Internet Protocol version 6 (IPv6)

Network Infrastructures A.A. 2020/21

Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
  - Security consideration
- IPv6 Addressing
  - type of addresses
  - unicast
  - multicast
  - subnetting
- Neighbor Discovery
  - basics
  - messages
  - host conceptual procedures and data structures
  - Router Discovery
  - Neighbor Discovery
  - Security Threats
- Host Autoconfiguration
  - SLAAC
  - DHCPv6
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
  - Security consideration
- IPv6 Addressing
  - type of addresses
  - unicast
  - multicast
  - subnetting
- Neighbor Discovery
  - basics
  - messages
  - host conceptual procedures and data structures
  - Router Discovery
  - Neighbor Discovery
  - Security Threats
- Host Autoconfiguration
  - SLAAC
  - DHCPv6
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

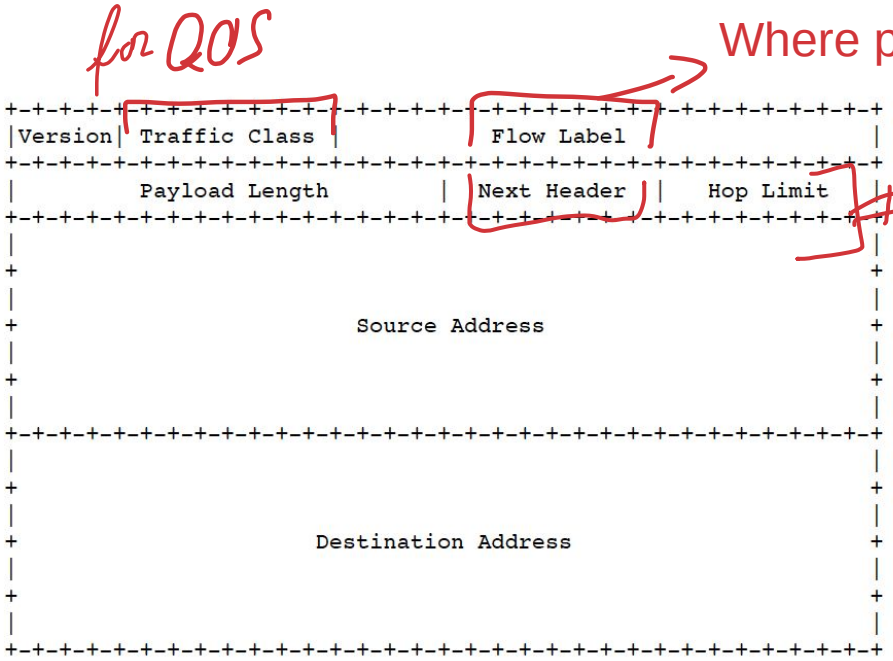
Design Principle

SAPIENZA
UNIVERSITÀ DI ROMA

- IP version 6 (IPv6) is a new version of the Internet Protocol (IP), designed as the successor to IP version 4 (IPv4)
- The changes from IPv4 to IPv6 fall primarily into the following categories:
  - expanded addressing capabilities → wrt v4
  - header format simplification
  - improved support for extensions and options
  - flow labeling capability → ADD A LABEL TO EACH PACKET
  - authentication and privacy capabilities

IPv6 Header Format

SAPIENZA
UNIVERSITÀ DI ROMA



IPv6 Header Format

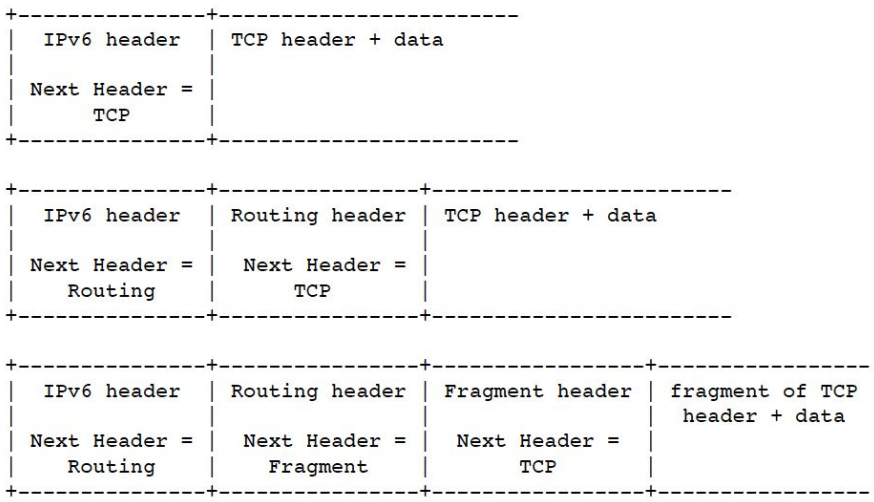
SAPIENZA
UNIVERSITÀ DI ROMA

- **Version**
  - (4-bit) Internet Protocol version number = 6
- **Traffic Class**
  - (8-bit) Traffic Class field
- **Flow Label**
  - (20-bit) flow label
- **Payload Length**
  - (16-bit) length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets
- **Next Header**
  - (8-bit) identifies the type of header immediately following the IPv6 header
- **Hop Limit**
  - (8-bit) decremented by 1 by each node that forwards the packet. When forwarding, the packet is discarded if Hop Limit was zero when received or is decremented to zero
- **Source Address**
  - (128-bit) address of the originator of the packet
- **Destination Address**
  - (128-bit) address of the intended recipient of the packet (possibly not the ultimate recipient, if a Routing header is present)

IPv6 Extension Headers

SAPIENZA UNIVERSITÀ DI ROMA

- In IPv6, optional internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper layer header in a packet



Processing an IPv6 packet

by intermediate nodes.

SAPIENZA UNIVERSITÀ DI ROMA

- Extension headers are not processed, inserted, or deleted by any node along a packet's delivery path, until the packet reaches the node identified in the Destination Address field of the IPv6 header
- At the destination node, normal demultiplexing on the Next Header field of the IPv6 header invokes the module to process
  - the first extension header, or
  - the upper-layer header if no extension header is present
- The contents and semantics of each extension header determine whether or not to proceed to the next header
- Extension headers must be processed strictly in the order they appear in the packet

(some) Extension Headers

SAPIENZA
UNIVERSITÀ DI ROMA

- A full implementation of IPv6 includes implementation of the following extension headers:
  - Hop-by-Hop Options
  - Fragment
  - Destination Options
  - Routing
  - Authentication
  - Encapsulating Security Payload

Order matter because router compute algorithm with decisions based on what they have to do next.

Extension Header Order

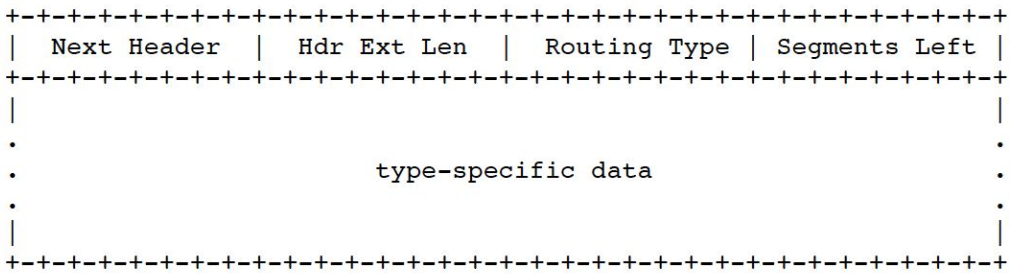
SAPIENZA
UNIVERSITÀ DI ROMA

- When more than one extension header is used in the same packet, it is recommended that those headers appear in the following order:
  1. IPv6 header
  2. Hop-by-Hop Options header
  3. Destination Options header
  4. Routing header
  5. Fragment header
  6. Authentication header
  7. Encapsulating Security Payload header
  8. Destination Options header
  9. Upper-Layer header

Routing Header

SAPIENZA
UNIVERSITÀ DI ROMA

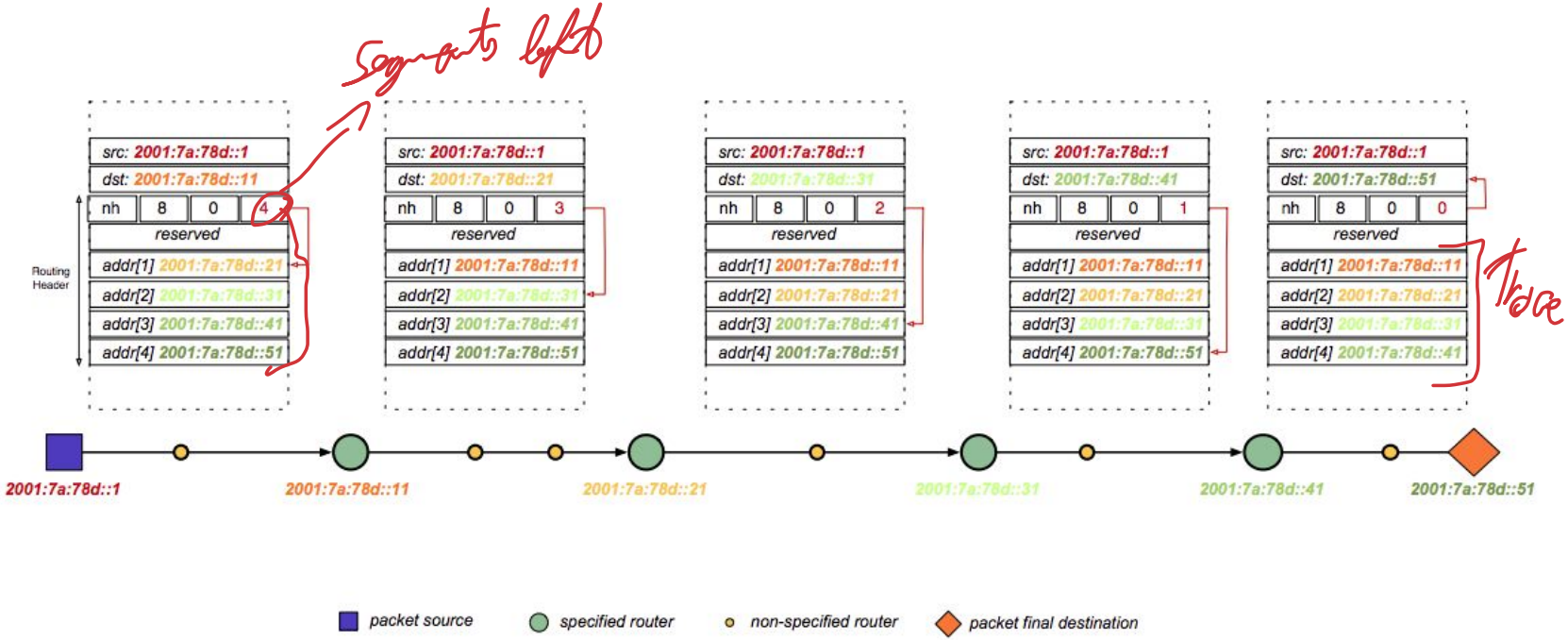
- The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination



- **Routing Type**
  - (8-bit) identifier of a particular Routing header variant
- **Segments Left**
  - (8-bit) number of explicitly listed intermediate nodes still to be visited before reaching the final destination

