Internet of Things
IO 4041
IP and IPv6

IPv4

Class A
Subnet Mask

Netwok	Host	Host	Host
255	0	0	0

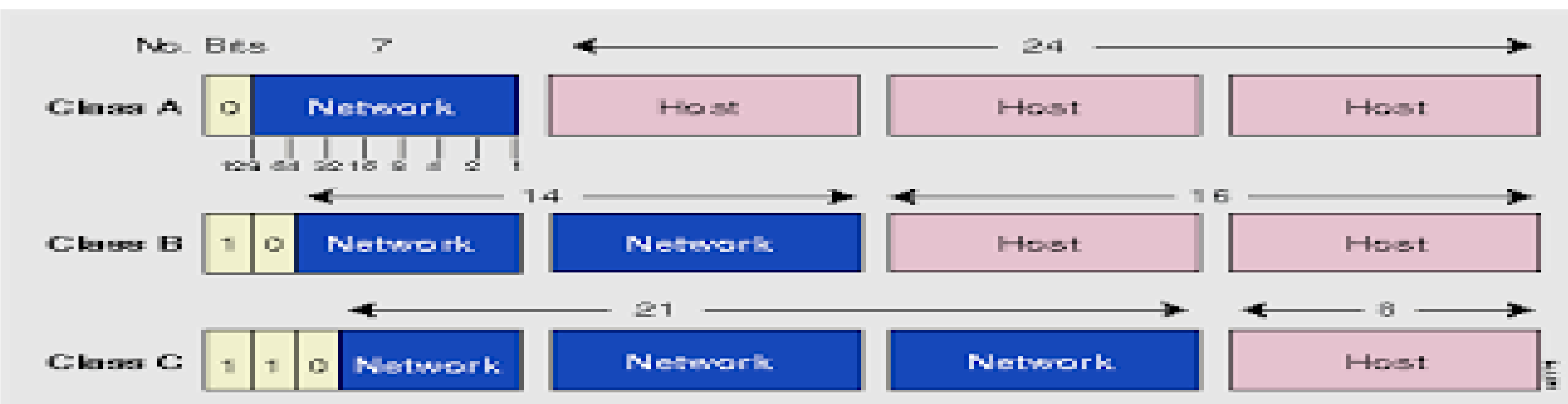
Class B
Subnet Mask

Netwok	Network	Host	Host
255	255	0	0

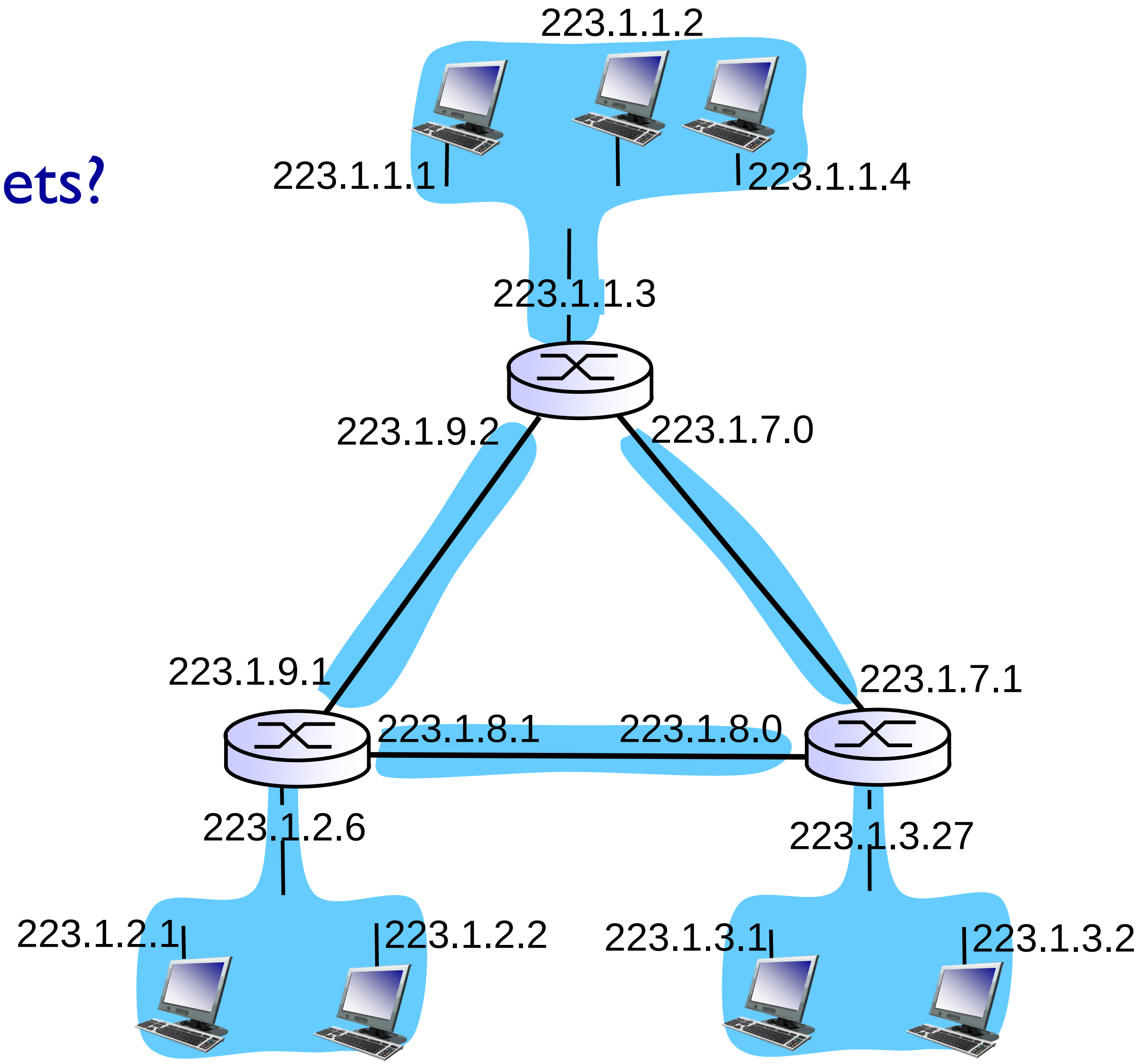
Class C
Subnet Mask

Netwok	Network	Network	Host
255	255	255	0

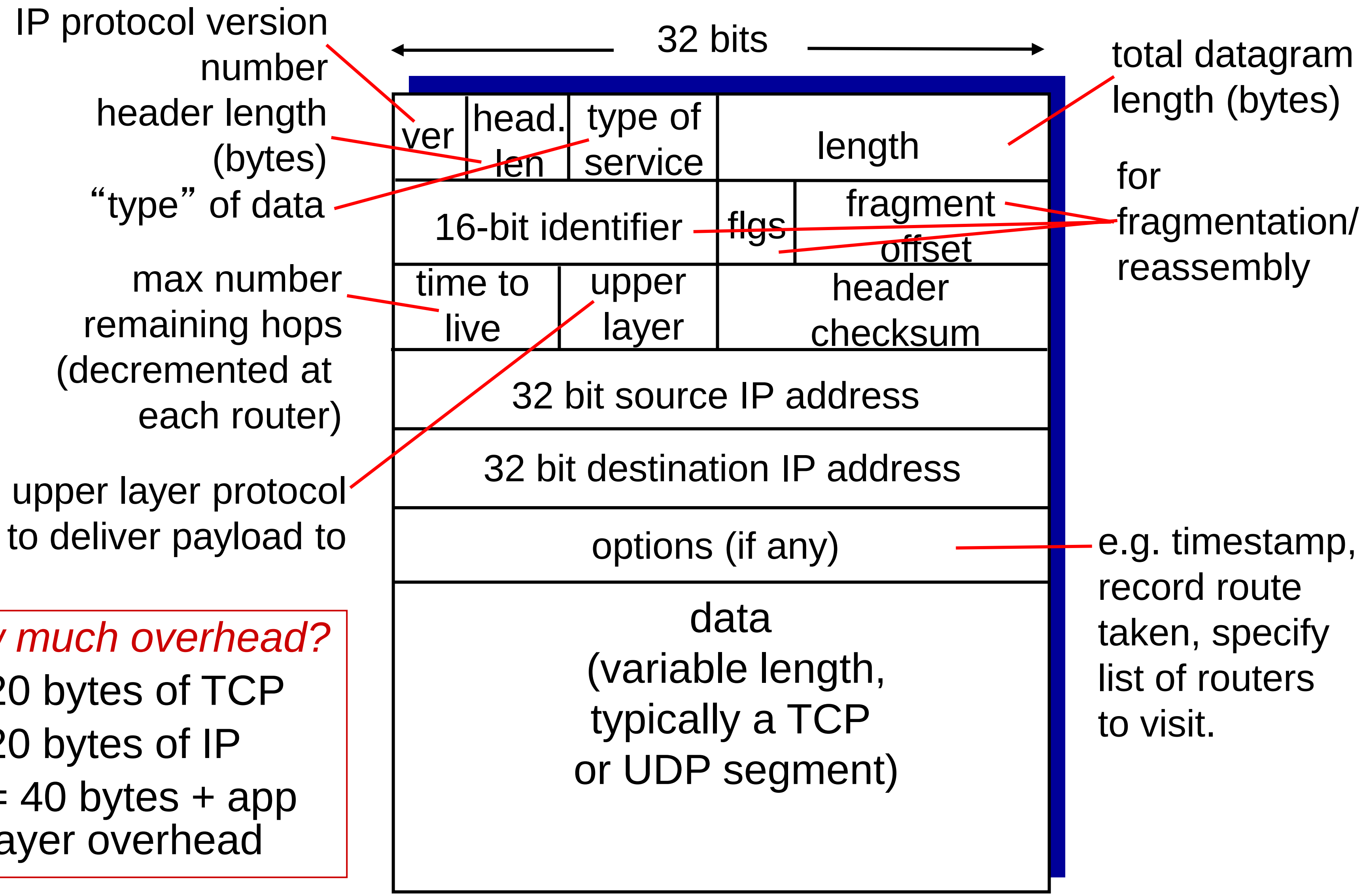
www.smartPCtricks.com



