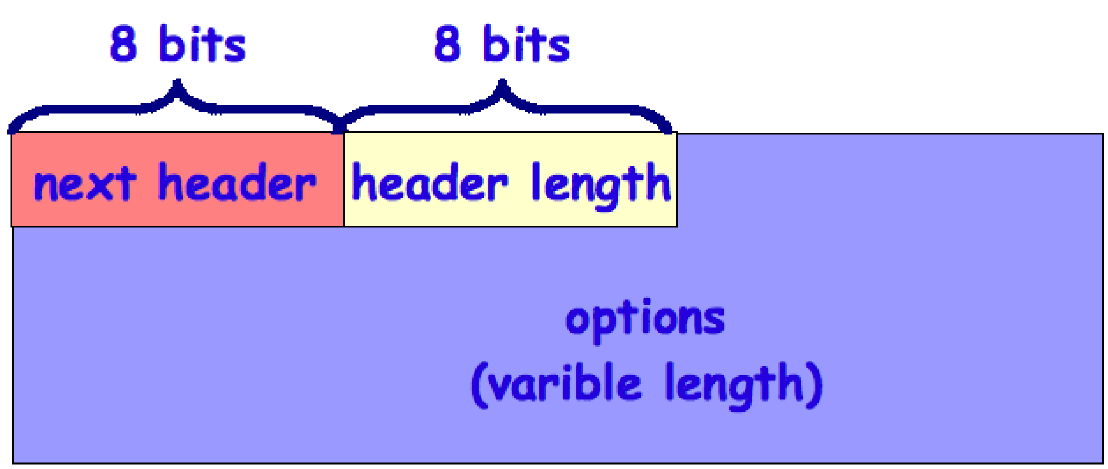


Extension Headers Processing

- All EHs (except for Hop-by-Hop options) are processed by the destination node only!
- Packet is dropped if any extension header isn't recognised
- **Recommended** order of headers (except for Hop-by-Hop Option)
- Reserved next header value: 59, «no next header»

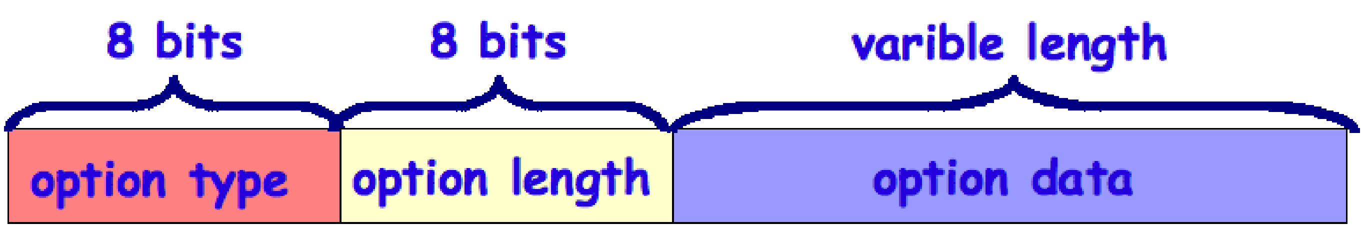
Options Headers

- Separating Hop-by-hop and Destination is useful:
  - not all options are examined along a packet's delivery path
  - encryption
  - fragmentation
- Hop-by-Hop Options: for every nodes along a path
- Destination Options: for a packet's destination node(s)
- A variable number of variable-length options