Type 0 Routing Header

SAPIENZA
UNIVERSITÀ DI ROMA



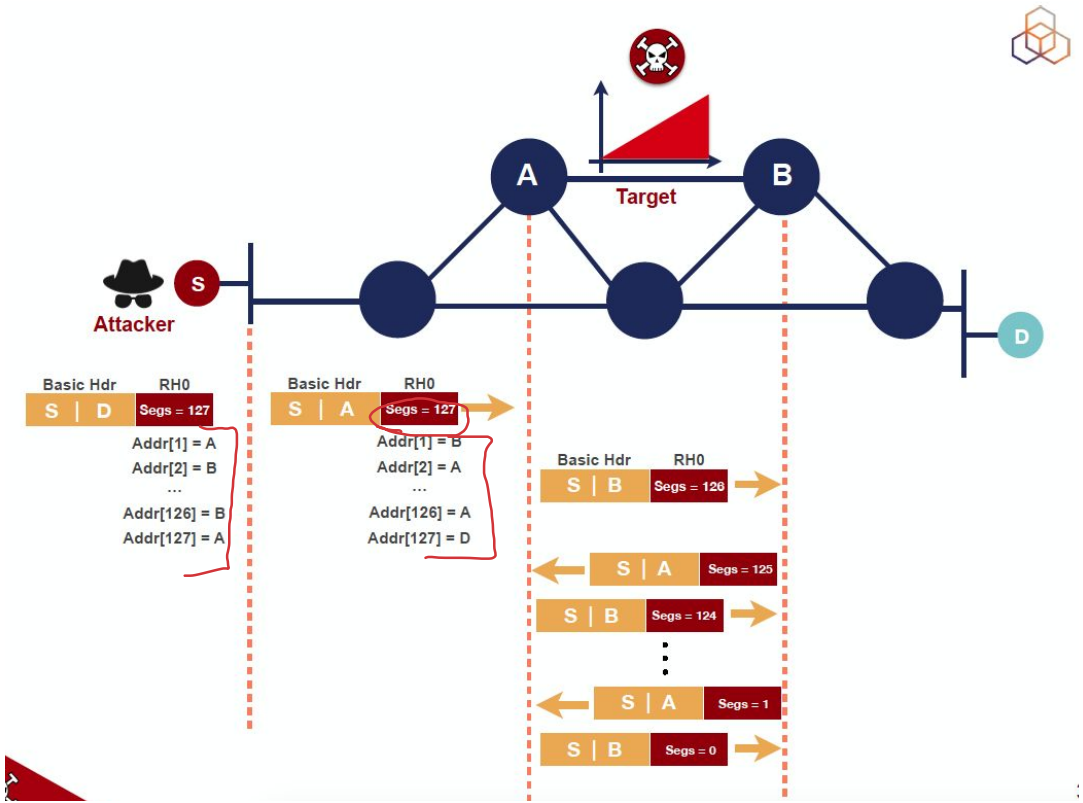
Type 0 Routing Header

SAPIENZA UNIVERSITÀ DI ROMA

- A single RH0 may contain multiple intermediate node addresses, and the same address may be included more than once in the same RH0
- This allows a packet to be constructed such that it will oscillate between two RH0-processing hosts or routers many times
- This allows a stream of packets from an attacker to be amplified along the path between two remote routers, which could be used to cause congestion along arbitrary remote paths and hence act as a denial-of-service mechanism

Type 0 Routing Header

SAPIENZA
UNIVERSITÀ DI ROMA



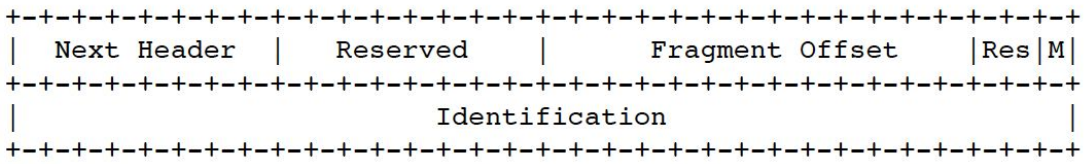
Fragment Header

SAPIENZA
UNIVERSITÀ DI ROMA

- The Fragment header is used by an IPv6 source to send a packet larger than would fit in the path MTU to its destination
- Unlike IPv4, fragmentation in IPv6 is performed only by source nodes, not by routers along a packet's delivery path
  - **pros:** less processing, less delay, overhead reduction
  - **cons:** security threats, need of path MTU discovery procedure
- IPv6 attempts to minimise the use of fragmentation
  - by minimising the supported MTU size
  - by allowing only the hosts to fragment datagrams
- IPv6 requires that every link in the Internet have an MTU of 1280 octets or greater

Fragment Header

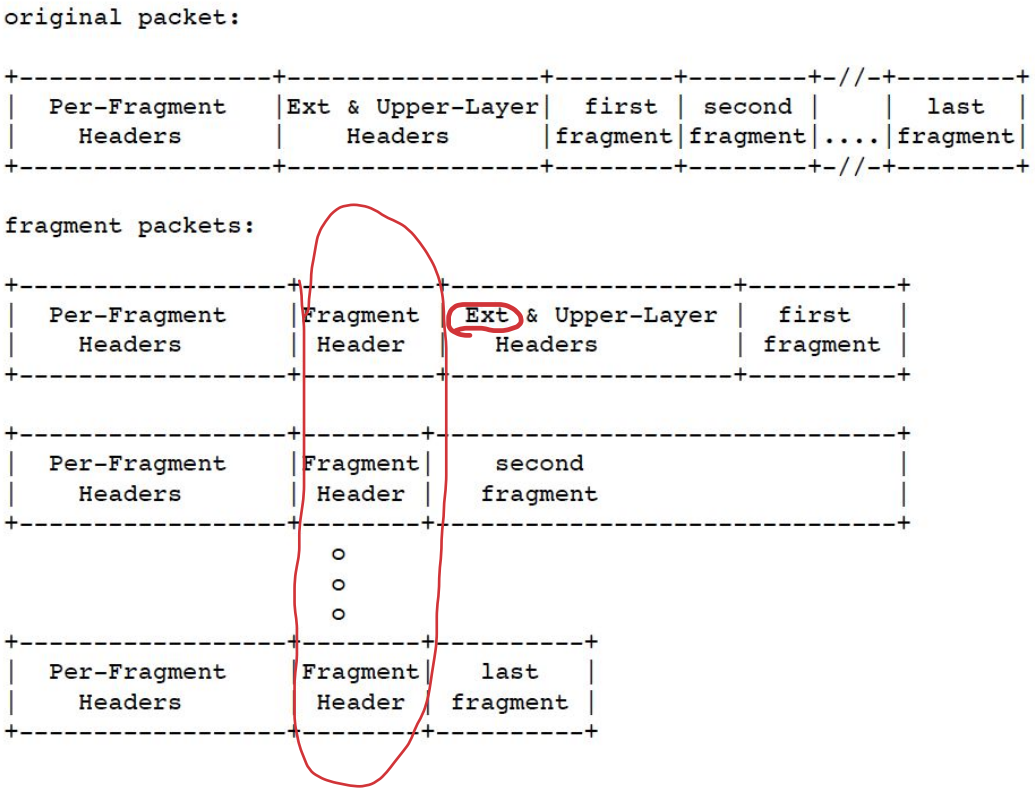
SAPIENZA
UNIVERSITÀ DI ROMA



- **Fragment Offset** defines the offset, in 8-octet units, of the data following this header relative to the start of the Fragmentable Part of the original packet
- **M flag** is a bit set to 1 when more fragments will follow or 0 if this is the last fragment
- **Identification** defines the fragments which belong to the same packet
  - this number must be different than that of any other fragmented packet sent recently with the same Source Address and Destination Address

Fragmentation

SAPIENZA
UNIVERSITÀ DI ROMA



Reassembly

SAPIENZA
UNIVERSITÀ DI ROMA

- An original packet is reassembled only from fragment packets that have the same Source Address, Destination Address, and Fragment Identification
- **rules**
  - The Per-Fragment headers of the reassembled packet consists of all headers up to, but not including, the Fragment header of the first fragment packet with the following two changes:
    - The Next Header field of the last header of the Per-Fragment headers is obtained from the Next Header field of the first fragment's Fragment header
    - The Payload Length of the reassembled packet is computed from the length of the Per-Fragment headers and the length and offset of the last fragment.

PL.orig = PL.first - FL.first - 8 + (8 * FO.last) + FL.last

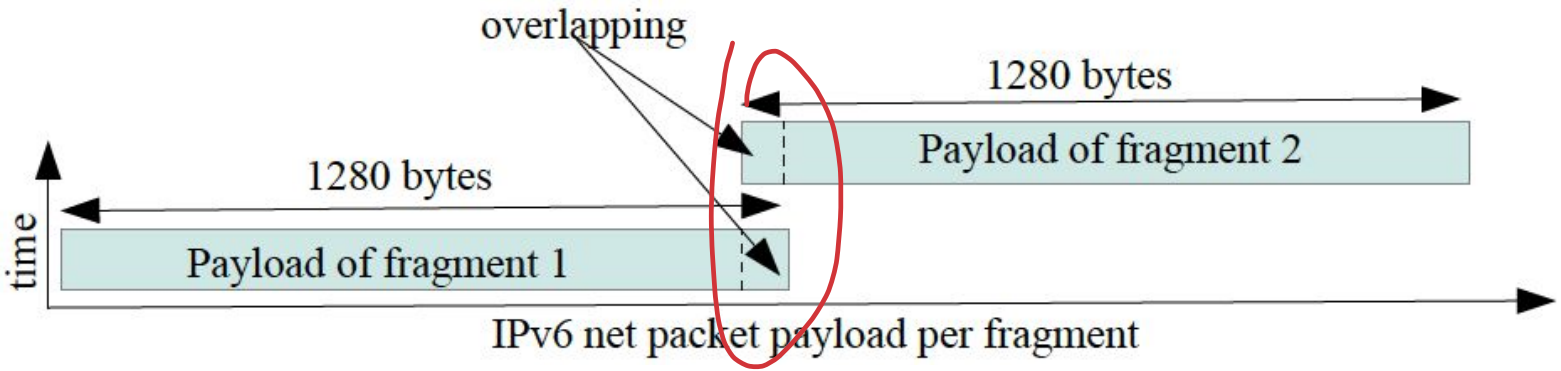
Errors in reassembly

SAPIENZA
UNIVERSITÀ DI ROMA

- If insufficient fragments are received to complete reassembly of a packet within 60 seconds of the reception of the first arriving fragment of that packet
  - reassembly must be abandoned
  - all received fragments are discarded
- the length of a fragment is not a multiple of 8 octets and the M flag is 1
  - discard the frame and send an ICMP Parameter Problem to the source
- fragments of a packet are overlapped
  - reassembly of that packet must be abandoned
  - all the received fragments must be discarded
  - no ICMP error messages should be sent

Security Threats related to Fragmentation

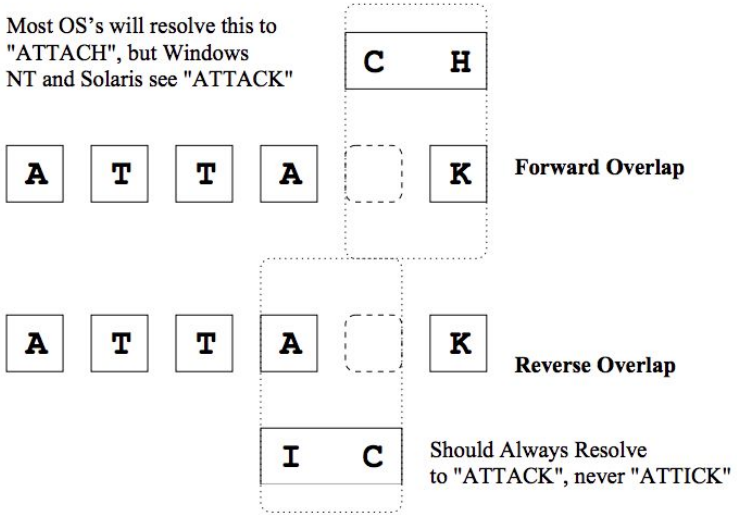
SAPIENZA
UNIVERSITÀ DI ROMA



Security Threats related to Fragmentation

SAPIENZA
UNIVERSITÀ DI ROMA

- Fragmentation overlap occurs when fragments of differing sizes arrive out of order and in overlapping positions
- Reconstruction process in different hosts might generate different packets
- This presents problems for an IDS
- An attacker that understands the specific inconsistency between an end system and an IDS can obscure her attack



Packet Size Issues

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 requires that every link in the Internet have an MTU of 1280 octets or greater
- On any link that cannot convey a 1280-octet packet in one piece, link specific fragmentation and reassembly must be provided at a layer below IPv6
- It is strongly recommended that IPv6 nodes implement Path MTU Discovery, in order to discover and take advantage of path MTUs greater than 1280 octets
- A minimal IPv6 implementation may simply restrict itself to sending packets no larger than 1280 octets, and omit implementation of Path MTU Discovery

Path MTU Discovery

SAPIENZA
UNIVERSITÀ DI ROMA

- Path MTU (PMTU) is equal to the minimum link MTU of all the links in a path

- When one IPv6 node has a large amount of data to send to another node, the data is transmitted in a series of IPv6 packets
- These packets can have a size
  - less than or equal to the PMTU
  - larger than PMTU
    - fragmentation is required