how many subnets?



IP datagram format

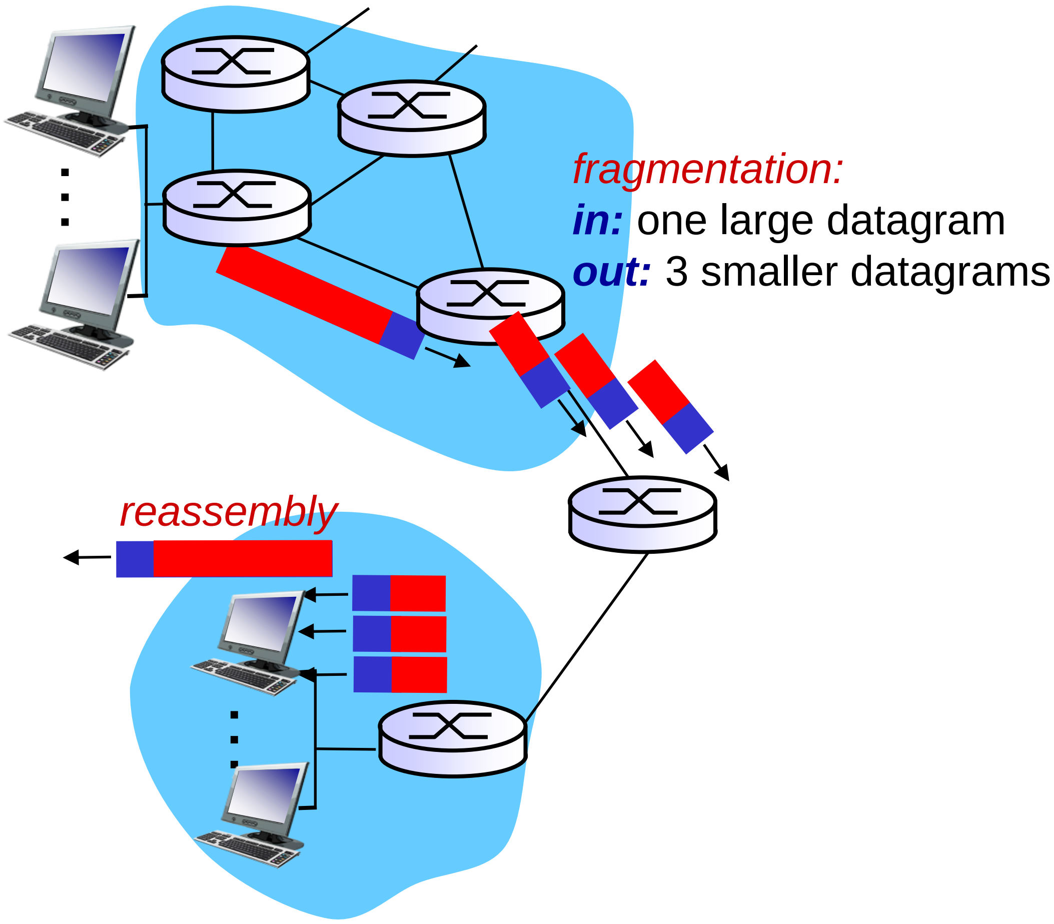


how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

IP fragmentation, reassembly

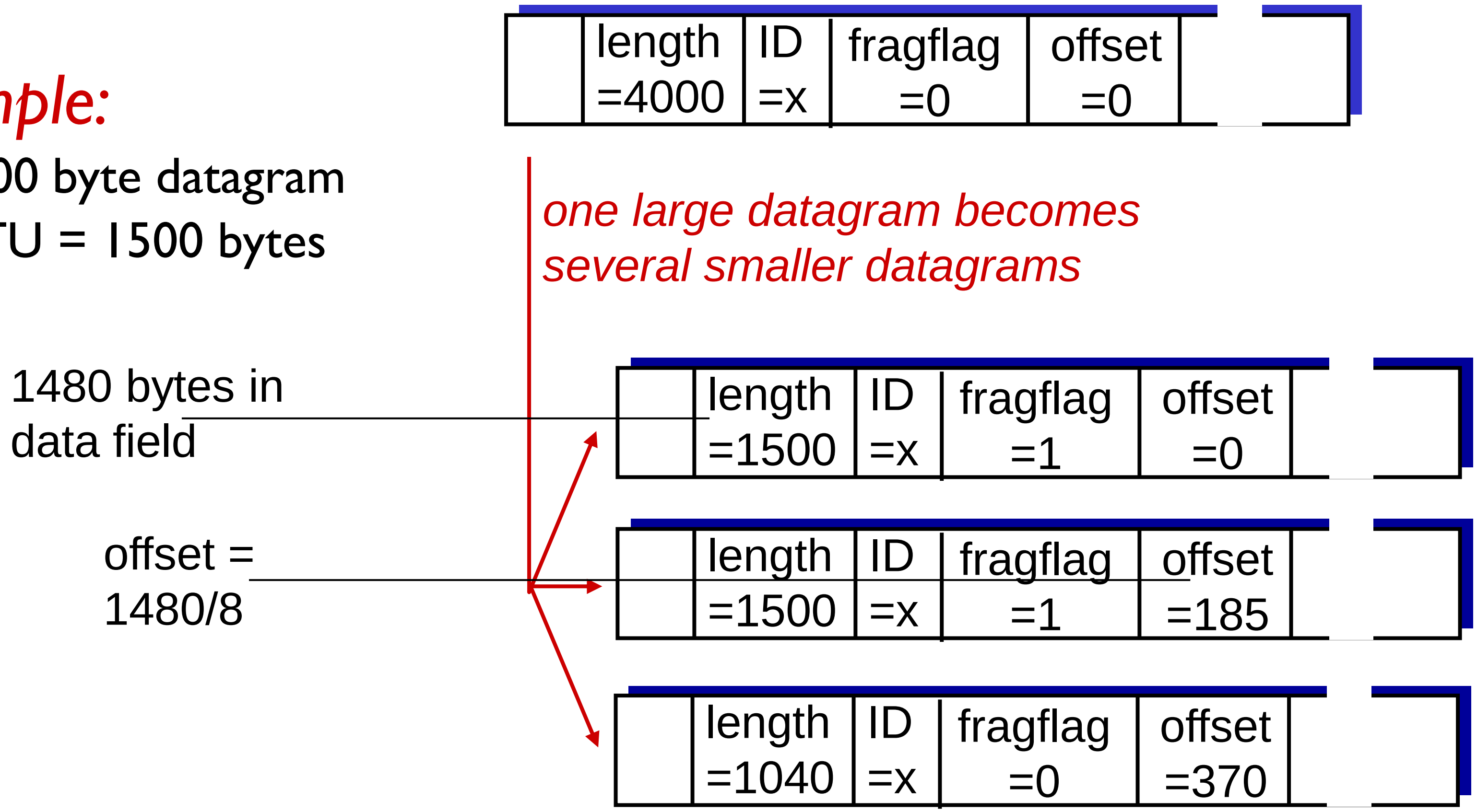
- network links have MTU (max.transfer size) - largest possible link-level frame
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments