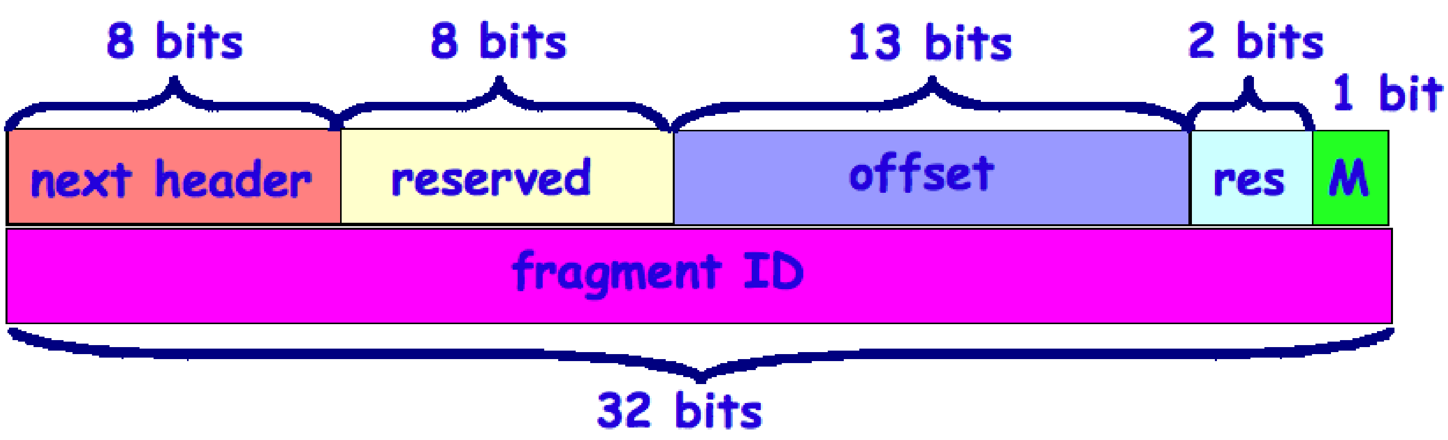
TLV-encoding (Type-Length-Value)

- Type: identifier of type of option
- Two highest bits of Type: unrecognised option processing:
  - 00 – skip over the option and continue
  - 01 – discard the packet
  - 10 – discard the packet and send ICMPv6
  - 11 – discard the packet and send ICMPv6 only if destination isn't IPv6 multicast address
- Third highest-order bit of Type: whether (1) or not (0) Option Data can change en-route to the final destination
- Length: length of the Option Data, in octets



Fragment Header

- **Offset**: the offset, in 8-octet units, of the data following this header, relative to the start of the Fragmentable Part of the packet
- **M flag**: 1 – more fragments, 0 – last fragment



Control Protocol(s)

- IPv4 Control Protocols:
  - ARP (for Ethernet)
  - ICMP
  - IGMP
- IPv6 Control Protocol:

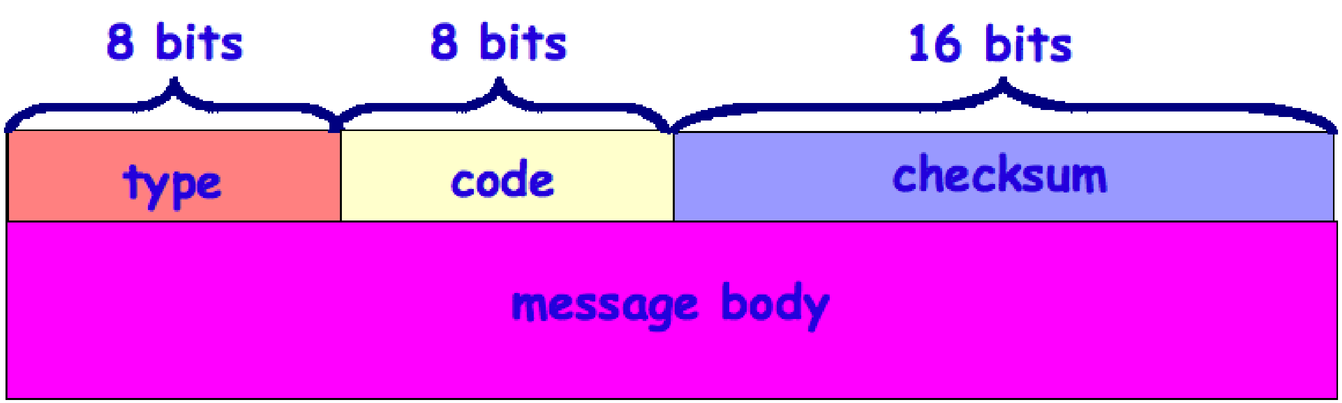
<u>ICMPv6</u>

(IPv6 Next Header value = 58)

Must be fully implemented & supported!