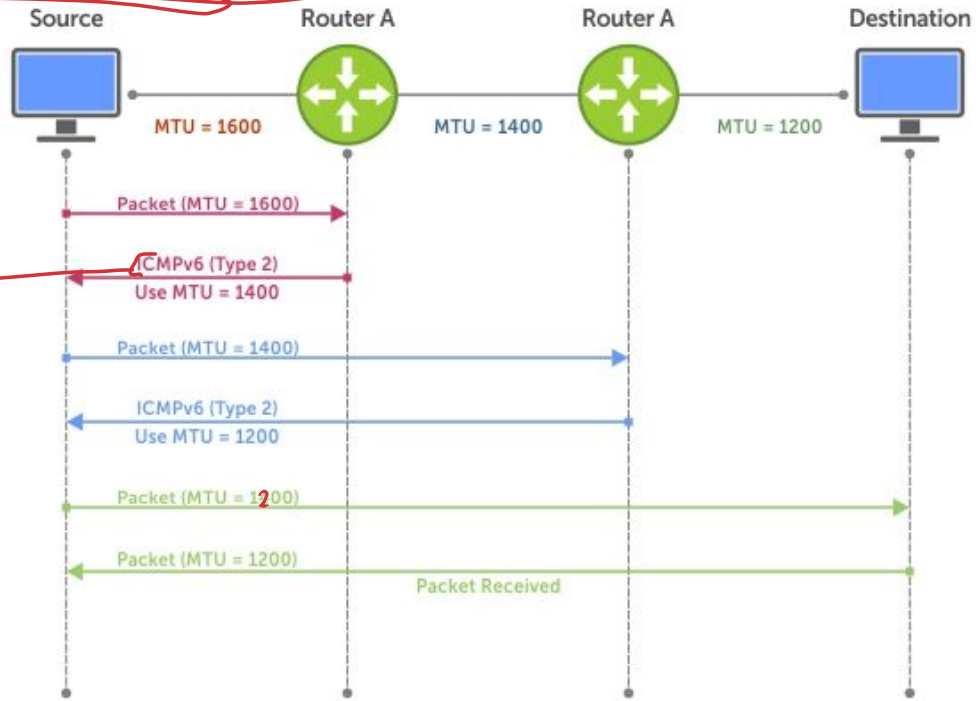
Path MTU Discovery

SAPIENZA
UNIVERSITÀ DI ROMA

- It is preferable that a source node sends packets having the largest size that can successfully traverse the path to the destination without the need for IPv6 fragmentation
- **Problem**
  - Source node does not know the PMTU
- **Solution**
  - Path MTU Discovery
- PMTUD it is an iterative procedure based on ICMP Packet Too Big messages that allows the source to discover the PMTU

Path MTU Discovery

SAPIENZA
UNIVERSITÀ DI ROMA



USED FOR C
PING ECHO ETC.

Path MTU Discovery

SAPIENZA
UNIVERSITÀ DI ROMA

- The PMTU of a path may change over time, due to changes in the routing topology
- Reductions of the PMTU are detected by Packet Too Big messages
- To detect increases in a path's PMTU, a node periodically increases its assumed PMTU

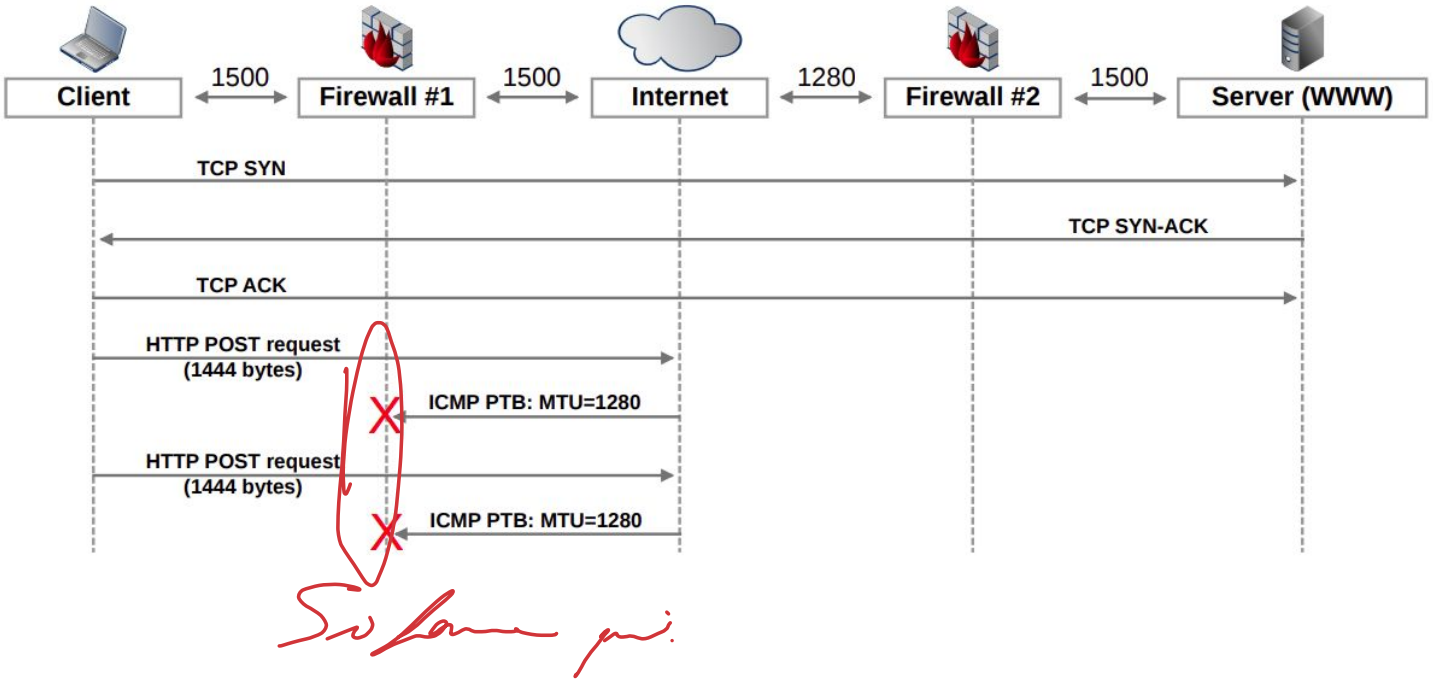
PMTUD and Black Holes

SAPIENZA
UNIVERSITÀ DI ROMA

- Black holes occur when the network infrastructure is UP but connection among end devices is DOWN
  - hard to detect
  - many possible causes (misconfiguration, bugs in software, etc.)
- Problems with the PMTUD can lead to the creation of a black hole in an IPv6 network
- Possible causes
  - routers not sending PTB messages
  - firewalls dropping PTB messages
  - firewalls dropping fragmented packets

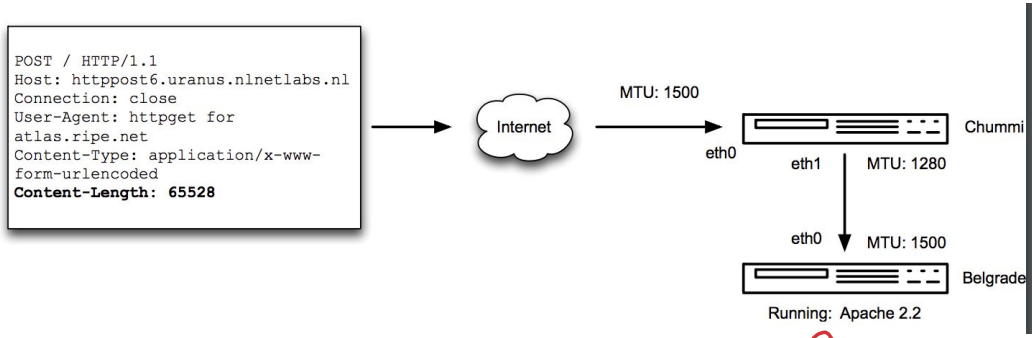
PMTUD and Black Holes: ICMP PTB filtering