IP fragmentation, reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes



IPv6

- an evolution of IPv4
- builds on IPv4 with no change in the
  - fundamental and architectural principles of the IP protocol suite
- addresses are 128 bits long
- 16 bytes of IPv6 address are represented as a group of hexadecimal digits, separated by colons, e.g.:

2000:fdb8:0000:0000:0001:00ab:853c:39a1

- Shorthand – leave out groups of zeros and leading zeros:

2000:fdb8:::1:ab:853c:39a1

- A few new features have been added but IPv6 fundamentally preserves the architectural principles of IP

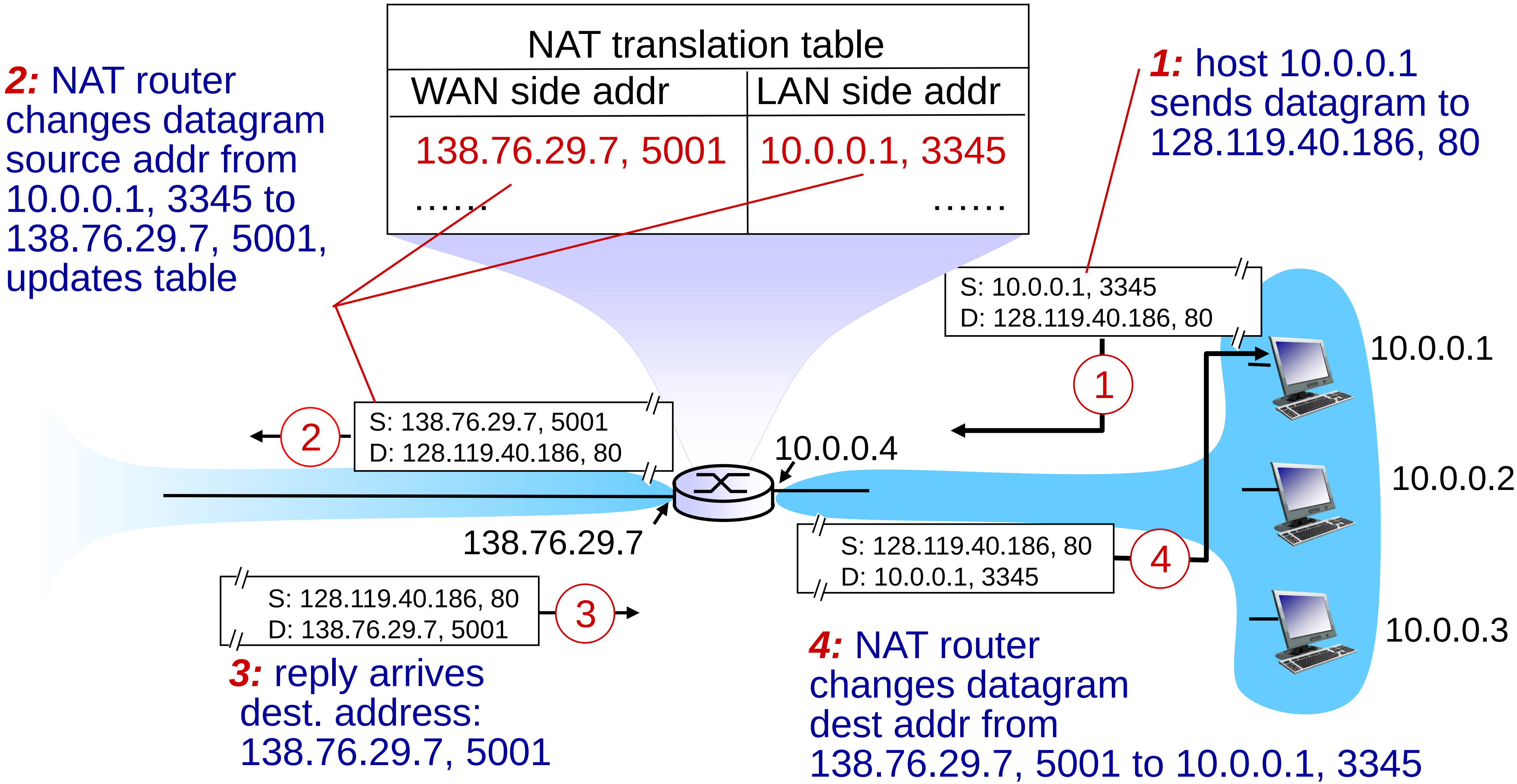
IPv6

- existing transport protocols UDP and TCP have not been modified

- **Cost of migration** has slowed down the adoption rate of IPv6

- IETF efforts to slow down the pace of IP address allocation while waiting for IPv6
  - Classless InterDomain Routing (CIDR)
  - Network Address Translation (NAT)

NAT: network address translation



NAT: network address translation

- NAT is **controversial (many in IETF object NAT)**:
  I. Port numbers meant for addressing processes, not for addressing hosts
     - Can cause problem for server processes as they wait for incoming requests at well- known port numbers)

  II. routers are supposed to process packets only up to layer 3

NAT: network address translation

- NAT is controversial:
  III. NAT violates end-to-end argument
    - Hosts should be talking directly with each other without any interference and change in IP and port numbers by interfering nodes

  IV. address shortage should instead be solved by IPv6 rather than patching up with NAT

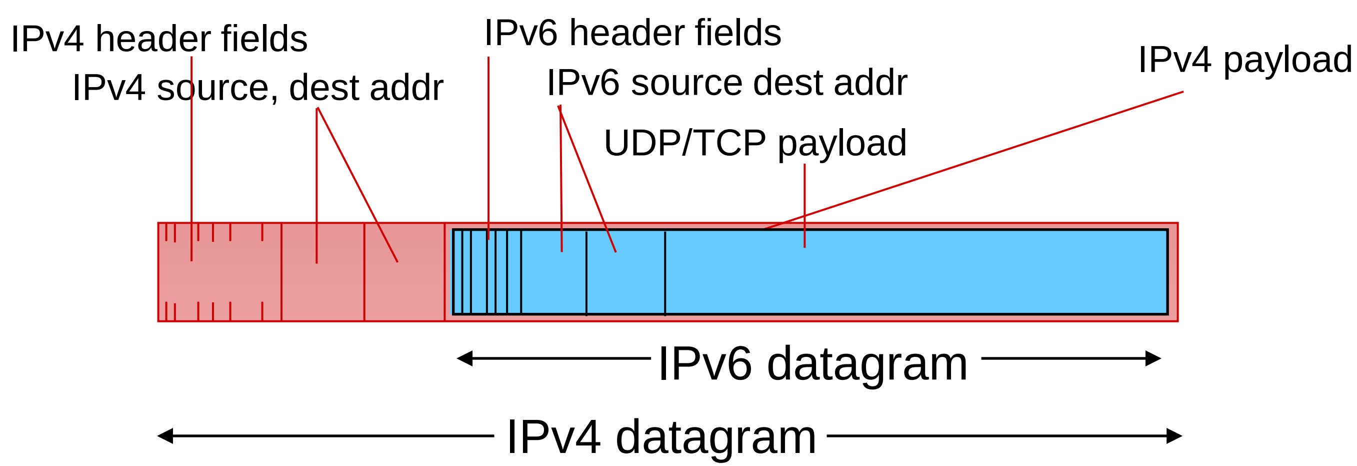
  V. NAT interferes P2P applications
    - P2P file sharing, VOIP