ICMPv6

- Type field:
  - 0 – 127: error messages
  - 128 – 255: informational messages
- Body includes the the start of the invoking packet!
- Must not be fragmented!
- Must not be originated in response to
  - ICMPv6 error or redirect messages
  - multicast/broadcast packets addresses (with some exceptions)



MULTIfunctions of MULTIcast

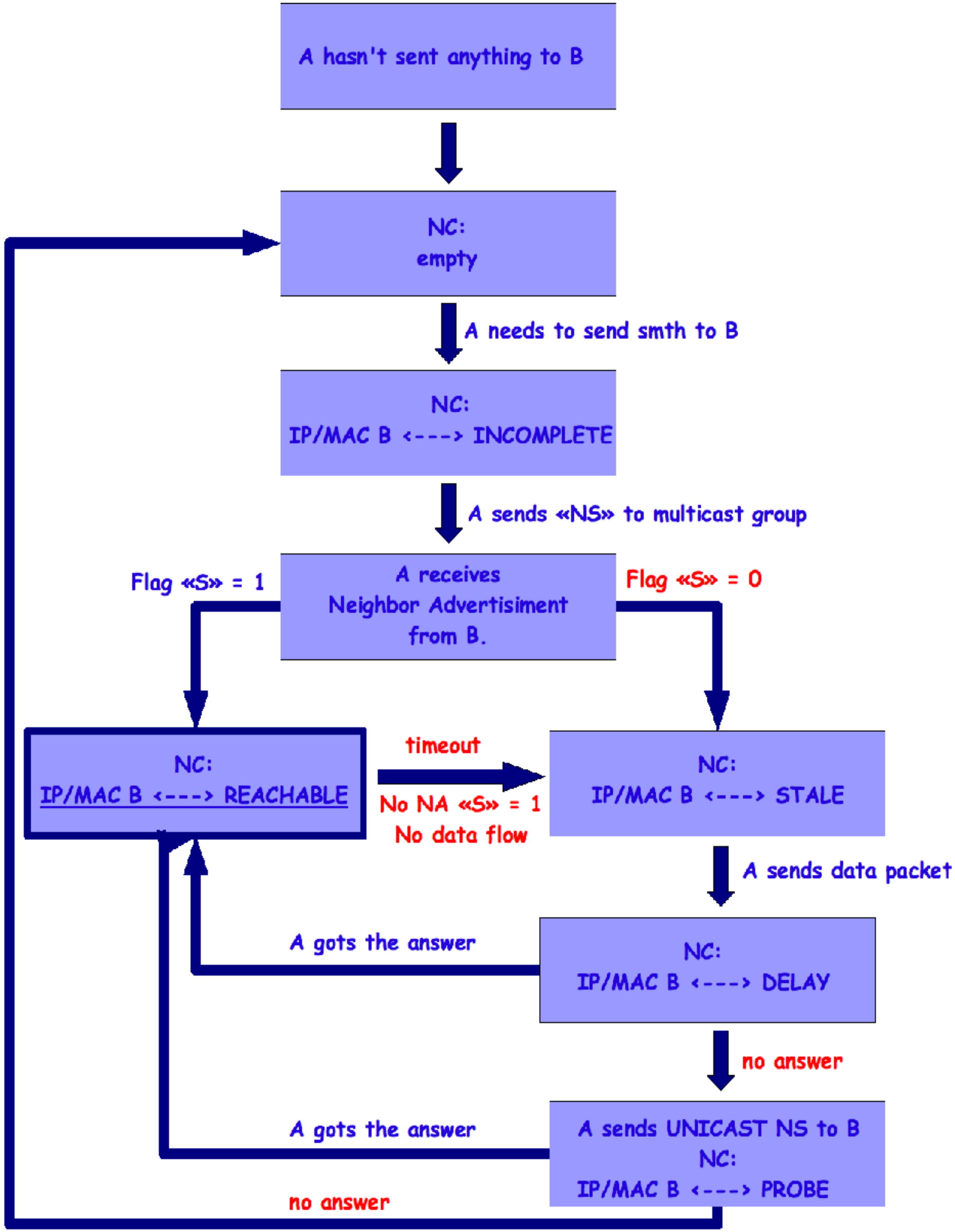
- IPv6 node **MUST** support multicast!
- Broadcast == «all nodes on this link» multicast group
  - *don't forget to enable IGMP snooping/GMRP on switches*
- All nodes with *"similar"* addresses share the same *solicited-node multicast address*
- Solicited-node multicast address format:
  - Globally-assigned prefix **FF02**::1:FF00:0:/104
  - low-order 24 bits of a node address

