IPv6 Addressing

Need for IPv6

- IPv4 is running out of addresses. IPv6 is the successor to IPv4. IPv6 has a much larger 128-bit address space.
- The development of IPv6 also included fixes for IPv4 limitations and other enhancements.
- With an increasing internet population, a limited IPv4 address space, issues with NAT and the IoT, the time has come to begin the transition to IPv6.

IPv6 Address Representation

IPv6 Addressing Formats

- IPv6 addresses are 128 bits in length and written in hexadecimal.
- IPv6 addresses are not case-sensitive and can be written in either lowercase or uppercase.
- The preferred format for writing an IPv6 address is x:x:x:x:x:x:x:x, with each “x” consisting of four hexadecimal values.
- Examples of IPv6 addresses in the preferred format:

  2001:0db8:0000:1111:0000:0000:0000:0200

  2001:0db8:0000:00a3:abcd:0000:0000:1234

Rule 1 – Omit Leading Zero

The first rule to help reduce the notation of IPv6 addresses is to omit any leading 0s (zeros).

Examples:

- 01ab can be represented as 1ab
- 09f0 can be represented as 9f0
- 0a00 can be represented as a00
- 00ab can be represented as ab

Note: This rule only applies to leading 0s, NOT to trailing 0s, otherwise the address would be ambiguous.

Type	Format
Preferred	2001 : **0**db8 : **000**0 : 1111 : **000**0 : **000**0 : **000**0 : **0**200
No leading zeros	2001 : db8 : 0 : 1111 : 0 : 0 : 0 : 200

Rule 2 – Double Colon

A double colon (::) can replace any single, contiguous string of one or more 16-bit hextets consisting of all zeros.

Example:

- 2001:db8:cafe:1:0:0:0:1 (leading 0s omitted) could be represented as 2001:db8:cafe:1::1

Note: The double colon (::) can only be used once within an address, otherwise there would be more than one possible resulting address.

Type	Format
Preferred	2001 : **0**db8 : **000**0 : 1111 : **0000** : **0000** : **0000** : **0**200
Compressed	2001:db8:0:1111::200

IPv6 Address Types

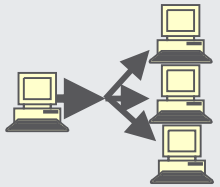
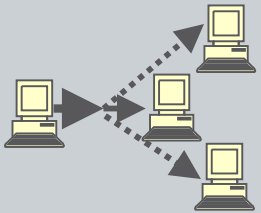
Unicast, Multicast, Anycast

There are three broad categories of IPv6 addresses:

- **Unicast** – Unicast uniquely identifies an interface on an IPv6-enabled device.
- **Multicast** – Multicast is used to send a single IPv6 packet to multiple destinations.
- **Anycast** – This is any IPv6 unicast address that can be assigned to multiple devices. A packet sent to an anycast address is routed to the nearest device having that address.

Note: Unlike IPv4, IPv6 does not have a broadcast address. However, there is an IPv6 all-nodes multicast address that essentially gives the same result.