Changes from IPv4

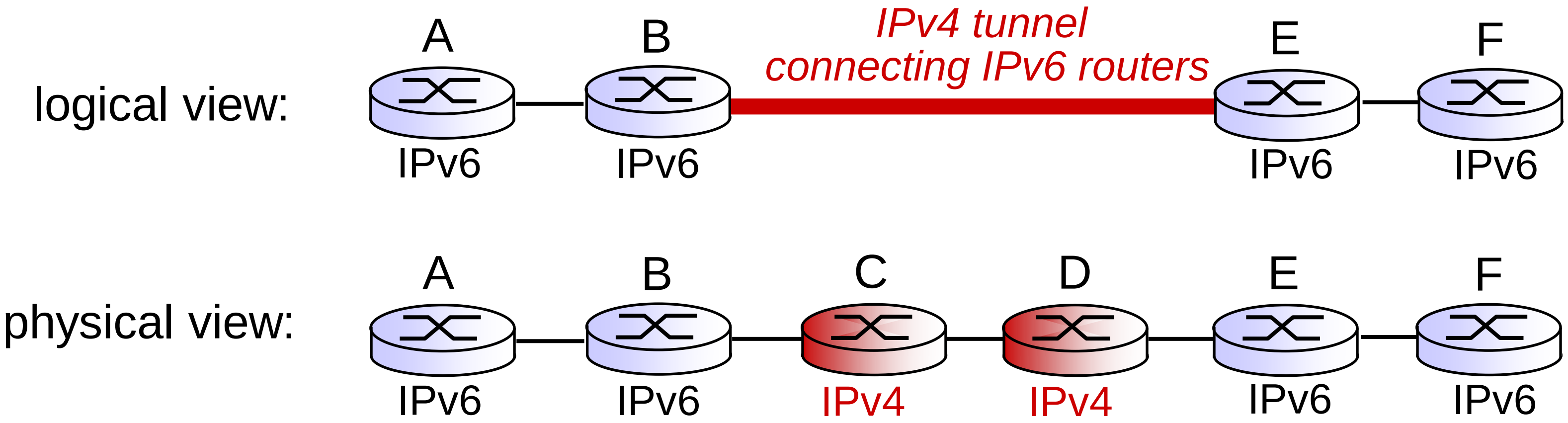
- fixed-length 40 byte header
- *checksum:* removed entirely to reduce processing time at each hop
- Fragmentation/Reassembly: not allowed at intermediate routers
- *options:* allowed, but outside of header, indicated by "Next Header" field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions
- does not impose any specific boundary for the network part similarly to CIDR in IPv4

Transition from IPv4 to IPv6

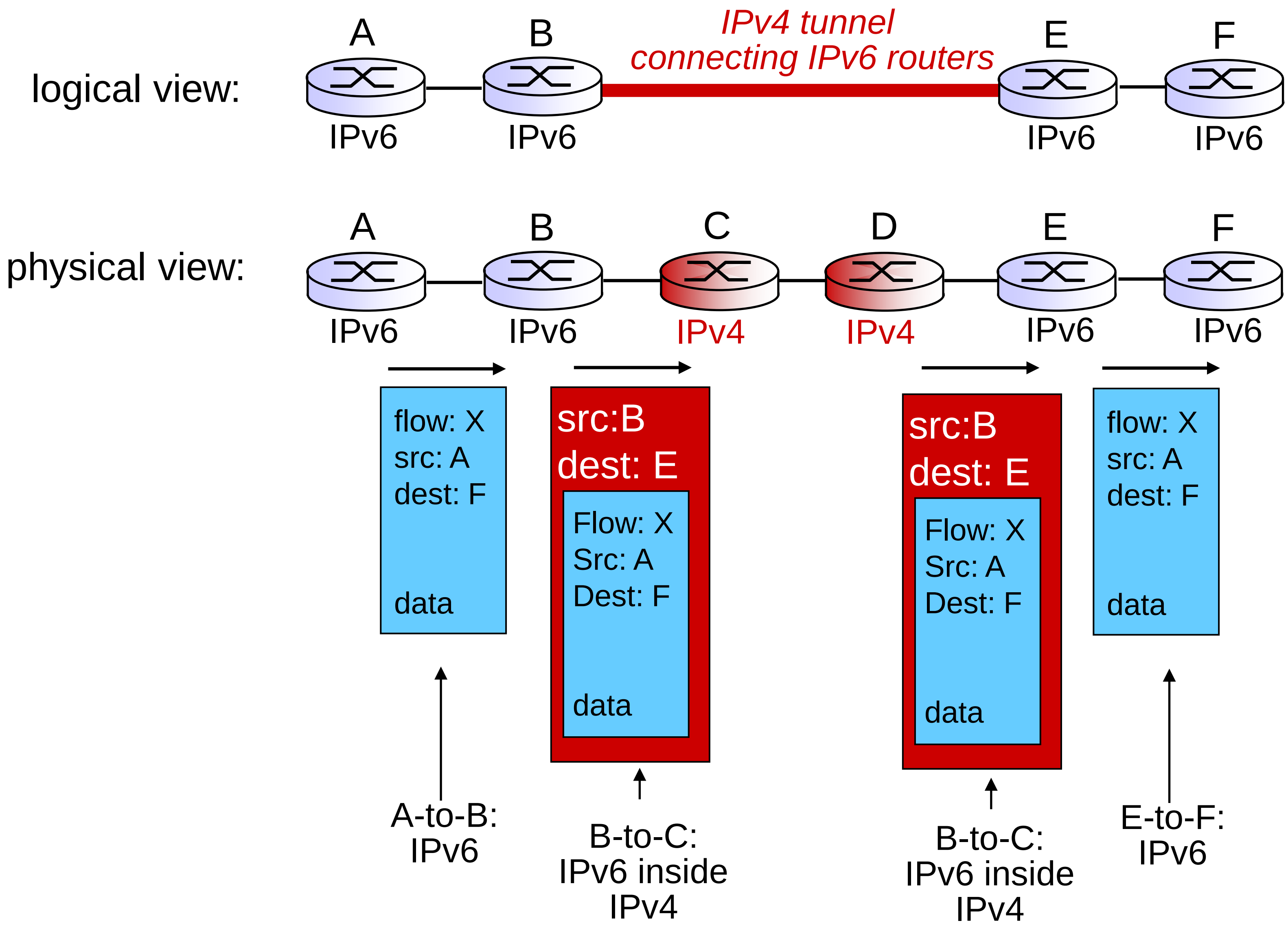
- not all routers can be upgraded simultaneously
  - no "flag days"
  - how will network operate with mixed IPv4 and IPv6 routers?
- *tunneling:* IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers



Tunneling

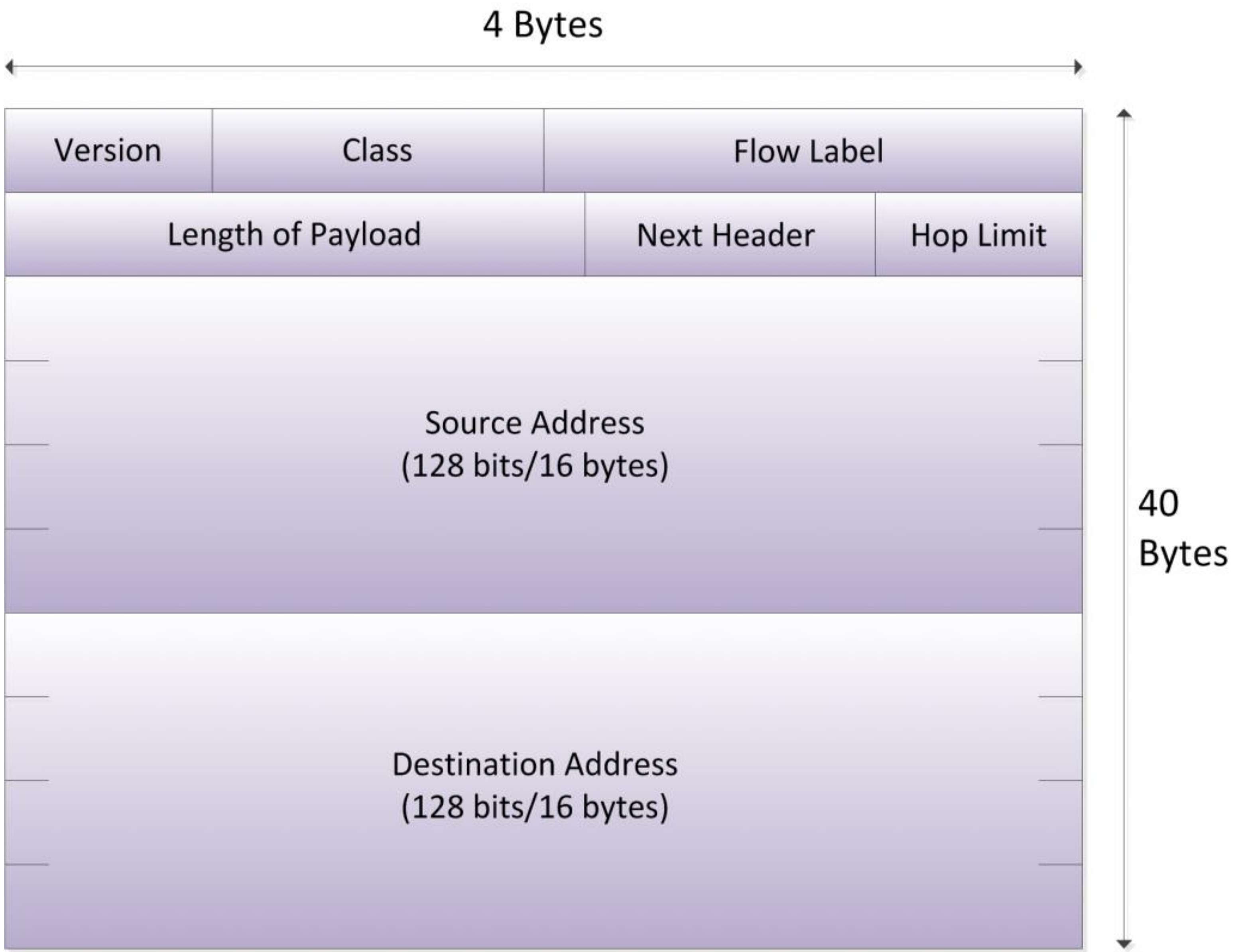

logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

Tunneling



logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

IPv6 header


4 Bytes
Version
Class
Flow Label
Length of Payload
Next Header
Hop Limit
Source Address
(128 bits/16 bytes)
Destination Address
(128 bits/16 bytes)
40
Bytes

IPv6 Packet header

- Version (4 bits): IP version number = 6
- Traffic class (8 bits): 8-bit field used to indicate the packet’s Class or priority
- Flow label (20 bits): indicates that this packet belongs to a specific sequence of packets between a source and destination.
  - requiring special handling by intermediate IPv6 routers
- Payload length (16 bits): this field indicates the length of payload excluding packet header
  - It includes extension headers and upper layer PDU

IPv6 Packet header

- Next header (8bits): This field identifies the header that follows the IPv6 packet header.
  - Indicates the type of the first extension header (if any) or
  - Protocol in the upper layer DPU
- Hop limit (8 bits): Number of intermediate nodes. Indicates the maximum number of links over which the IPv6 packet can travel before being discarded
  - When it equals 0 at a router, the router sends an ICMPv6 Time Exceeded-Hop Limit Exceeded in Transit message to the source and discards the packet.
- Source address (128 bits)
- Destination address (128 bits)

Key Functionalities of IPv6

Larger address space required for large-scale networks:

- Urban networks, Smart Grids, and industrial automation networks are examples where IP smart object networks will potentially comprise hundreds of thousands of nodes.
- Extending the address space from 32 to 128 bits,
  - a larger number of addressable nodes,
  - many more levels of addressing hierarchy (key for routing table efficiency) and
  - autoconfiguration features

Auto-configuration:

- In large scale networks, management at large (provisioning, configuration, management of faults, inventory, performance analysis) quickly becomes very challenging.
- IPv6 has native auto-configuration features

Key Functionalities of IPv6

Header Change:

- Unused IPv4 header fields have been removed (e.g., fragmentation, checksum) and
- a simpler structure with a fixed header has been adopted.
- Flow label field is added

Authentication and privacy:

- Authentication, data integrity, and (potentially) confidentiality
- Security: IPSec (optional in IPv4) is mandatory in IPv6

IPv6 next header

Next header: 8 bit field

- To identify protocol contained within upper layer data such as TCP or UDP

- To Identify the type of header that immediately follows the IPv6 header (**extension headers**)
- Extension headers are to provide an improved option mechanism over IPv4
- IPv6 options are placed in separate extension headers
  - located between the IPv6 header and the transport-layer header in a packet

IPv6 next header

- IPv6 packet may contain zero, one or more extension headers

- but these should be present in their recommended order.

40 bytes header in IPv6 vs 20 bytes header in IPv4

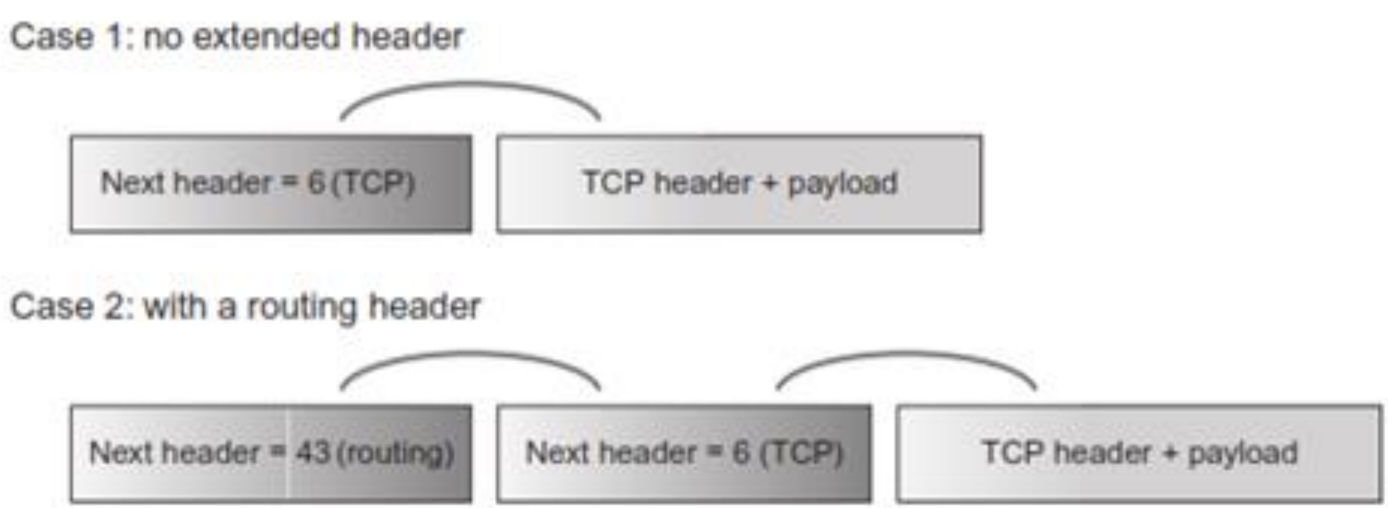
- The 6LoWPAN Working Group specified various header compression schemes to reduce the header overhead

In contrast with IPv4,

- there is no checksum in the IPv6 header.
- Thus all the transport layer protocols are required to compute a checksum taking into account the IPv6 header.
- Thus, the UDP checksum (optional in IPv4) is mandatory in IPv6

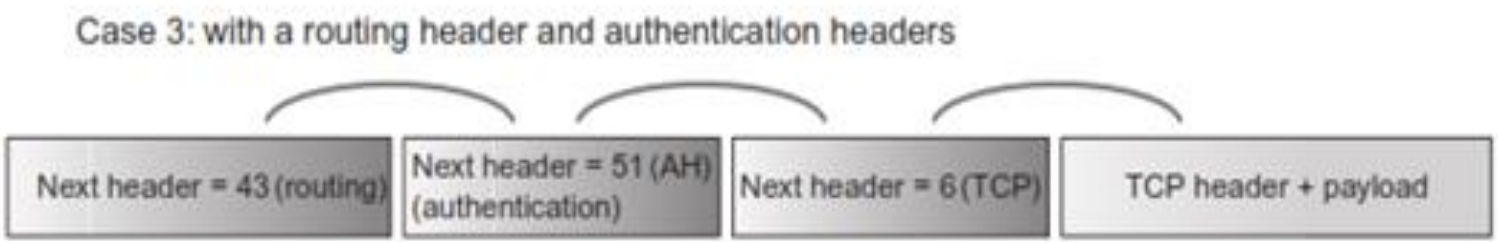
Extended Headers

- IPv6 has a fixed header optionally followed by a daisy chain of headers called extended headers
- **Optional headers follow the fixed header and precede the transport header**
- The next header value simply identifies the type of the following header
- When the next header value is equal to 6, it identifies a TCP header