Example: a node 2001:db8::1:20cd:f345:5432:51d8 joins the multicast group FF02::1:FF00:0:32:51d8

Neighbor Discovery (ND)

- ICMPv6 is used for ND messages
- Multicast is used (unlike ARP)
- To request the link-layer address: *neighbor solicitation (NS) query*
- To provide some info: *neighbor advertisement (NA)*
  - *Soliced flag: S=1 – in response to NS*
  - *S=0 – «unsolicited» NA*
- Information is stored in:
  - *neighbor cache (NC)*
  - *destination cache (DC)*

Information exchange with upper-layer!


A hasn't sent anything to B
NC:
empty
A needs to send smth to B
NC:
IP/MAC B <---> INCOMPLETE
A sends «NS» to multicast group
A receives
Neighbor Advertisiment
from B.
Flag «S» = 1
Flag «S» = 0
NC:
IP/MAC B <---> REACHABLE
timeout
No NA «S» = 1
No data flow
NC:
IP/MAC B <---> STALE
A sends data packet
A gots the answer
NC:
IP/MAC B <---> DELAY
no answer
A gots the answer
A sends UNICAST NS to B
NC:
IP/MAC B <---> PROBE
no answer

ND-proxy

- The target host or a **ND-proxy** could respond to NS query.
- Nodes should give preference to non-proxy NA
- Flag «O» (override)
  - ND-proxy: O=0 (REACHABLE -> STALE)
  - target host: O=1 (Neighbor Cache is updated)

ARP is Dead, Long Live ND!

- Much more than ARP (see Router Discovery and redirects)
- Reducing network load (multicast vs broadcast)
- Improving robustness of packet delivery
  - Neighbor unreachability detection (incl. half-link failures detection)
  - Notification from/to upper-layer!

Anycast

-
- The same "anycast" address is assigned to a group of interfaces (nodes)
- A packet sent to an anycast address is delivered to the "nearest" interface (node) having this address
- Allow to increase the service reliability
- Allocated from the unicast address space

IPv6 Node Configuration

- IPv6 address configuration:
  - Interface ID
    - manual
    - auto (stateful or stateless)
  - Network ID
    - manual
    - auto (stateful or stateless)
    - **pre-defined well-known prefix (link-local, FE80::/10)**
  - additional parameters (routes, e.g.)