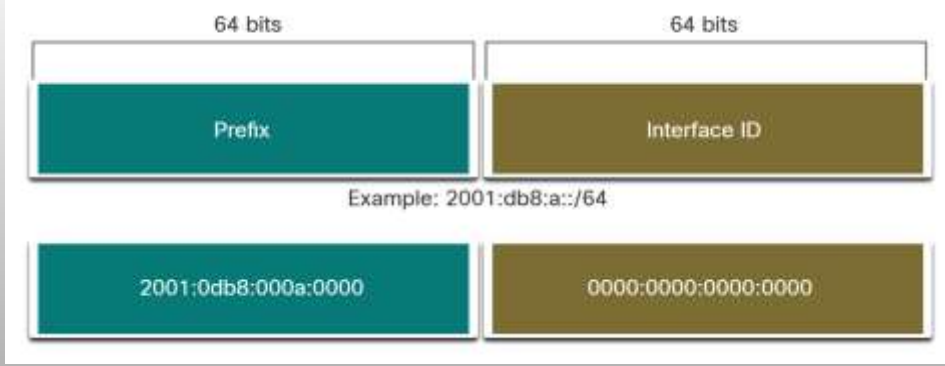
IPv6 Address Types

Address Type	Description	Topology
Unicast	"***One to One***" • An address destined for a single interface. • A packet sent to a unicast address is delivered to the interface identified by that address.	
Multicast	"***One to Many***" • An address for a set of interfaces (typically belonging to different nodes). • A packet sent to a multicast address will be delivered to all interfaces identified by that address.	
Anycast	"***One to Nearest***" (Allocated from Unicast) • An address for a set of interfaces. • In most cases these interfaces belong to different nodes. • A packet sent to an anycast address is delivered to the closest interface as determined by the IGP.	

IPv6 Prefix Length

Prefix length is represented in slash notation and is used to indicate the network portion of an IPv6 address.

The IPv6 prefix length can range from 0 to 128. The recommended IPv6 prefix length for LANs and most other types of networks is /64.



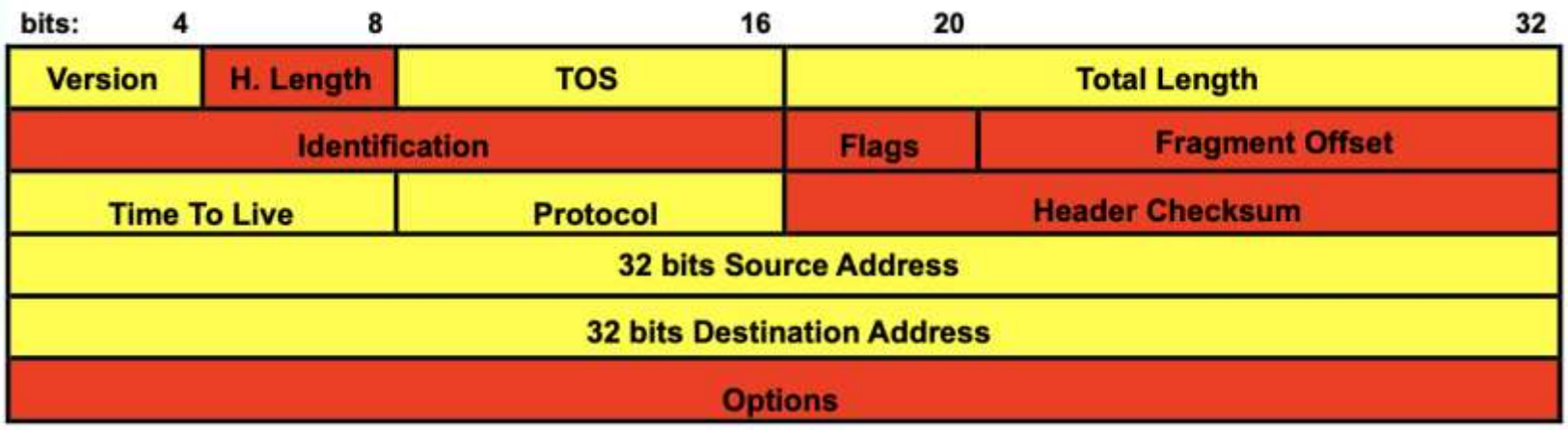
IPv6 Feature Summary

- Increased size of address space
- Header simplification
- Autoconfiguration
  - Stateless (RFC 2462) or stateful (DHCPv6)
  - Facilitates renumbering
- QoS
  - Integrated services (int-serv), Differentiated services (diff-serv and RFC2998)
  - RFC 3697
- IPSec
  - As for IPv4
- Transition techniques
  - Dual stack
  - Tunnelling

IPv6 header

- IPv6 header is simpler than IPv4
  - IPv4: 14 fields, variable length (20 bytes +)
  - IPv6: 8 fields, fixed length (40 bytes)
- Header fields eliminated in IPv6
  - Header Length
  - Identification
  - Flag
  - Fragmentation Offset
  - Checksum
- Header fields enhanced in IPv6
  - Traffic Class
  - Flow Label