Extended Headers

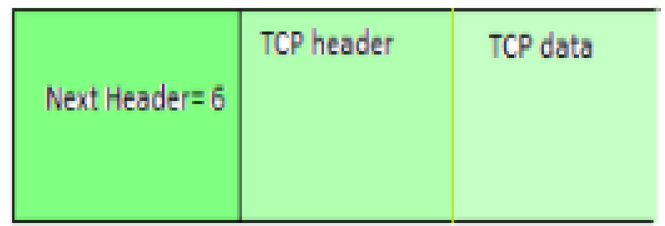
- **Case 3:** a series of three extended headers follows the fixed IPv6 that are daisy-chained.
- The IPv6 next header value is 43, indicating that the first extended header is a routing header,
- the second next header field has a value of 51 that indicates the presence of an authentication header.
- The transport header is specified by the value of 6
- The hop-by-hop header is the only header that must be processed by all the routers along the path including the source and the destination



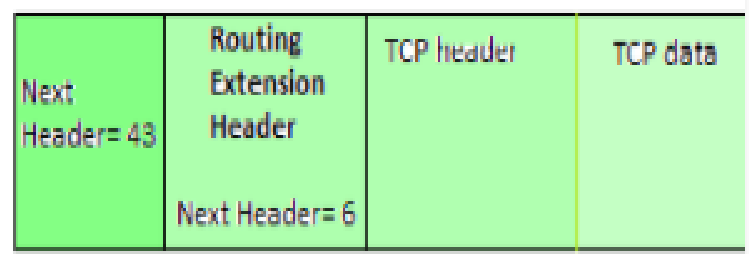
Next header in IPv6

Order	Header Type	Next Header Code
1	Basic IPv6 Header	-
2	Hop-by-Hop Options	0
3	Destination Options (with Routing Options)	60
4	Routing Header	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulation Security Payload Header	50
8	Destination Options	60
9	Mobility Header	135
	No next header	59
Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58

Example: *TCP is used in IPv6 packet*



Example2:



IPv6 support Extended Headers

All IPv6 implementation must support the following extended headers:

- Hop-by-hop options
- Routing (type 0)
- Fragment
- Destination options
- Authentication
- Encapsulating security payload

IPv6 Extended headers

- Extension headers are processed in the order in which they are present.
- **Hop-by-Hop options** header(if present) should always be placed after the IPv6 base header.
  - only extension header that must be processed by every node on the path
  - Used to carry extra information like delivery parameters at each hop on the path
  - Identified by the value of 0 in IPv6 base header

IPv6 next header

Conventions:

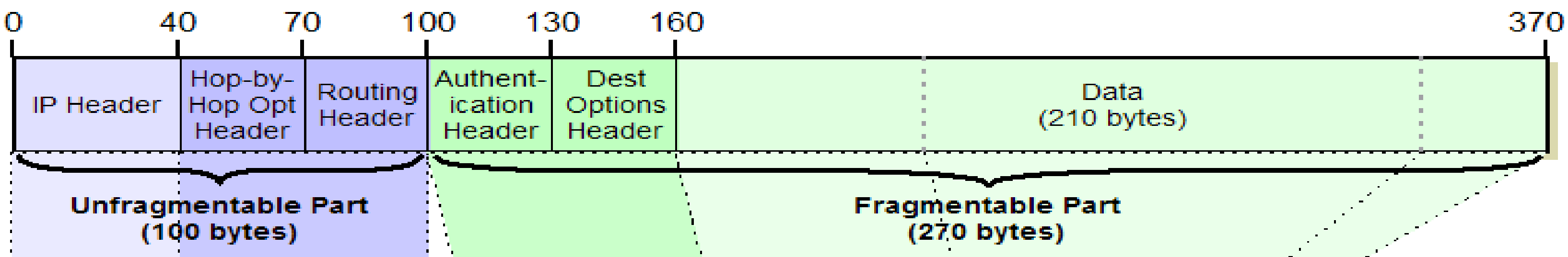
- Any extension header can appear at most once except Destination Header
  - Destination Header is present two times in the list itself.
- If Destination Header is present before Routing Header
  - then it will be examined by all intermediate nodes specified in the routing header.
- If Destination Header is present just above the Upper layer
  - then it will be examined only by the Destination node.

IPv6 Extended Headers

- IPv6 mandates that each link must be able to carry1280-byte packets, which is not always the case in LLN.
- The MTU of IEEE 802.15.4 links is 127 bytes.
- In this case, it is required to handle packet fragmentation and reassembly at the link layer (6LoWPAN Working Group).
- IPv6 should support mechanisms to discover the minimum MTU supported on each link along the path to the destination performed using a procedure called path maximum transmission discovery (PMTU).
- It uses a sequence of ICMP packets along the path until it discovers the minimum MTU along the path.
- An IPv6 source node **fragments** a packet each time its size is larger than the minimum MTU along the path to the destination.

Fragmentation process: Example

- ✓ Suppose an IPv6 datagram is exactly 370 bytes wide,
  - ✓ consisting of a 40-byte IP header,
  - ✓ four 30-byte extension headers, and
  - ✓ 210 bytes of data.



- ✓ Two of the extension headers are unfragmentable, while two are fragmentable.
- ✓ Suppose we need to send this over a link with an MTU of only 230 bytes.

How many fragments will be required?

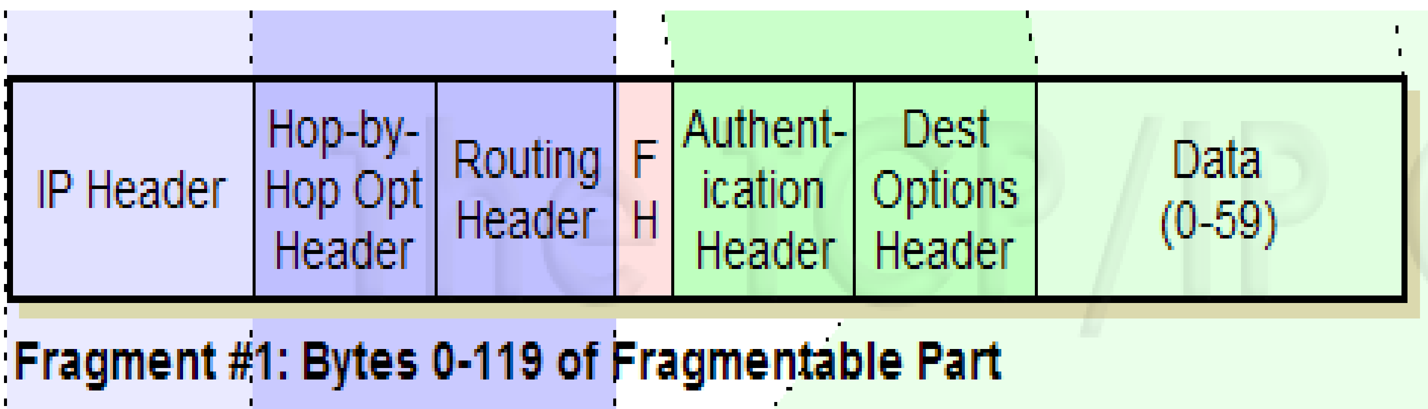
Fragmentation process: Example

- ✓ actually we require three fragments, how?
- ✓ We need to put two 30-byte unfragmentable extension headers in each fragment, and
- ✓ the requirement that each fragment be a length that is a multiple of 8.

Fragmentation process: Example

First Fragment:

- ✓ 100-byte *Unfragmentable Part*,
- ✓ followed by an 8-byte *Fragment* header and
- ✓ the first 120 bytes of the *Fragmentable Part* of the original datagram.
  - ✓ This would contain the two fragmentable extension headers and the first 60 bytes of data.
  - ✓ This leaves 150 bytes of data to send.