Interface Autoconfiguration

- Modified EUI-64 constructed from MAC
  - *see next slides for some alternatives*
- What about collisions?
  - duplicate MAC addresses
  - duplicate interface ID (manual configuration, e.g.)
- Neighbor Discovery locates the owner of given IP address
- **Duplicate Address Detection (DAD)** based on ND

Duplicate Address Detection

1. Node X is going to assign IP address A on its interface “I”
2. Interface “I” joins the multicast groups:
    1. FF02::1 (*“all nodes”*)
    2. FF02::1:FF00:0:A' *(the solicited-node multicast address «all nodes with IP = A»)*
3. Is there any NS queries? *(dst ip = FF02::1:FF00:0:A*, *src ip = ::)*
4. X sends NS query *(dst ip = FF02::1:FF00:0:A*, *src ip = ::)*
5. Is there any NA *(flag S = 0)* sent to address FF02::1?
6. In case of events 3 or 5 - **the address isn't unique!**
7. Else – **the address is unique**

Must be performed on all unicast addresses (except for anycast)

StateLess Address Auto Configuration (SLAAC)

- Link-local address is already here:
  - *well-know network ID*
  - *modified EUI-64 as interface ID*
  - *DAD to ensure uniqueness*
- **Ready to communicate with neighbors!**
- What's next?
  - other IPv6 network IDs (global, e.g.)
  - default gateway(s)
  - routing table
- **Routers have this info already!**

Your Router Is Your Neighbor!

- Neighbor Discovery (RFC4861)
- Routers join the “all routers” multicast group FF02::2
- Cliens send a «Router Solicitation» query (RS)
- Routers send out «Router Advertisiment» messages (RA)
  - periodically
  - in response to the RS query

Router Advertisement

<table>
<tr><td>bits</td><td>8</td><td>10</td><td></td><td>16</td><td>32</td></tr>
<tr><td>type = 134</td><td colspan="3">code = 0</td><td colspan="2">checksum</td></tr>
<tr><td>hop limit</td><td>M</td><td>O</td><td>reserved</td><td colspan="2">router lifetime</td></tr>
<tr><td colspan="6">reachable time</td></tr>
<tr><td colspan="6">retransmit timer</td></tr>
<tr><td colspan="6">options (variable length)</td></tr>
</table>

Src IP = link-local, Dst IP = the source IP of the RS query or FF02::1

- M,O flags: indicate that addresses (M) or other configuration info (O) is available via DHCPv6
- Router lifetime (in seconds) – the lifetime associated with the default router *(0 - the router isn't default router, shouldn't appear on the default router list)*
- Reachable time (millisecs) – how long the neighbor is reachable after receiving a reachability confirmation *(NC record goes from Reachable -> Stale then)*
- Retransmit timer (millisecs) – the interval between retransmitted NS messages

RA: possible options

Additional configuration info:

- Prefixes
  - prefix ID and length
  - Lifetime
  - usage: for stateless configuration or destination cache
- MTU
- Link-layer address of the interface from which RA is sent

NB: Unmatched advertised parameters could lead to unstable network!

How to secure ND

- ND takes place on-link (between adjacent nodes)
- ND messages are not to be routed
- Routers decrement TTL (Hop Count)
- TTL < 255 may mean *'the packet was routed'* *(NB: «0 – 1=255»!!)*
- **Generalized TTL Security Mechanism (GTSM) (RFC5082)**

How to secure ND (cont.)

- One of major threats: address spoofing attacks
- How to authenticate NA?
- Cryptography is our friend!
- Symmetric: key protection is an issue
- Asymmetric:
  - key distribution is an issue.
  - how to authenticate the peer?

Give me the place to stand, and I shall move the earth

- Neighbor IP address is already known!
- IP address can be used to authenticate the peer
- IP and public key are associated
- Public key is attached to ND message
- Public key is verified against IP address
- **Cryptographically Generated Addresses (CGA, RFC3972)**

Cryptographically Generated Address

1. A private/public key pair is generated for a node
2. Interface ID is calculated as an public key fingerprint
3. Subnet prefix and interface ID are concatenated
4. Duplicate Address Detection is performed (CGA is re-calculated if necessary – up to 3 times)
5. CGA parameter is formed:
   - IPv6 address
   - Public key
   - Some additional parameters
6. DNS and other records are updated..