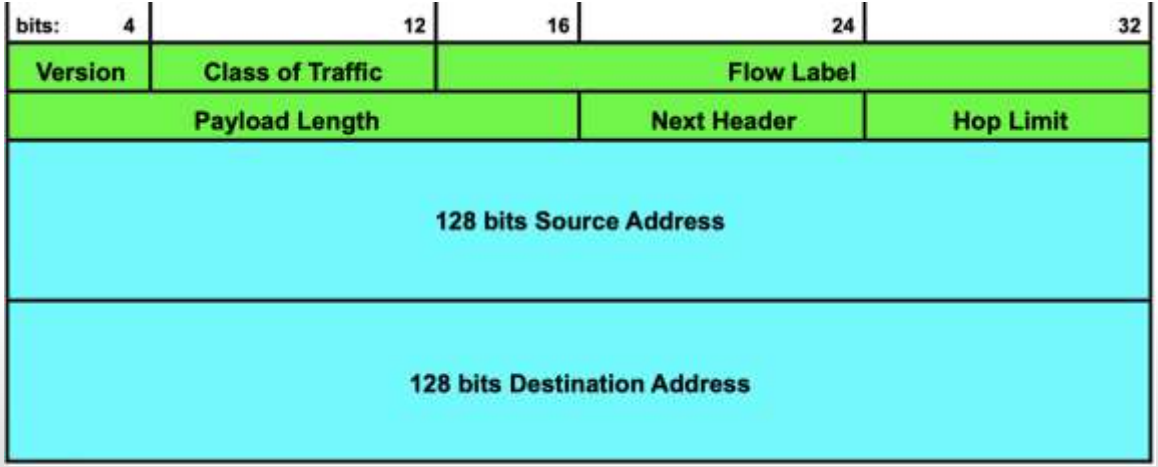
IPv6 vs IPv4 header



Modified Field

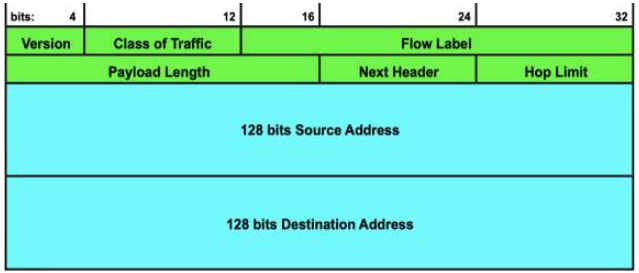
Deleted Field

IPv6 vs IPv4 header



- The **Traffic Class field** is an **8-bit field** in the IPv6 header.
- Similar to the **Type of Service (ToS)** field in IPv4.
- Allows to **classify and prioritize packets** (e.g., voice, video, file transfer).

IPv6 Traffic Class



How it works?

The **Traffic Class** field is divided into two parts (defined by Differentiated Services – **DiffServ**):

- **Differentiated Services Code Point (DSCP, 6 bits):** Defines packet priority/quality of service (QoS).
- **Explicit Congestion Notification (ECN, 2 bits):** Used for congestion control.

So, the **Traffic Class** = **DSCP (6 bits) + ECN (2 bits).**

IPv6 Traffic Class

Using Class of Traffic (examples)

Network devices (routers, switches, firewalls) **read this field** to decide how to handle packets:

- High priority (e.g., VoIP, video calls) → Low delay, low loss.
- Normal priority (e.g., web browsing).
- Low priority (e.g., file downloads, background updates).

Example values:

- 101110xx → Expedited Forwarding (for voice).
- 001010xx → Assured Forwarding (for streaming).
- 000000xx → Best Effort (default traffic).

IPv6 Flow Label

Resource Reservation

IPv6 uses the **Flow Label** field together with the **Traffic Class** field to identify these flows.

Flow

A flow is defined as a sequence of packets from a source to a destination with the same properties (e.g., same DSCP, same port).

Protocols Used

RSVP (Resource Reservation Protocol): Used to signal and reserve resources along the path.

IPv6 Flow Label

Resource Reservation

IPv6 uses the **Flow Label** field together with the **Traffic Class** field to identify these flows.

IPv6 Field	Purpose in IntServ
Traffic Class	Marks the type/priority of the flow
Flow Label	Identifies a specific flow across the network

IPv6 Fields used by IntServ

Together, these allow routers to recognize flows and provide per-flow QoS guarantees.