Fragmentation process: Example

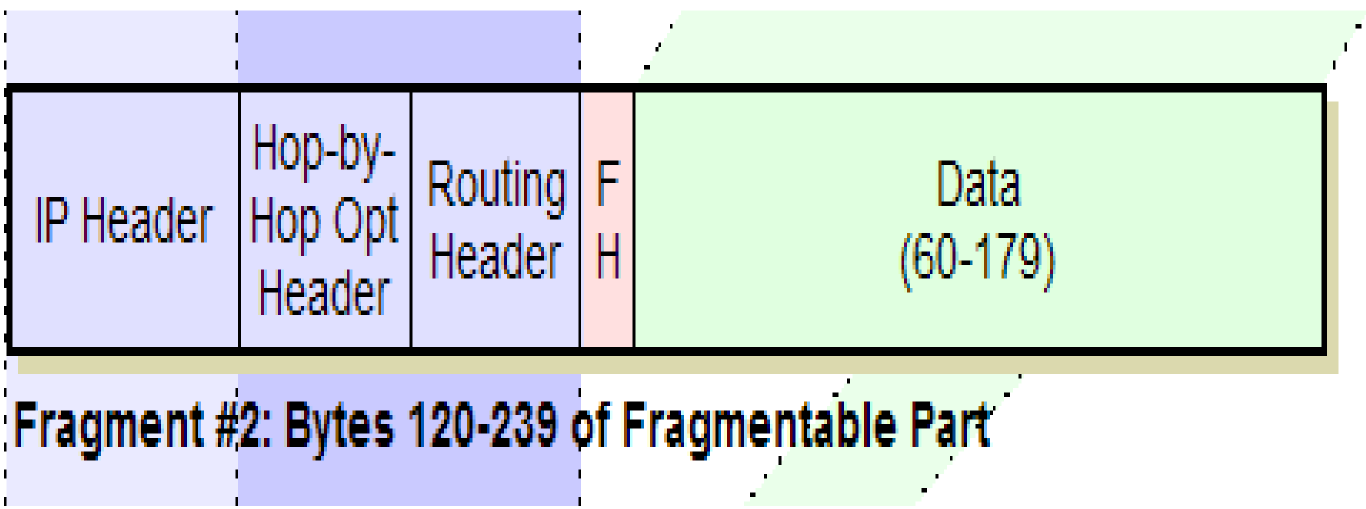
Second Fragment:

✓ again the 100-byte *Unfragmentable Part*,

✓ followed by a *Fragment* header and

✓ 120 bytes of data (bytes 60 to 179).

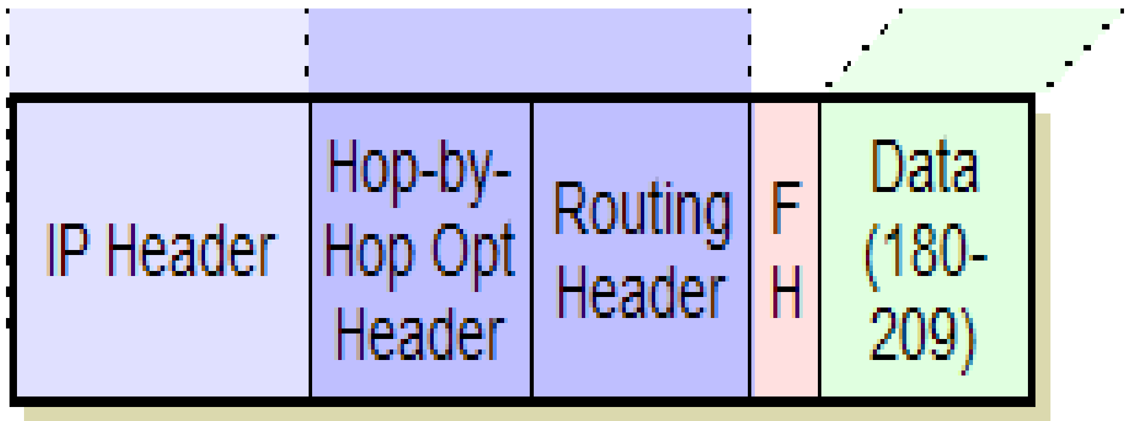
✓ This would leave 30 bytes of data remaining.



Fragmentation process: Example

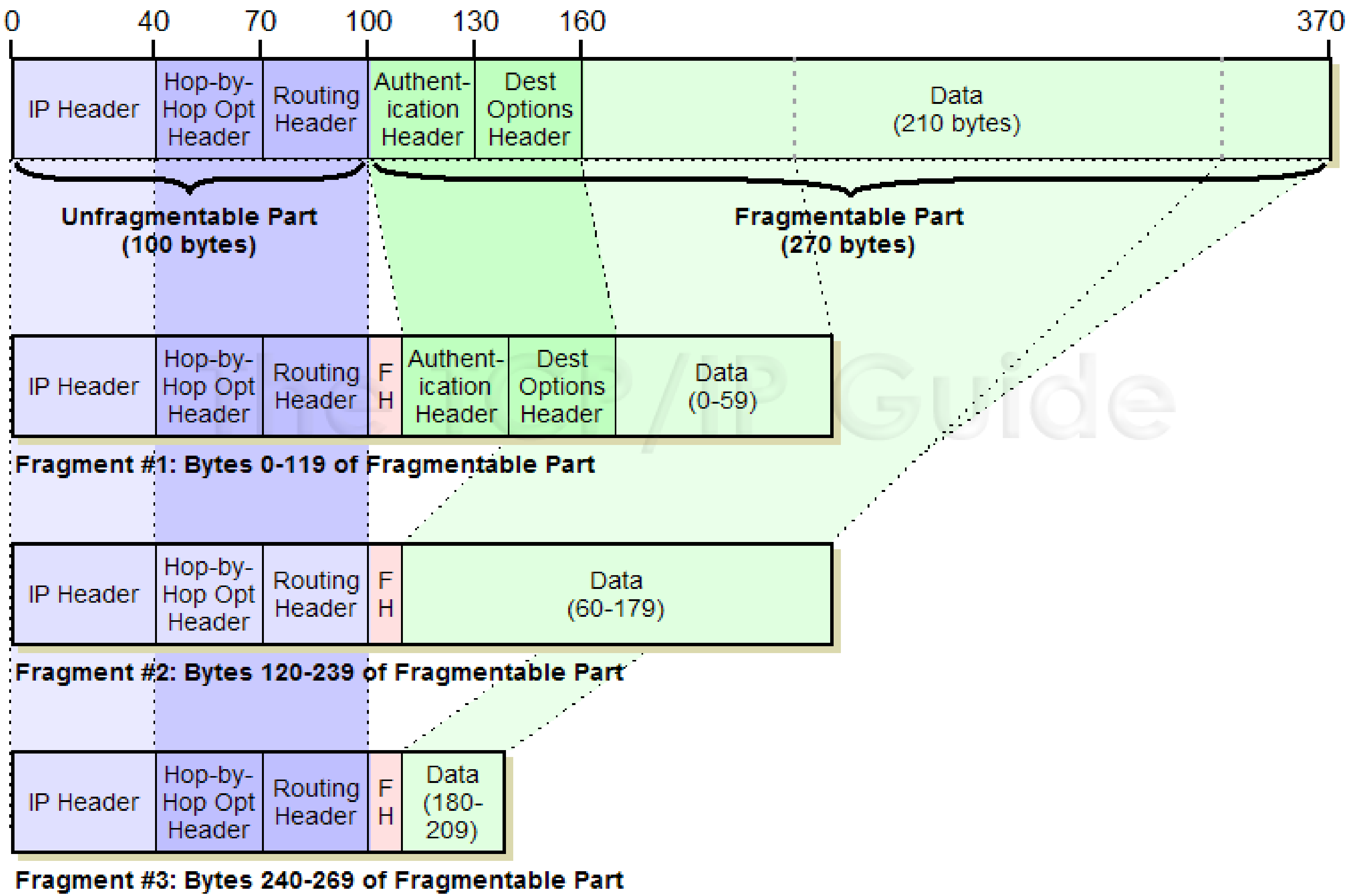
Third Fragment:

- ✓ The last fragment would contain the 100-byte *Unfragmentable Part*,
- ✓ a *Fragment* header and the final 30 bytes of data



Fragment #3: Bytes 240-269 of Fragmentable Part

Fragmentation process



IPv6 Addressing Architecture

A <u>unicast address</u> uniquely identifies a single interface by its address.