The random modifier allows to change the fingerprint (IP address) periodically

CGA Verification

1. The verifier know the sender IP address (CGA)
2. The verifier gets the sender public key from CGA parameter
3. The verifier checks the association between IPv6 CGA and the corresponding pubic key
4. After then, the digital signature of ND message is verified

No PKI, CA or trusted servers is needed!

SEcure Neihgbor Discovery (SEND, RFC3971) describes Neighbor Discovery threats and protection

SEND: SLAAC protection

- Router Advertisement IP address can be spoofed
- RA IP is unknown => GCA can not be used
- Routers **ARE** authorised to act as routers
- Routers **MAY** be authorised to advertise prefixes
- Routers are given certificates from a trust anchor
- The hosts are configured with trust anchor(s)

Big Brother is watching you!

- MAC addresses are globally unique (in most cases)
- SLAAC: Interface ID is derived from MAC
- Users are mobile *(home – office – internet-cafe – business trips – travels – office - home..)*:
  - network prefixes are changing
  - interface ID remains constant over time!
- User can be identified and tracked!

Privacy Extensions for SLAAC

- Task: provide privacy for users
- Requirements: do not broke SLAAC
- Approach: change the interface ID over time
- Interface ID must be **locally** (on-link) unique
- Interface ID can be random
- Duplicate Address Detection ensures uniqueness
- In case of collision a new random address is generated

Default Address Selection

- There are a number of ways to assign IPv6 addresses
- Requirements may be conflicting:
- Corporate environment: easily identification of a node
- Internet-connectivity: privacy is an issue
- IPv6 nodes are multi-addressed usually (+link-local)
- What address to choose for communication?

See RFC5220 «Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules»

Fragmentation

"Fragmentation considered harmful"

- Inefficient use of resources of hosts, routers and bandwidth
- Degraded performance due to loss of fragments
- Reassembly is difficult

Why fragmentation?

- MTU mismatch along the packet path (!tunnels!)
- TCP/IP implementations

Blocking PMTUD leads to packets disappearing into "black hole"

IPv6 Fragmentation

By the source host only, not by routers along the packet's path!

- No “Don't Fragment” bit anymore
- Minimum MTU = 1280 bytes
- If a packet size > MTU, the packet is dropped, ICMPv6 is sent

How to choose a packet's size:

- Always fragment to 1280 bytes (1232 bytes of payload)
- Use PMTUD, store MTU value in Destination Cache (DC)
- Applications can access IPv6 layer using API (Berkley sockets, e.g: see RFC3542)

Socket Option	Description
IPV6_USE_MIN_MTU	Disable PMTUD, use minimum MTU = 1280 bytes
IPV6_PATHMTU	Retrieve the current MTU value for the socket
IPV6_RECVPATHMTU	Enable the receipt of the current MTU from recvfrom()
IPV6_DONTFRAG	Disable the inserting of a fragment header

IPv6 & DNS

New Resource Record introduced: AAAA

furry:~ furry$ dig www.kame.net aaaa

www.kame.net.   IN   AAAA   2001:200::8002:203:47ff:fea5:3085

Reverse Delegation:

- the pseudo-domain ipv6.arpa
- Each label is a *nibble* (4 bits, one hex number)

Example:

PTR RR for an IPv6 address **2001:db8::20:219f:bd8c:17af**

f.a.7.1.c.8.d.b.f.9.2.1.0.2.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ipv6.arpa.   PTR

Don't forget to use $ORIGIN to simplify your DNS zone file!