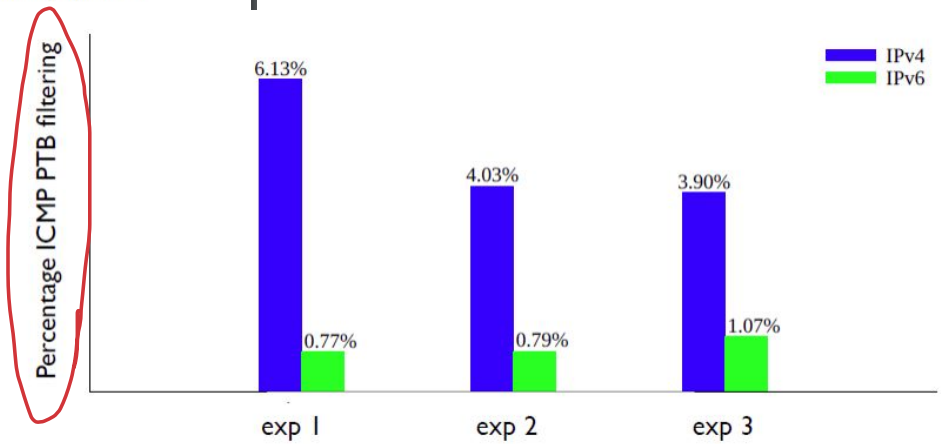
SAPIENZA UNIVERSITÀ DI ROMA



PMTUD and Black Holes: ICMP PTB filtering

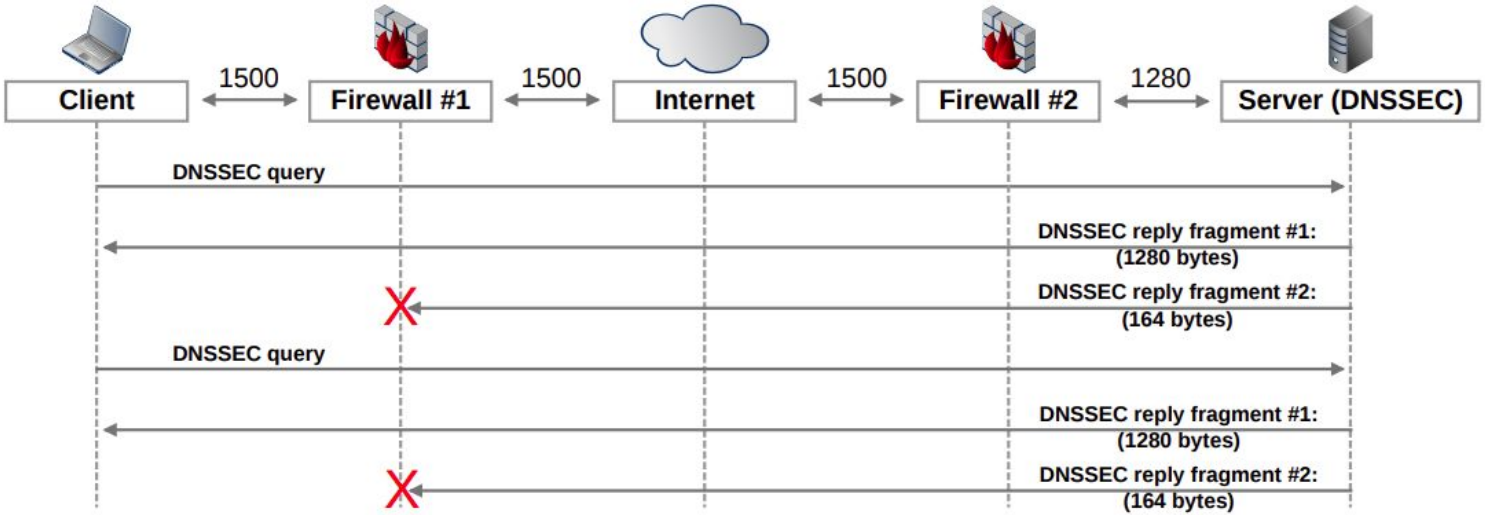
SAPIENZA UNIVERSITÀ DI ROMA





PMTUD and Black Holes: IP fragment filtering

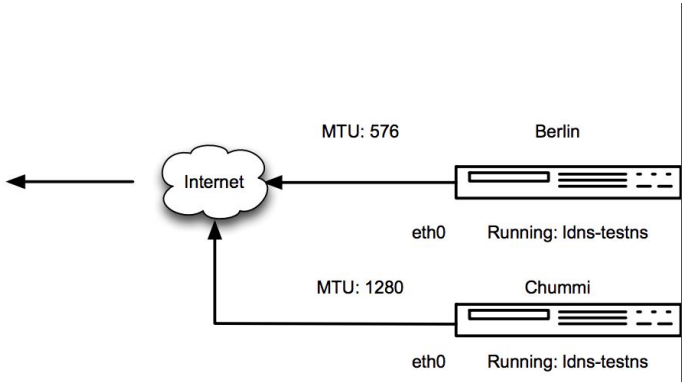
SAPIENZA UNIVERSITÀ DI ROMA

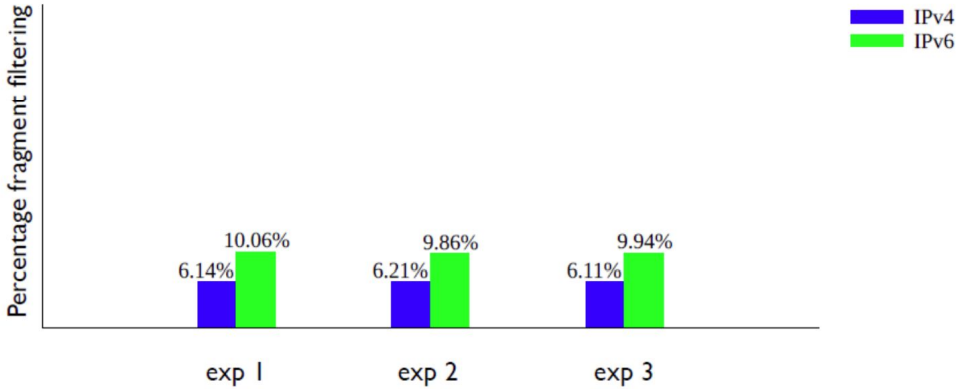


PMTUD and Black Holes: IP fragment filtering

SAPIENZA UNIVERSITÀ DI ROMA

```
version.bind. 60 CH TXT

1,002,003,004,005,006,007,008,009,
010,011,012,013,014,015,016,017,01
8,019,020,021,022,023,024,025,

[truncated]

347,348,349,350,351,352,353,354,35
5,356,357,358,359,360,361,362,363,
364,365,366,367,368,369,370,371,37
2,373,374,375,376,377,378,379,380,
381,382,383

MSG SIZE snd: 1590
```





IPv6 Security Considerations

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 has security properties that are similar to IPv4
  - Eavesdropping, where on-path elements can observe the whole packet of each IPv6 datagram
  - Packet insertion, where the attacker forges a packet with some chosen set of properties and injects it into the network
  - Packet modification, where the attacker removes a packet from the wire, modifies it, and re-injects it into the network
  - Man-in-the-middle (MITM) attacks, where the attacker subverts the communication stream in order to pose as the sender to receiver and the receiver to the sender
  - Denial-of-service (DoS) attacks, where the attacker sends large amounts of legitimate traffic to a destination to overwhelm it

Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
  - Security consideration
- IPv6 Addressing
  - type of addresses
  - unicast
  - multicast
  - subnetting
- Neighbor Discovery
  - basics
  - messages
  - host conceptual procedures and data structures
  - Router Discovery
  - Neighbor Discovery
  - Security Threats
- Host Autoconfiguration
  - SLAAC
  - DHCPv6
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

IPv6 Address

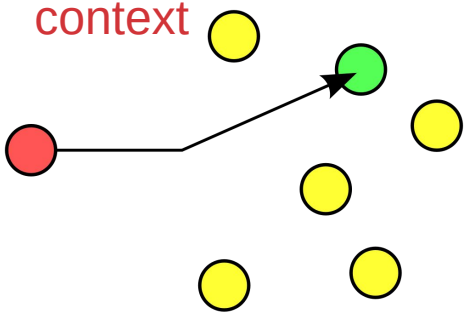
SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 addresses are 128-bit identifiers for interfaces and sets of interfaces
- There are three types of addresses:
  - **Unicast**
    - An identifier for a single interface
    - A packet sent to a unicast address is delivered to the interface identified by that address
  - **Anycast**
    - An identifier for a set of interfaces (typically belonging to different nodes)
    - A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one)
  - **Multicast**
    - An identifier for a set of interfaces
    - A packet sent to a multicast address is delivered to all interfaces identified by that address
- There are no broadcast addresses in IPv6

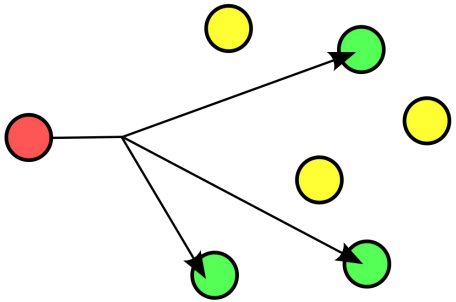
IPv6 Addresses

SAPIENZA
UNIVERSITÀ DI ROMA

No one else can give you that context



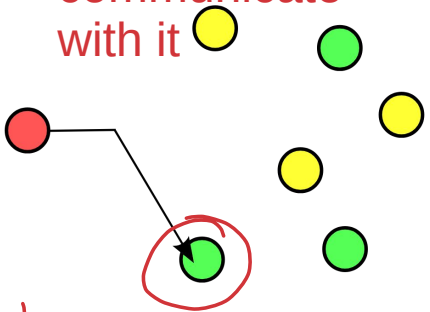
Unicast



Multicast

(e.g.) streaming

You want to communicate with it



(e.g.) NEAREST

Anycast

Text Representation of Addresses

SAPIENZA
UNIVERSITÀ DI ROMA

- The general form for an IPv6 address is x:x:x:x:x:x:x:x (8)
  - 'x's are one to four hexadecimal digits of the eight 16-bit pieces of the address

It would be impossible writing it in binary

- **Examples**

ABCD:EF01:2345:6789:ABCD:EF01:2345:6789

2001:DB8:0:0:8:800:200C:417A

0000 0000 → 0008 → 0800

can remove leading zeros

  - it is not necessary to write the leading zeros in an individual field
  - there must be at least one numeral in every field

Address Compression

SAPIENZA UNIVERSITÀ DI ROMA

- It is common for addresses to contain long strings of zero bits
  - the use of "::" indicates one or more groups of 16 bits of zeros
  - the "::" can only appear once in an address

2001:DB8:0:0:8:800:200C:417	2001:DB8::8:800:200C:417	*a unicast address*
FF01:0:0:0:0:0:0:101	FF01::101	*a multicast address*
0:0:0:0:0:0:0:1	::1	*the loopback address*
0:0:0:0:0:0:0:0	::	*the unspecified address*

IPv6 Address Prefix

SAPIENZA
UNIVERSITÀ DI ROMA

- An IPv6 address prefix is represented by the notation:

  ipv6-address/prefix-length

  - **ipv6-address** is an IPv6 address
  - **prefix-length** is a decimal value specifying how many of the leftmost contiguous bits of the address comprise the prefix
- **Examples**
  - 2001:0DB8:0000:CD30:0000:0000:0000:0000/60
  - 2001:0DB8::CD30:0:0:0:0/60
  - 2001:0DB8:0:CD30::/60

Don’t exist classes

Prefix Representation

SAPIENZA
UNIVERSITÀ DI ROMA

2001:0DB8:0000:CD30:0000:0000:0000:0000/60

- the following are NOT legal representations of the above prefix

2001:0DB8:0:CD3/60

2001:0DB8::CD30/60 -> 2001:0DB8:0000:0000:0000:0000:0000:CD30

2001:0DB8::CD3/60 -> 2001:0DB8:0000:0000:0000:0000:0000:0CD3

Address Type Identification

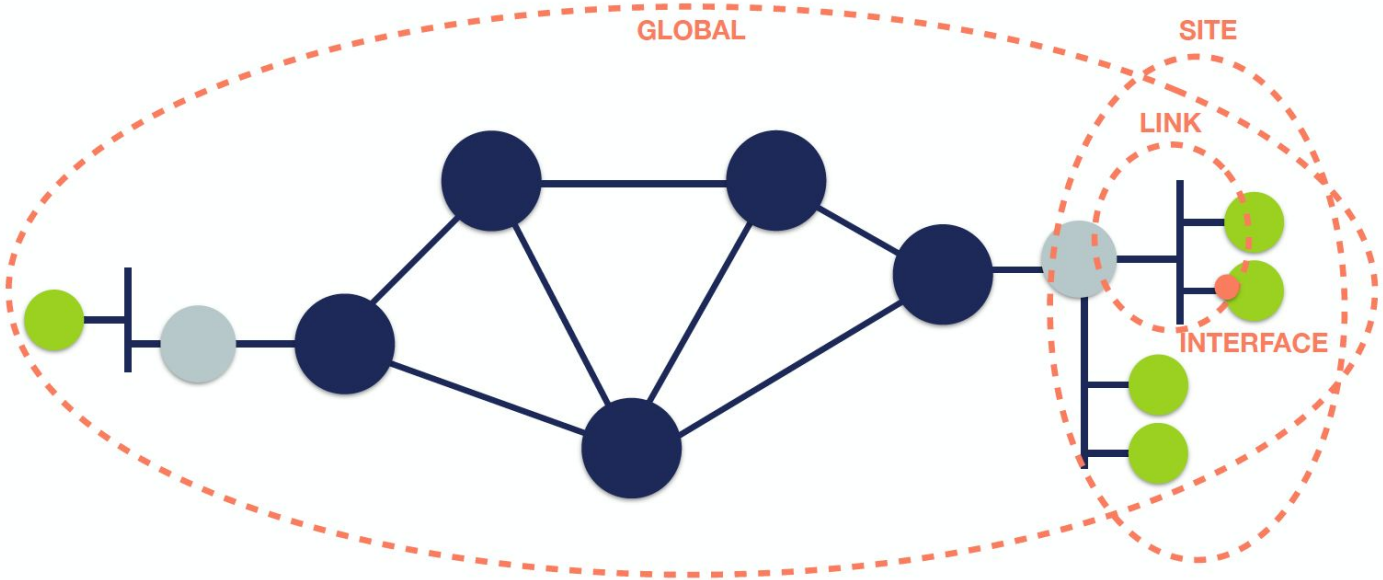
SAPIENZA
UNIVERSITÀ DI ROMA

Address type	Binary prefix	IPv6 notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Global Unicast	(everything else)	

- Anycast addresses are taken from the unicast address spaces

Address Scope

SAPIENZA
UNIVERSITÀ DI ROMA



Interface Identifiers

SAPIENZA
UNIVERSITÀ DI ROMA

- Interface identifiers in IPv6 unicast addresses are used to identify interfaces on a link
- They are required to be unique within a subnet prefix
- An interface's identifier can be derived directly from that interface's link-layer address by using the following process
  1. split the MAC address in two strings 24 bit long
  2. insert the hexadecimal string FFFE in between the two obtained parts
  3. flip the 7th most significant bit
- Interface Identifier is considered for unicast addresses, except those starting with 000

Interface Identifiers

SAPIENZA
UNIVERSITÀ DI ROMA

- Example

39	a7	94	07	cb	d0
00111001	10100111	10010100	00000111	11001011	11010000

OUI — Device ID

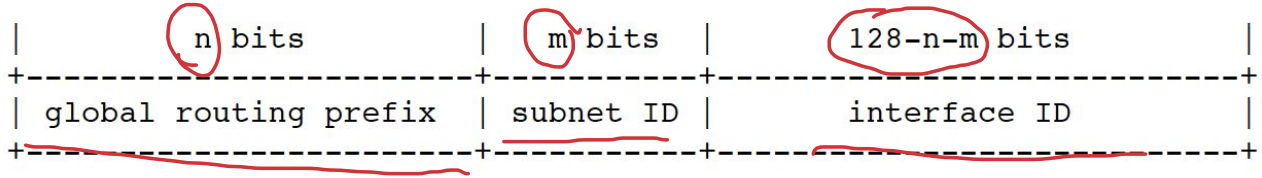
SPLIT

1	00111001	10100111	10010100			00000111	11001011	11010000
2	00111001	10100111	10010100	11111111	11111110	00000111	11001011	11010000
3	00111011	10100111	10010100	11111111	11111110	0000011	11001011	11010000
	3b	a7	94	ff	fe	07	cb	d0