- An interface can have multiple unicast addresses and
- must have at least one link-local address
  - an address used on a link between two nodes

An <u>anycast address</u> is an identifier for a set of interfaces:

- a packet sent to an anycast address is only delivered to one of the interfaces of the set,
- typically the closest one according to routing metrics

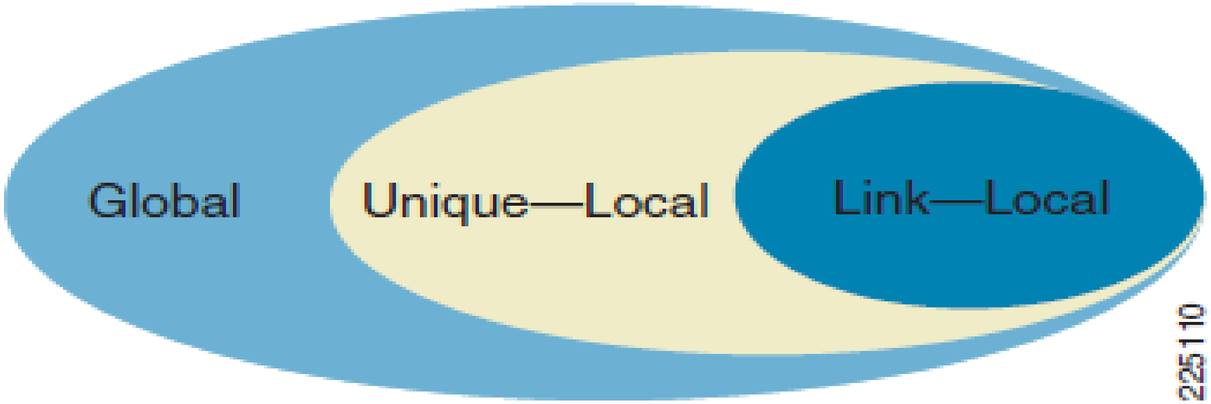
IPv6 Addressing Architecture

multicast address, In contrast to anycast,

- a packet sent to a **multicast address** is delivered to all interfaces identified by the multicast address.
-
- There is no broadcast in IPv6, so multicast addresses are used

Unicast Addresses

- A **unicast address** is made of a subnet prefix and an interface identifier (interface ID).
- Interface IDs are used to identify an interface on a link and thus must be unique on that link;
- it is very common for the interface ID to be identical to the link layer address of the interface
- Global Unicast IPv6 Addresses
- Local Unicast IPv6 Addresses



Global Unicast IPv6 Addresses

- Global unicast addresses have their three leftmost bits set to **001**.
- Consequently, a global unicast address belongs to the range from 2000:: to 3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF.

In most cases,

- the leftmost 64 bits are used to identify the network portion of the address and
- the rightmost 64 bits are used to identify the host portion of the address
- The network portion of the address is subdivided into a
  - 48-bit field (prefix provided by the Service Provider)
  - 16-bit field to allocate subnets within a site (2^{16} available subnets)
  - 64-bit field host part (the interface ID)

Local Unicast IPv6 Addresses

Two types of local unicast IPv6 addresses:

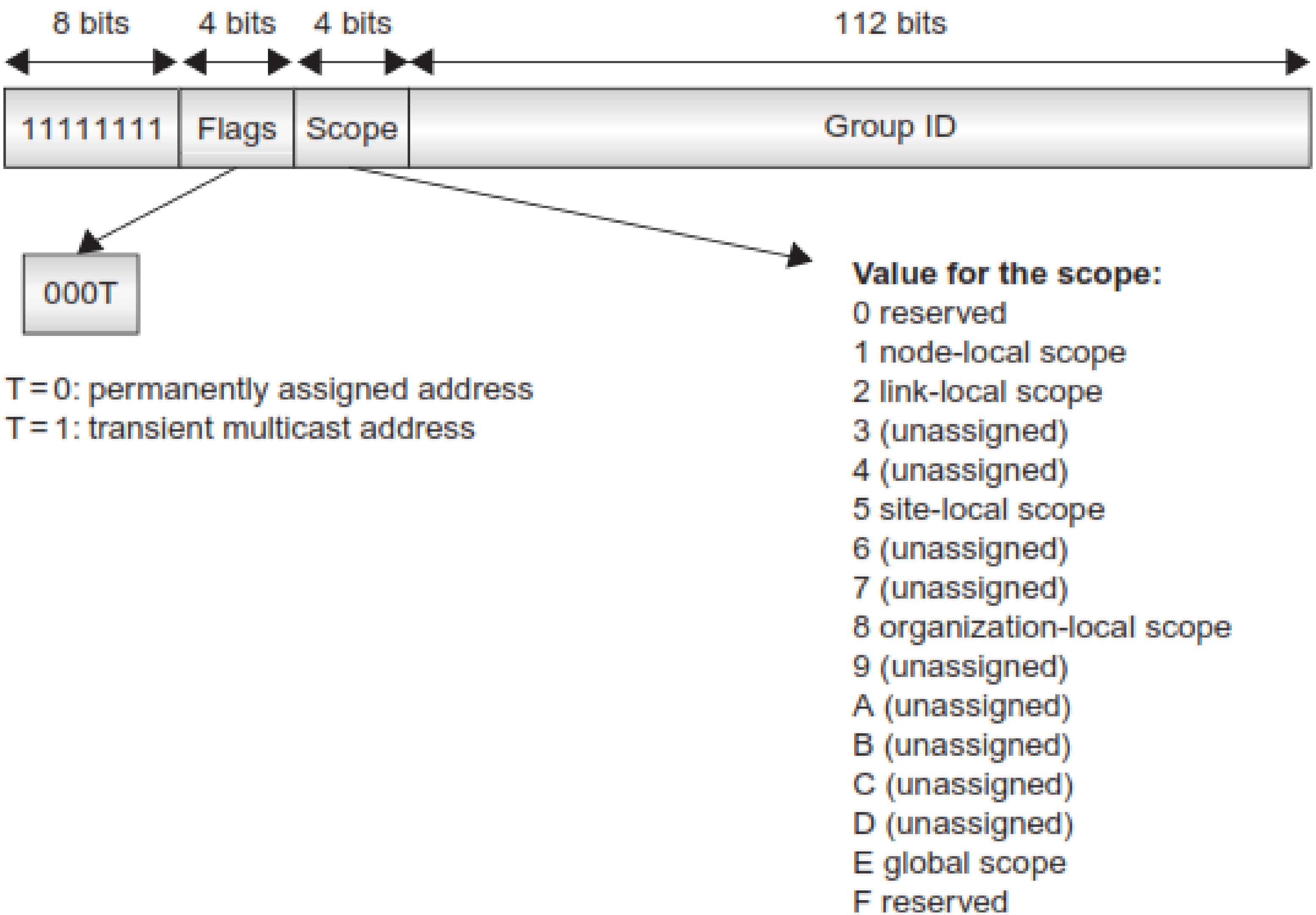
- **Link-local unicast addresses:** used on a single link for autoconfiguration, neighbor discovery, or in the absence of router.
- Since the scope is local
- packets with link-local scope are never forwarded by the router beyond the scope of the link [prefix: FD00::/8]
- **Site or link local address** is an address forwarded within a site that does not need to reach in the Internet.
- Packets with such addresses were not forwarded by routers outside of the site. New implementations must treat site-local addresses as unique local addresses
- A link-local scope unicast address always starts with **FE80:0:0:0** followed by the interface ID.

Anycast Addresses

- an address allocated to a set of interfaces that typically belong to different routers
- When a packet is destined to an anycast address,
  - it is delivered to the closest interface that has this anycast address. ("closest" is determined by the routing protocol)
- It must be assigned to a router not a host and cannot be used as a source address
- **Example:** the subnet-router anycast address.
- This address format is formed by a subnet prefix of n bits that identifies a specific link followed by 128-n bits all set to 0.
- A packet sent to the subnet-router anycast address is delivered to one of the routers on that subnet link

Multicast Addresses

- Multicast addresses are used in many contexts and are very important (IPv6 does not use broadcast addresses)
- A multicast address identifies a group of nodes called a multicast group and must not be used as a source address or in a routing header
- All multicast addresses start with FF (first 8 bits of the address), followed by a 4-bit flag field, a 4-bit scope field, and a 112-bit group ID
- Some of the multicast addresses are reserved


8 bits
4 bits
4 bits
112 bits
11111111
Flags
Scope
Group ID
000T
T = 0: permanently assigned address
T = 1: transient multicast address
Value for the scope:
0 reserved
1 node-local scope
2 link-local scope
3 (unassigned)
4 (unassigned)
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

IPv6 Autoconfiguration

The ability for a node to support auto-configuration is very important,

- especially when the number of nodes is extremely large and
- the nodes are unattended, which is precisely the case in smart object networks.
  - Nodes cannot be manually configured
- The set of auto-configuration features supported by IPv6 is particularly well suited to smart object networks