Migration

- Dual-stack nodes (IPv6+IPv4)
  - most workstations are IPv6-enabled
  - Windows: prefers IPv6 in some cases
  - uncontrolled connectivity is a security issue!
- Tunnels: connection of IPv6 domains via IPv4 clouds
- Address translations: interconnection between IPv6 and IPv4 domains

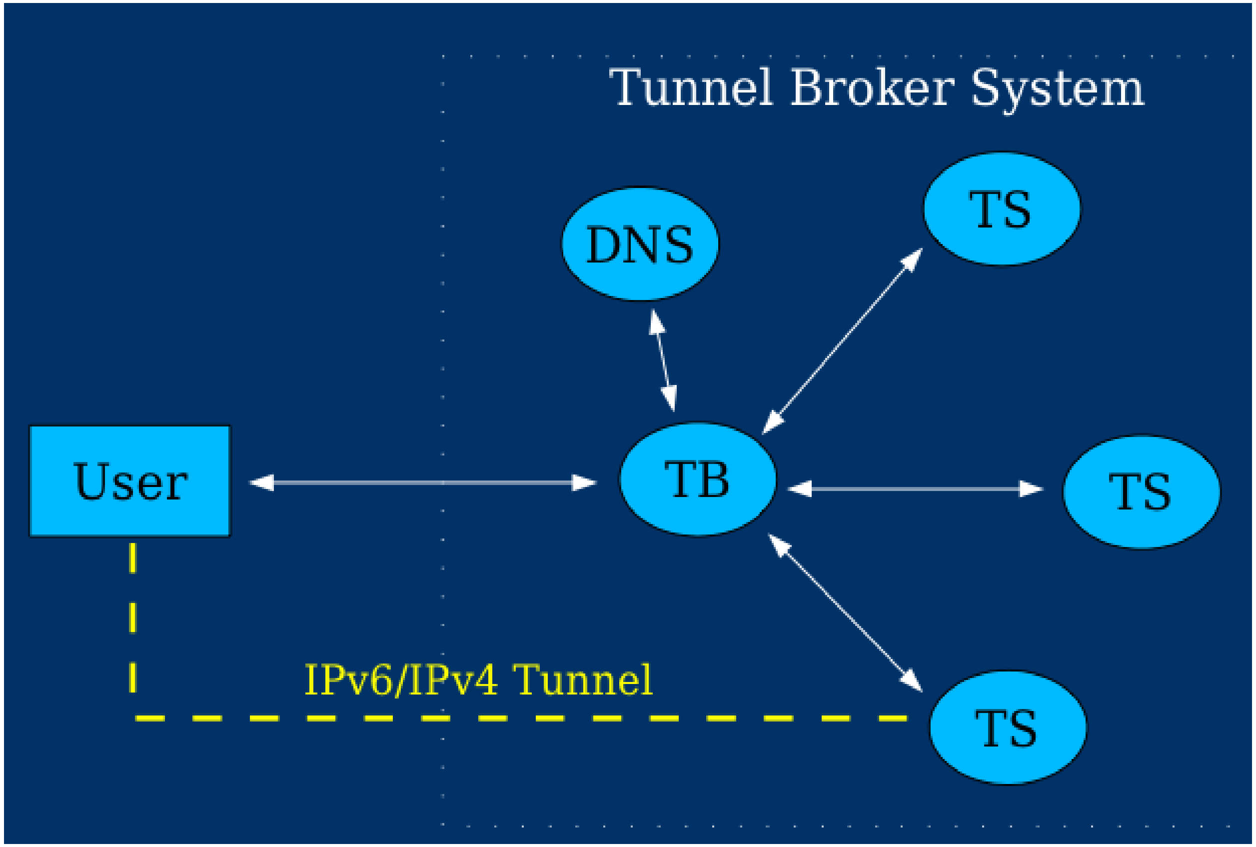
Tunnelling

- 6to4 – the most common IPv6 over IPv4 tunnelling protocol. Tunnel endpoints must have public IPv4 addresses
- Teredo – encapsulating IPv6 inside IPv4/UDP
  - NAT-T is supported
  - Globally unique IPv6 address is assigned to each endpoint
  - Windows Vista: enabled, but not active by default (**teredo.ipv6.microsoft.com)**

Can be a security issue!!