Global Unicast Addresses

SAPIENZA
UNIVERSITÀ DI ROMA

- The general format for **IPv6 Global Unicast** addresses is as follows:



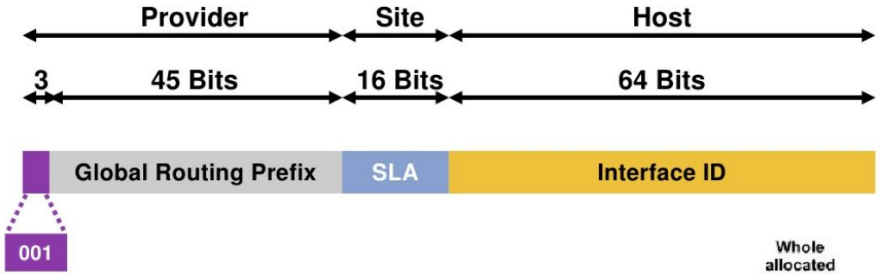
  where:

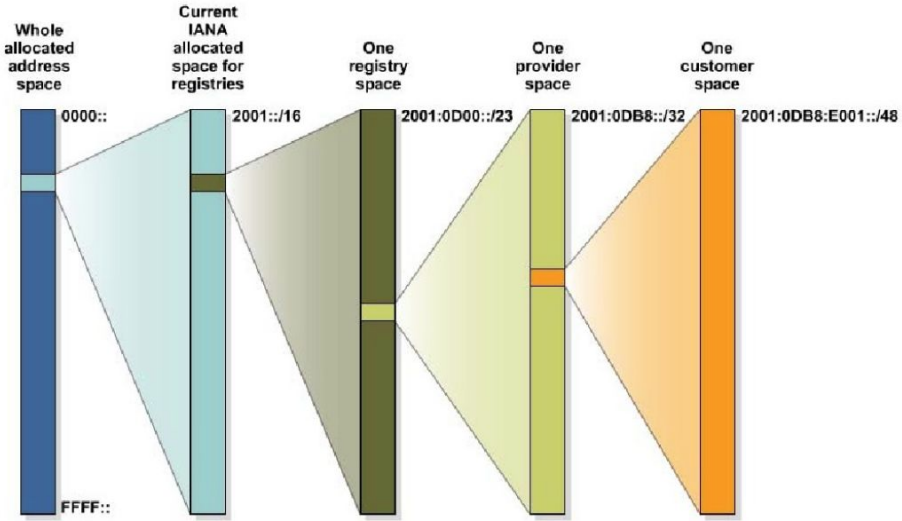
  - the global routing prefix is a value assigned to a site
  - the subnet ID is an identifier of a link within the site
  - interface ID is determined starting from the link layer address

- All Global Unicast addresses other than those that start with binary 000 have a 64-bit interface ID field (i.e., n + m = 64)

Global Unicast Addresses

SAPIENZA
UNIVERSITÀ DI ROMA

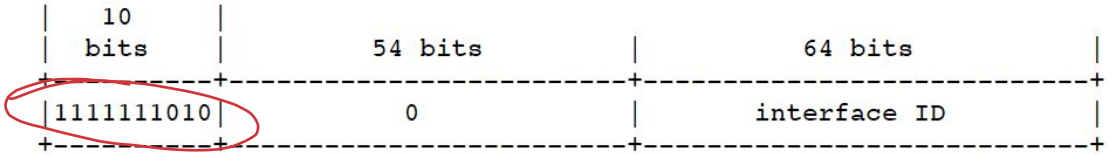




Link-Local IPv6 Unicast Addresses

SAPIENZA UNIVERSITÀ DI ROMA

- **Link-Local addresses** are designed to be used for addressing on a single link for purposes such as
  - automatic address configuration
  - neighbor discovery
  - when no routers are present

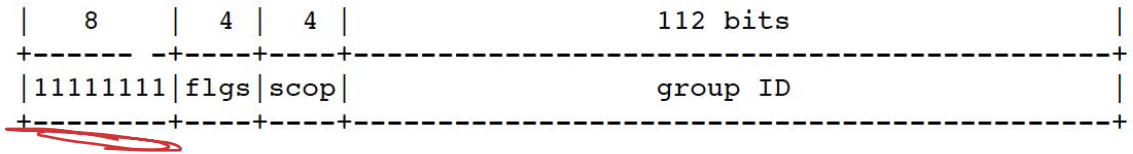


- Routers must not forward any packets with Link-Local source or destination addresses to other links

Multicast Addresses

SAPIENZA
UNIVERSITÀ DI ROMA

- An **IPv6 multicast address** is an identifier for a group of interfaces
- An interface may belong to any number of multicast groups



- **flgs** is a set of 4 flags: 0 | R | P | T
  - T = 0, permanent address (well-known)
  - T = 1, non-permanent address
  - R, P used for special purposes

Multicast Addresses

SAPIENZA
UNIVERSITÀ DI ROMA

- **scop** is a 4-bit multicast scope value used to limit the scope of the multicast group
  - 1 - Interface-Local scope
    - spans only a single interface on a node (loopback)
  - 2 - Link-Local scope
    - spans the same topological region
  - 4 - Admin-Local scope
  - 5 - Site-Local scope
    - span a single site
  - 8 - Organization-Local scope
    - span multiple sites belonging to a single organization
  - E - Global scope

Multicast Addresses: an example

SAPIENZA UNIVERSITÀ DI ROMA

- The "NTP servers group" is assigned a permanent multicast address with a group ID of 101 (hex), then
  - FF02:0:0:0:0:0:0:101
    - means all NTP servers on the same link as the sender
  - FF05:0:0:0:0:0:0:101
    - means all NTP servers in the same site as the sender
  - FF0E:0:0:0:0:0:0:101
    - means all NTP servers in the Internet

Some Well-Known Multicast Addresses

SAPIENZA UNIVERSITÀ DI ROMA

- **All Nodes Addresses**
  - FF01:0:0:0:0:0:0:1 (interface-local)
  - FF02:0:0:0:0:0:0:1 (link-local)

- **All Routers Addresses**
  - FF01:0:0:0:0:0:0:2 (interface-local)
  - FF02:0:0:0:0:0:0:2 (link-local)
  - FF05:0:0:0:0:0:0:2 (site-local)

Solicited-Node Address

SAPIENZA UNIVERSITÀ DI ROMA

- Solicited-Node Address:

FF02:0:0:0:0:1:FFXX:XXXX

- Solicited-Node multicast address are computed as a function of a node's unicast addresses
- It is formed by taking the low-order 24 bits of an address and appending those bits to the prefix FF02:0:0:0:0:1:FF00::/104

4037::01:800:200E:8C6C -> FF02::1:FF0E:8C6C

- A node is required to compute and join the associated Solicited-Node multicast addresses for all its unicast addresses

IPv6 Subnetting - Overview

SAPIENZA UNIVERSITÀ DI ROMA

- Subnetting with IPv6 is not drastically different than subnetting with IPv4, we just need to keep a few things in mind:
  - Each character in an IPv6 address represents 4 bits
    - Since 0xF is 1111 in binary, it's easy to fall back into an IPv4 habit and forget that 0x11 is actually 0001 0001 in binary
  - Each IPv6 set represent 16 bits (4 characters at 4 bits each)
    - Keeping this in mind can make breaking up subnets a bit easier
  - Once it's in binary nothing changes!

https://community.cisco.com/t5/networking-documents/ipv6-subnetting-overview-and-case-study/ta-p/3125702

Setting the Ground Rules

SAPIENZA
UNIVERSITÀ DI ROMA

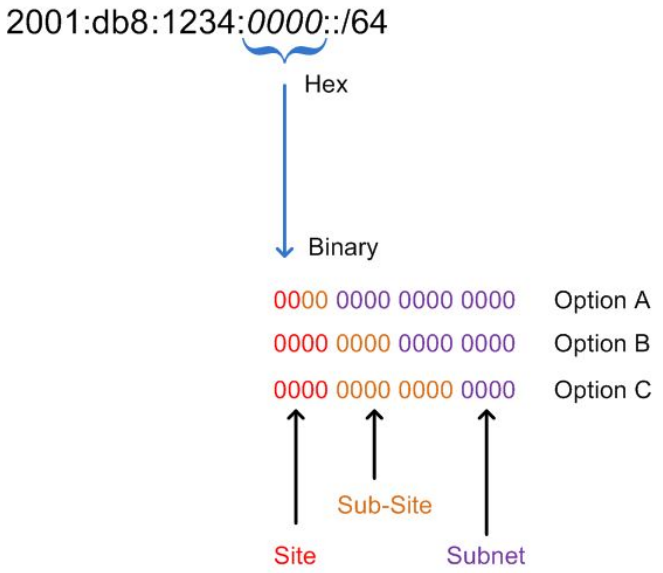
- The leading practice is to receive at least a /48 prefix from an ISP
  - This leaves you with 2^{80} bits to manipulate (128 bit address - 48 bits that can't be changed = 80 bits to use)
    - More bits than the entire IPv4 address space!
- According to RFC 4291 the current recommended smallest prefix is a /64
- With so many addresses in IPv6 there isn't the same need for address conservation as there is in IPv4
  - You can assign a /64 to a point-to-point link and not feel guilty
- This gives us one block of hex digits, or 16-bits, to use for subnetting
  - One block might not sound like much, but 16-bits is half of the entire IPv4 address space

Defining the Site ID

SAPIENZA UNIVERSITÀ DI ROMA

- In order to allow for proper route aggregation and summarization you should define Site IDs that you can use at each location (an office, data center or geographic region)
- This is where we need to define at least a Site ID, and possibly a sub-Site ID

Example



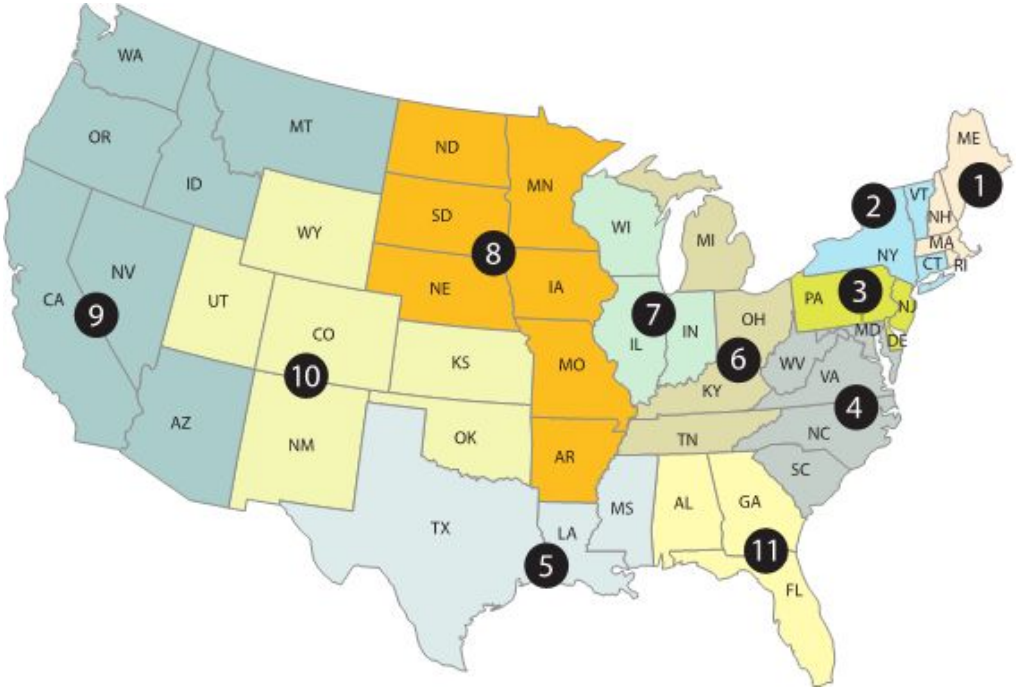
IPv6 Subnetting Case Study

SAPIENZA UNIVERSITÀ DI ROMA

- Let’s consider a mid-sized company with offices and data centers across the United States
- **Assigned block:** 2001:db8:abcd::/48
  - We need to allocate this across our enterprise
- We have branches in most states
  - let’s use Option B, giving us 16 sites, 16 sub-sites and 256 subnets per site
    - "site" == geographic region of the country
    - “sub-site” == a city within the geographic region