Tunnel brokers

- A service to provide encapsulated connectivity
- See RFC3053 "IPv6 Tunnel Broker" for details
- Extensive list can be found at: *http://en.wikipedia.org/wiki/List_of_IPv6_tunnel_brokers*

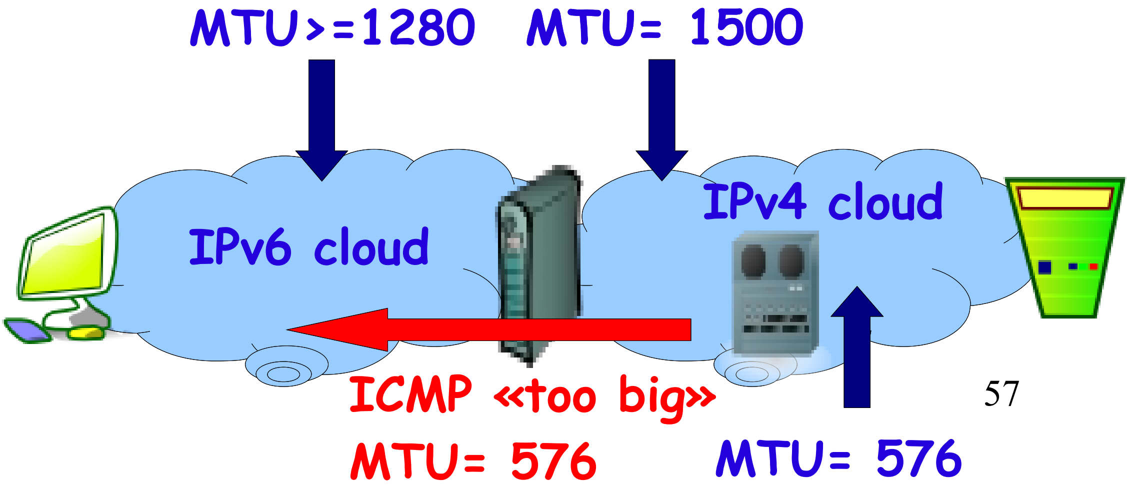


Address Translation: NAT64

- http://tools.ietf.org/html/draft-bagnulo-behave-nat64-02
- Packet headers are translated according to Stateless IP/ICMP Translation Algorithm (SIIT)
- IPv6 {address + port} is mapped into IPv4 {address + port}
- IPv4 addresses are mapped into IPv6 addresses as Pref64::IPv4 (Pref64 is an /96 IPv6 address pool)

Fragmentation & NAT64

- IPv4 minimum MTU: 68 bytes
- IPv6 minimum MTU: 1280 bytes
- IPv4-node may originate ICMP “too big” with MTU < 1280
- What IPv6-node can do?
  - include a Fragment header or
  - reduce the size of subsequent packets



IPv6 Advantages

- More efficient address space allocation
- End-to-end addressing; no NAT anymore!
- Fragmentation only by the source host
- Routers do not calculate header checksum (speedup!)
- Multicasting instead of broadcasting
- Built-in security mechanisms
- Single control protocol (ICMPv6)
- Auto-configuration
- Modular headers structure

Myths and Legends
«How can I remember...»

- Use the Force (of DNS), Luke!
- Manual configuration: easy-readable addresses
- Use a compact notation (a lot of network prefixes to choose from)

Just compare:

furry:~ furry$ dig www.ipv6porn.co.nz aaaa

www.ipv6porn.co.nz. 3324 IN AAAA 2002:3cea:4c32::1 (17 chars)

www.ipv6porn.co.nz. 3324 IN AAAA 2001:388:f000::285 (18 chars)

furry:~ furry$ dig www.ipv6porn.co.nz a

www.ipv6porn.co.nz. 10000 IN A 60.234.76.50 (12 chars)

Myths and Legends
«I don't want it, I don't need it...»

- IPv6 is already here!
- Spontaneous self-organised and uncontrolled IPv6 networks are security issues
- Better be pro-active rather than reactive
- IPv6 is becoming more popular: get ready to meet it!