IPv6 Subnetting Case Study

SAPIENZA
UNIVERSITÀ DI ROMA



The sites that we are rolling IPv6 to are in:

- San Francisco (Site 9)
- Seattle (Site 9)
- Omaha (Site 8)
- Newark (Site 3)
- New York City (Site 2)
- Boston (Site 1)

Site Addresses

SAPIENZA
UNIVERSITÀ DI ROMA

- Site 0 - 2001:db8:abcd:0000::/52 (for future use)
- Site 1 - 2001:db8:abcd:1000::/52
- Site 2 - 2001:db8:abcd:2000::/52
- Site 3 - 2001:db8:abcd:3000::/52
- ...
- Site 8 - 2001:db8:abcd:8000::/52
- Site 9 - 2001:db8:abcd:9000::/52
- Site 10 - 2001:db8:abcd:a000::/52 (for future use)
- Site 11 - 2001:db8:abcd:b000::/52 (for future use)
- Site 12 - 2001:db8:abcd:c000::/52 (for future use)

Sub-site Addresses

SAPIENZA
UNIVERSITÀ DI ROMA

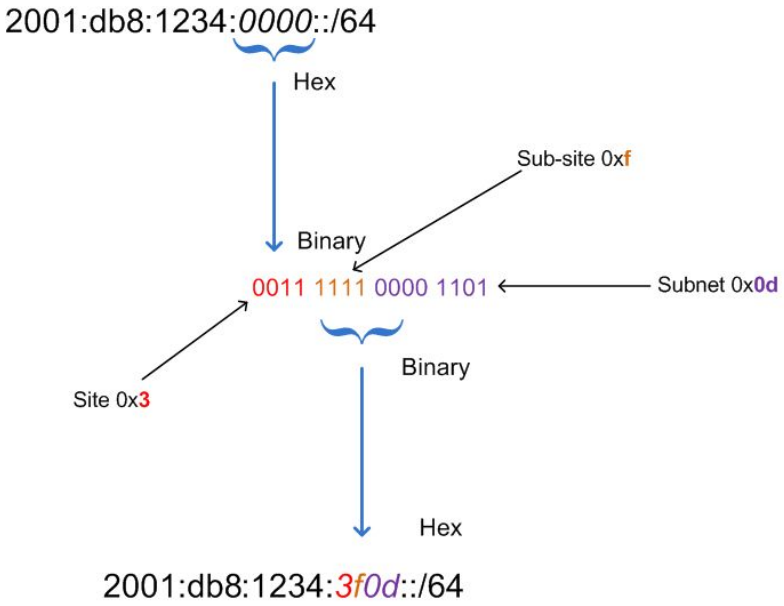
- site 1
  - Boston - 2001:db8:abcd:1100::/56
  - others for future use
- site 2
  - New York City - 2001:db8:abcd:2000::/56
- site 3
  - Newark - 2001:db8:abcd:3f00::/56
- site 8
  - Omaha - 2001:db8:abcd:8000::/56
- site 9
  - San Francisco - 2001:db8:abcd:9100::/56
  - Seattle - 2001:db8:abcd:9200::/56

Subnet Addresses

SAPIENZA UNIVERSITÀ DI ROMA

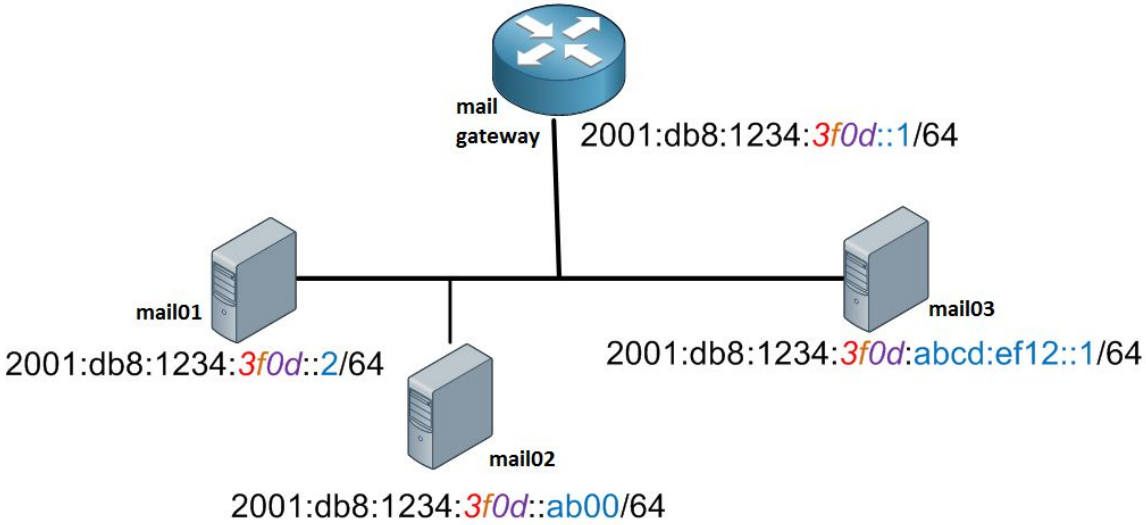
We will use our Newark site as an example

- Firewall Outside: 2001:db8:abcd:3f00::/64
- Webservers: 2001:db8:abcd:3f01::/64
- Database Servers: 2001:db8:abcd:3f02::/64
- ....
- Mail Servers: 2001:db8:abcd:3f0d::/64
- ....
- Management: 2001:db8:abcd:3fee::/64
- Loopbacks: 2001:db8:abcd:3fff::/64



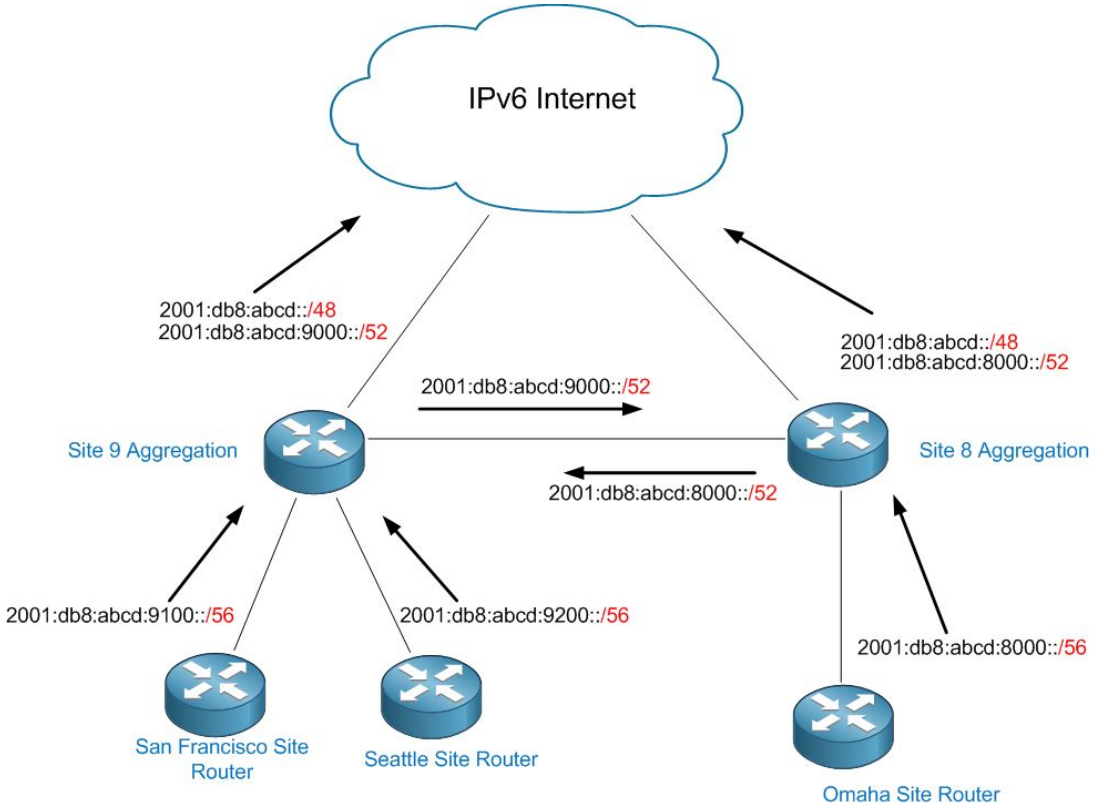
Assigning Addresses to hosts

SAPIENZA
UNIVERSITÀ DI ROMA



Mail Servers: 2001:db8:1234:3f0d::/64

Routing


IPv6 Internet
2001:db8:abcd::/48
2001:db8:abcd:9000::/52
2001:db8:abcd::/48
2001:db8:abcd:8000::/52
2001:db8:abcd:9000::/52
Site 9 Aggregation
Site 8 Aggregation
2001:db8:abcd:8000::/52
2001:db8:abcd:9100::/56
2001:db8:abcd:9200::/56
2001:db8:abcd:8000::/56
San Francisco Site Router
Seattle Site Router
Omaha Site Router

Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
  - Security consideration
- IPv6 Addressing
  - type of addresses
  - unicast
  - multicast
  - subnetting
- Neighbor Discovery
  - basics
  - messages
  - host conceptual procedures and data structures
  - Router Discovery
  - Neighbor Discovery
  - Security Threats
- Host Autoconfiguration
  - SLAAC
  - DHCPv6
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

Basics

SAPIENZA
UNIVERSITÀ DI ROMA

- Nodes (hosts and routers) use Neighbor Discovery to
  - **determine the link-layer addresses for neighbors** known to reside on attached links
  - **purge cached values** that become invalid
  - **find neighboring routers** that are willing to forward packets on their behalf
  - actively **keep track of which neighbors are reachable** and which are not
  - detect changed link-layer addresses
- The procedure is based on the following IPv6 addresses:
  - all-nodes multicast address
  - all-nodes multicast address
  - solicited-node multicast address
  - link-local address
  - unspecified address

Messages

SAPIENZA UNIVERSITÀ DI ROMA

- Router Solicitation
  - Hosts send Router Solicitations in order to prompt routers to generate Router Advertisements quickly

src: link-local or unspecified	IP
dst: all-router multicast	IP
opt: source link-layer address	ICMP

- Router Advertisement
  - Routers send out Router Advertisement messages periodically, or in response to Router Solicitations

src: router interface link-local address	IP
dst: host that sent RS/ all-nodes	IP
opt: router link-layer address, MTU, prefix	ICMP

Messages

SAPIENZA UNIVERSITÀ DI ROMA

- Neighbor Solicitation
  - Nodes send Neighbor Solicitations to request the link-layer address of a target node while also providing their own link-layer address to the target

src: link-local or unspecified	IP
dst: solicited-node or link-layer of target	
target: IP address of the target of the solicitation	ICMP
opt: source link-layer address	

- Neighbor Advertisement
  - A node sends Neighbor Advertisements in response to Neighbor Solicitations and sends unsolicited Neighbor Advertisements in order to (unreliably) propagate new information quickly

SA: an address of the sending interface	IP
DA: host that sent NS/ all-nodes	
target: Target Address field in the NS	ICMP
opt: target link-layer address	

Messages

SAPIENZA UNIVERSITÀ DI ROMA

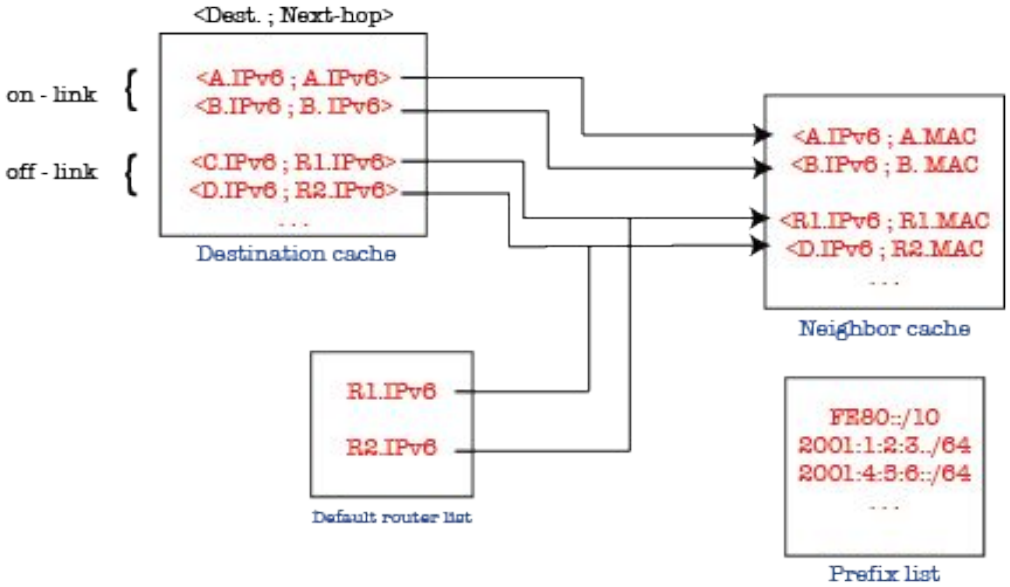
- Redirect
  - Routers send Redirect packets to inform a host of a better first-hop node on the path to a destination

Src addr: router interface link-local	IP
Dst addr: Source Address of the packet that triggered the redirect	
Target addr: An IP address that is a better first hop to use for the ICMP DA	ICMP
Dst Addr: IP address of the destination that is redirected to the target	
opt: link-layer address for the target	

Host (conceptual) Data Structures

SAPIENZA UNIVERSITÀ DI ROMA

- Hosts need to maintain the following (conceptual) data structures for each interface:

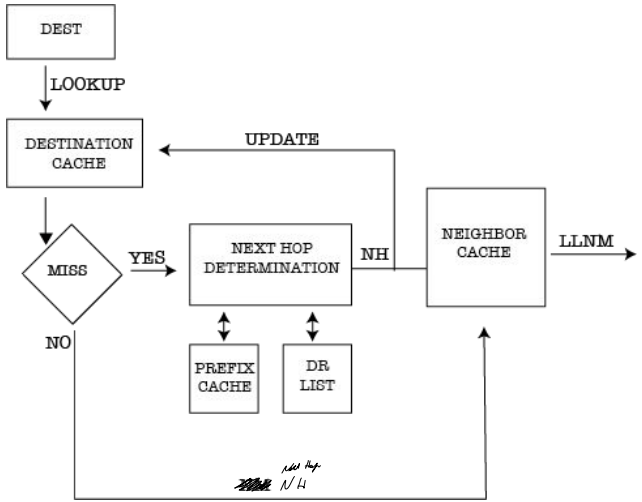


(conceptual) Sending Algorithm

SAPIENZA
UNIVERSITÀ DI ROMA

- NH Determination (NHD)
  - The sender performs a longest prefix match against the Prefix List to determine whether the packet's destination is on- or off-link
    - If the destination is on-link, the next-hop address is the same as the packet's destination address
    - Otherwise, the sender selects a router from the Default Router List
- For efficiency reasons, NHD is not performed on every packet that is sent
  - the results of NHD computations are saved in the Destination Cache

(conceptual) Sending Algorithm



- Each time a Neighbor Cache entry is accessed, the sender checks Neighbor Unreachability Detection (NUD) related information according to the NUD algorithm
  - Neighbor Solicitation to verify that the neighbor is still reachable

Router Discovery

SAPIENZA
UNIVERSITÀ DI ROMA

- Router Discovery is used to locate neighboring routers

```
Router Solicitation Message ->
Src: fe80::c072:7a5f:c1b5:24d1
Dst: ff02::2 (all routers multicast)
ICMPv6 Type 133 (RS)
Option:
  Src Link Layer Addr (my MAC addr)
```

(Routers also periodically send out unsolicited router advertisements.)

```
<- Router Advertisement Message
Src: router's link local addr
Dst: ff02::1 (all nodes or solicitor)
ICMPv6 Type 134 (RA)
Flags (M=0, O=0, pref=0)
Router Lifetime: 1800
Reachable time: 0
Retrans time: 0
Options:
  Src Link Layer Addr (my Mac)
  MTU: 1500
  Prefix Info
    prefix: 2001:db8:ab:cd::/64
    valid life: 2592000
    preferred lifetime: 604800
```

Neighbor Discovery

SAPIENZA
UNIVERSITÀ DI ROMA

Neighbor Solicitation Message ->
Src: A's IPv6 address
Dst: Solicited-node multicast of B
ICMPv6 Type 135 (NS)
Target: B's IPv6 address
Options:
  Src Link Layer Addr (A's MAC addr)

(Summary: A is asking: what is the link layer address associated with B's IPv6 address?)

<- Neighbor Advertisement Message
Src: B's IPv6 address
Dst: A's IPv6 address
ICMPv6 Type 135 (NA)
Target: B's IPv6 address
Options:
  Src Link Layer Addr (B's MAC addr)

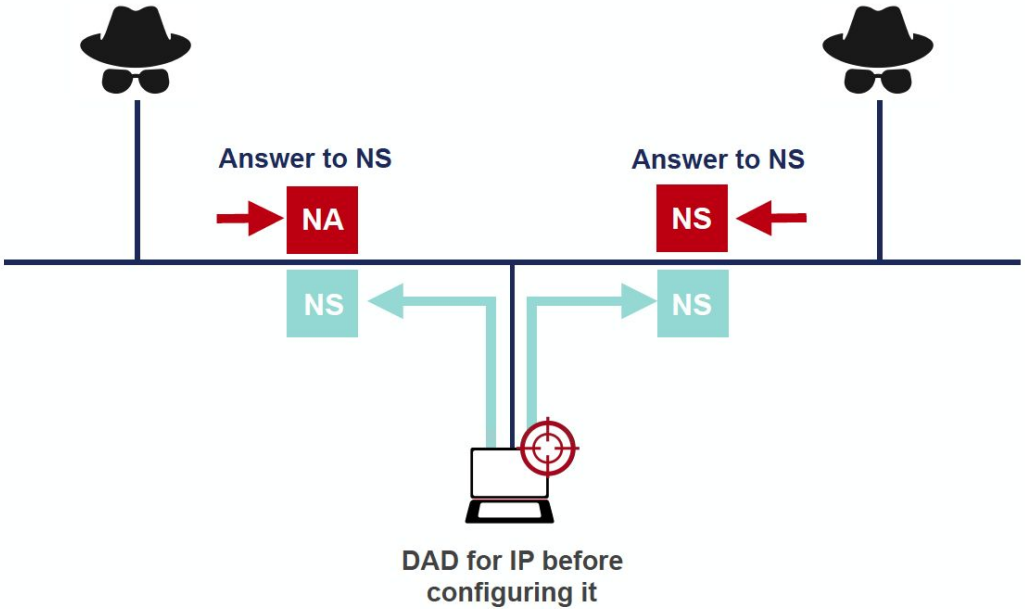
ND Security Threats

SAPIENZA
UNIVERSITÀ DI ROMA

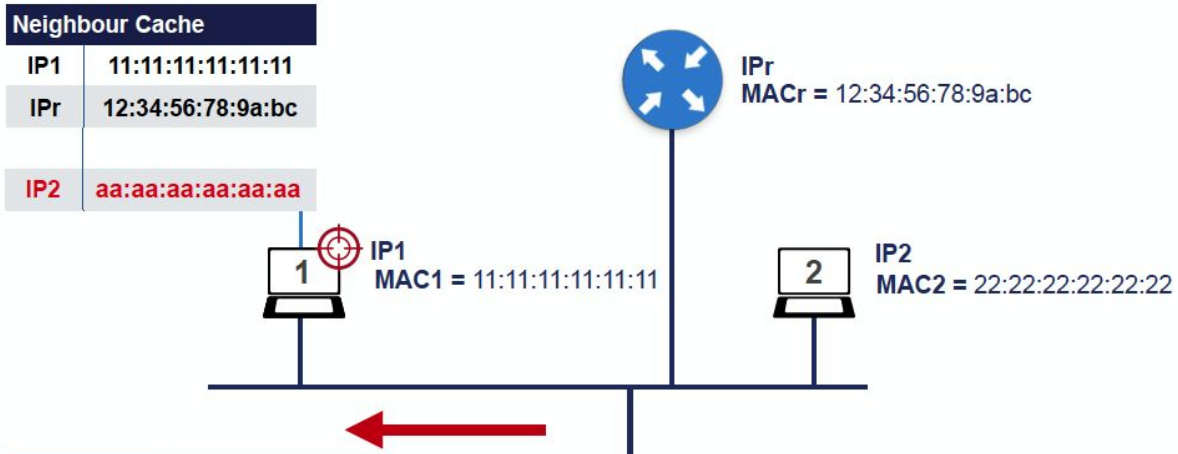
- Neighbor Discovery is subject to attacks that cause IP packets to flow to unexpected places
- The main vulnerabilities of the protocol fall under three categories:
  - Denial-of-Service (DoS) attacks
  - Address spoofing attacks
  - Router spoofing attacks

DAD DoS Attack

SAPIENZA
UNIVERSITÀ DI ROMA

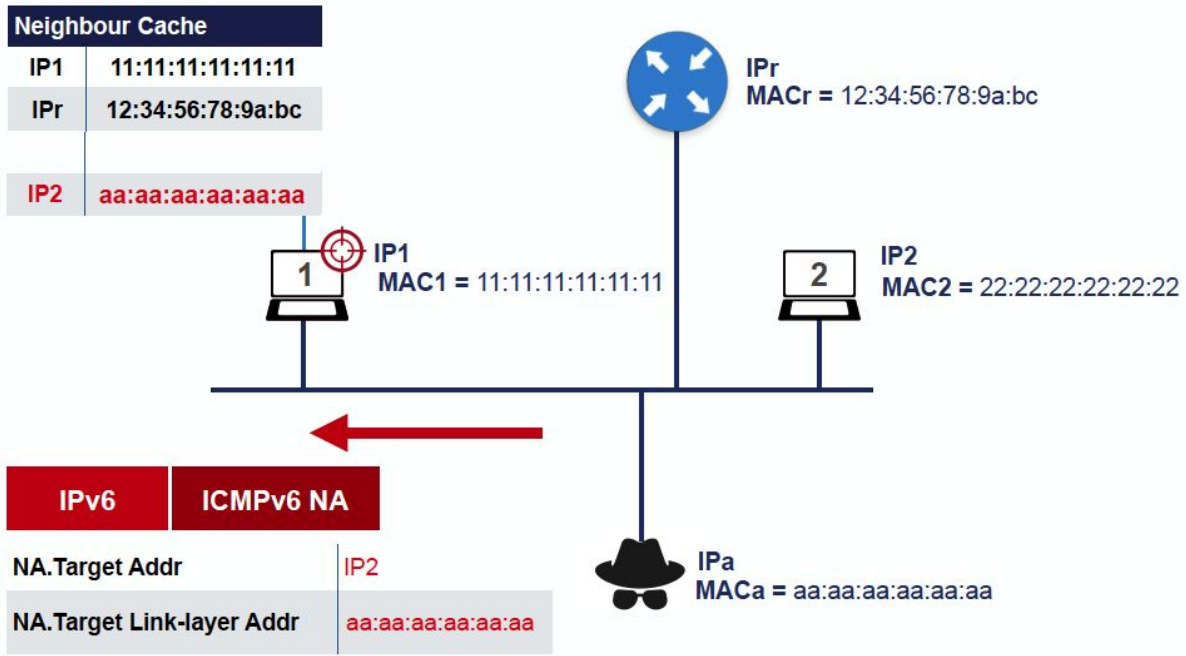


NS Spoofing



IPv6	ICMPv6 NS
IPv6.Source IPv6	IP2
IPv6.Destination IPv6	IP1
NS.Target Addr	IP1
NS.Src Link-layer Addr	aa:aa:aa:aa:aa:aa

Unsolicited NA



Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
  - Security consideration
- IPv6 Addressing
  - type of addresses
  - unicast
  - multicast
  - subnetting
- Neighbor Discovery
  - basics
  - messages
  - host conceptual procedures and data structures
  - Router Discovery
  - Neighbor Discovery
  - Security Threats
- Host Autoconfiguration
  - SLAAC
  - DHCPv6
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

Introduction

SAPIENZA
UNIVERSITÀ DI ROMA

- There are three way a host can get an IPv6 address:
  - **manual configuration**
  - **stateless autoconfiguration**
    - used when a site is not particularly concerned with the exact addresses hosts use (SLAAC)
  - **stateful autoconfiguration** (with DHCPv6)
    - used when a site requires tighter control over exact address assignments

Available Methods

SAPIENZA
UNIVERSITÀ DI ROMA

RA Address Allocation Method	A Flag (SLAAC) Default: **on**	O Flag (Stateless DHCPv6) Default: **off**	M Flag (Stateful DHCPv6) Default: **off**
SLAAC (default)	1 (on)	0 (off)	0 (off)
SLAAC and stateless DHCPv6	1 (on)	1 (on)	0 (off)
Stateful DHCPv6	0 (off)	N/A	1 (on)

Stateless Address Autoconfiguration (SLAAC)

SAPIENZA UNIVERSITÀ DI ROMA

- The IPv6 StateLess Address AutoConfiguration mechanism requires no manual configuration of hosts
- A host generates its own addresses using a combination of locally available information and information advertised by routers
  - Routers advertise prefixes that identify the subnet(s) associated with a link
  - hosts generate an "interface identifier" that uniquely identifies an interface on a subnet
- Use of ICMPv6 messages
  - Router Advertisement
  - Router Solicitation

SLAAC terminology

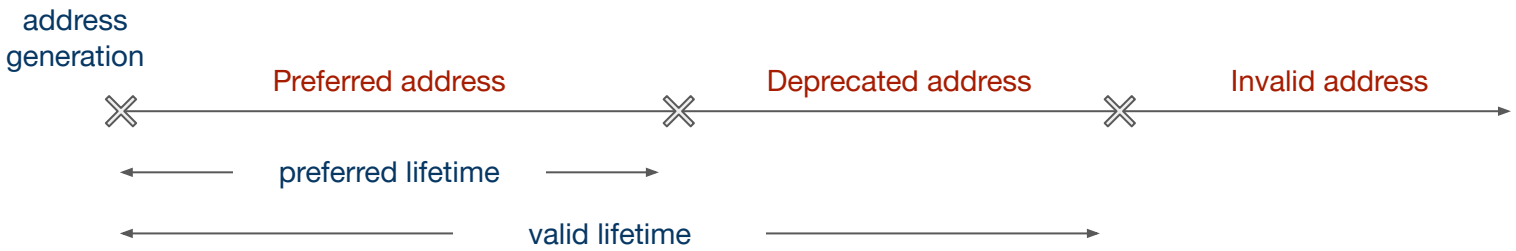
SAPIENZA
UNIVERSITÀ DI ROMA

- **tentative address**
  - an address whose uniqueness on a link is being verified, prior to its assignment to an interface
- **preferred address**
  - an address assigned to an interface whose use by upper-layer protocols is unrestricted
- **deprecated address**
  - an address assigned to an interface whose use is discouraged, but not forbidden
- **valid address**
  - a preferred or deprecated address
- **invalid address**
  - a valid address becomes invalid when its valid lifetime expires

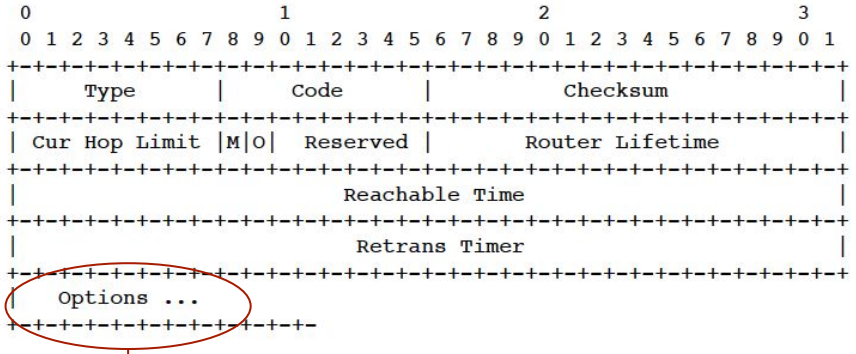
Lifetime of a SLAAC address

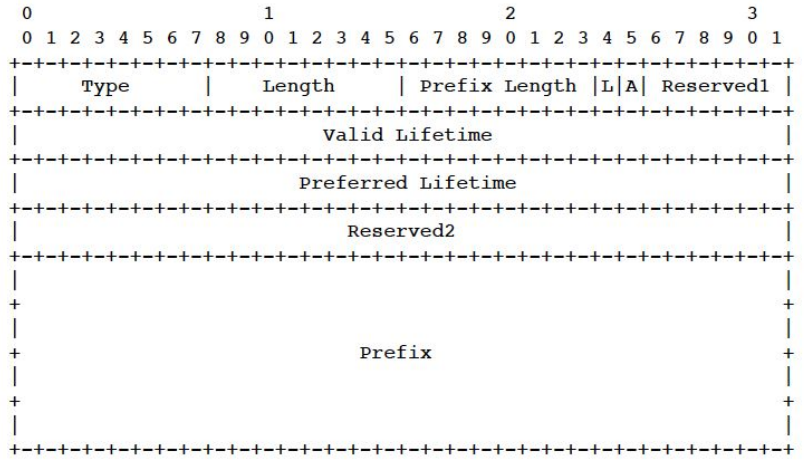
SAPIENZA
UNIVERSITÀ DI ROMA

- The lifecycle of a SLAAC configured address depends on the lifetime of the prefix used to generate it
- Prefixes have two lifetime:
  - preferred
  - valid



SLAAC overview





SLAAC overview

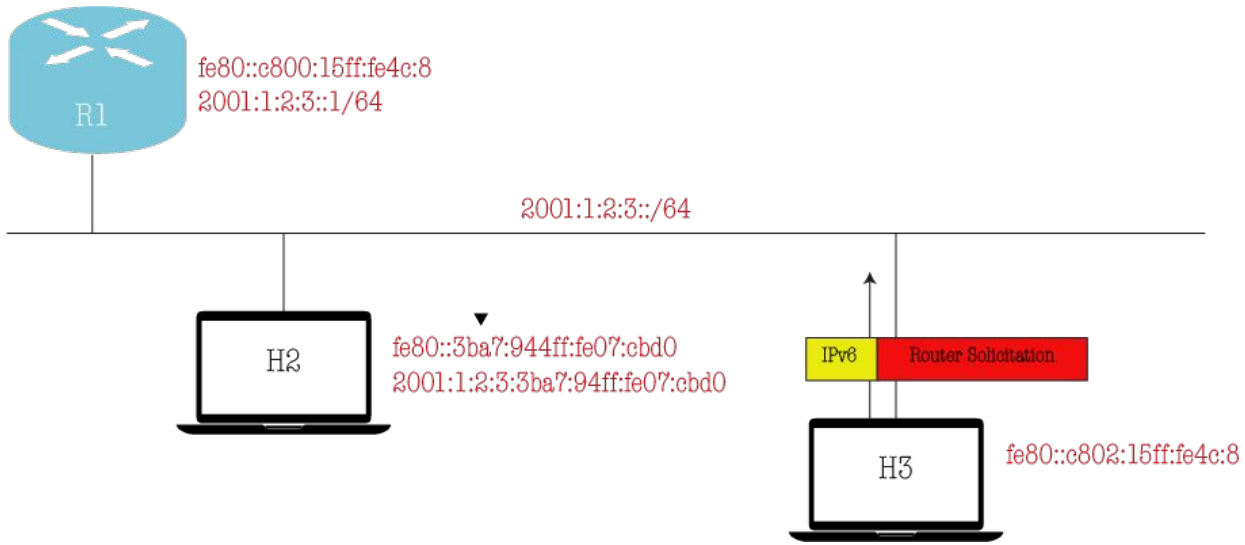
SAPIENZA UNIVERSITÀ DI ROMA

- host H3 connects to the subnet and immediately generates an IPv6 Link Local address
  - it uses Duplicate Address Detection to verify the uniqueness of the generated address

SLAAC overview

SAPIENZA
UNIVERSITÀ DI ROMA

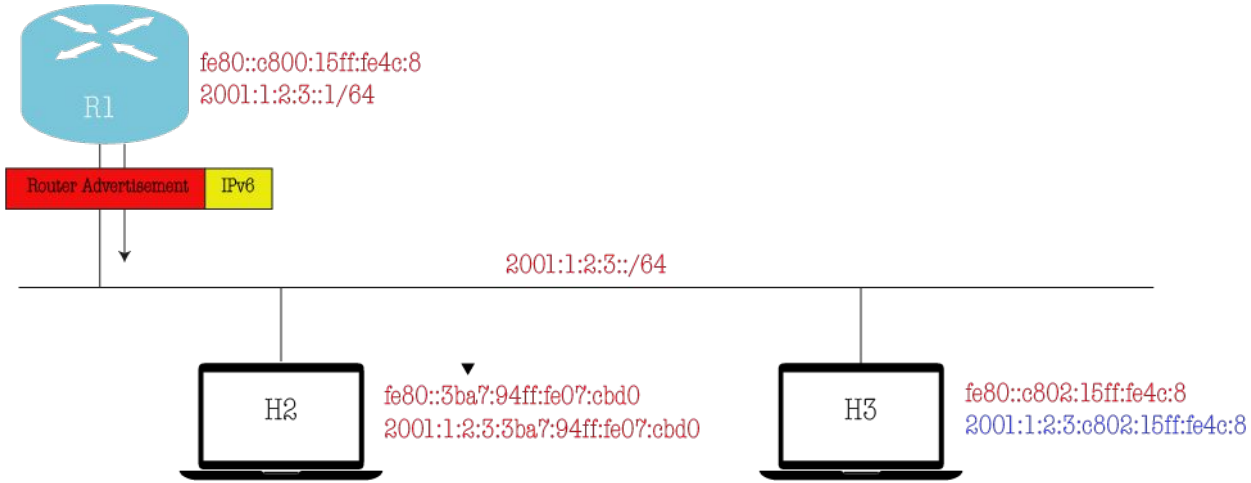
- host H3 sends one (or more) router solicitation to discover routers in the subnet



SLAAC overview

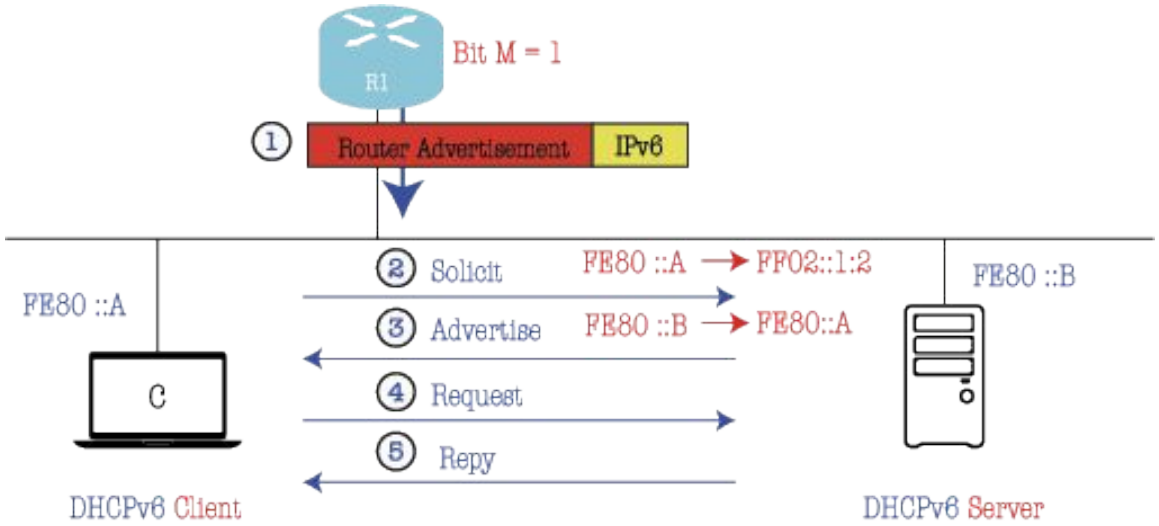
SAPIENZA
UNIVERSITÀ DI ROMA

- Router R1 sends a router advertisement containing (as an option) the subnet prefix
- with this information, the host H3 generates a global unicast IPv6 address using the subnet prefix information



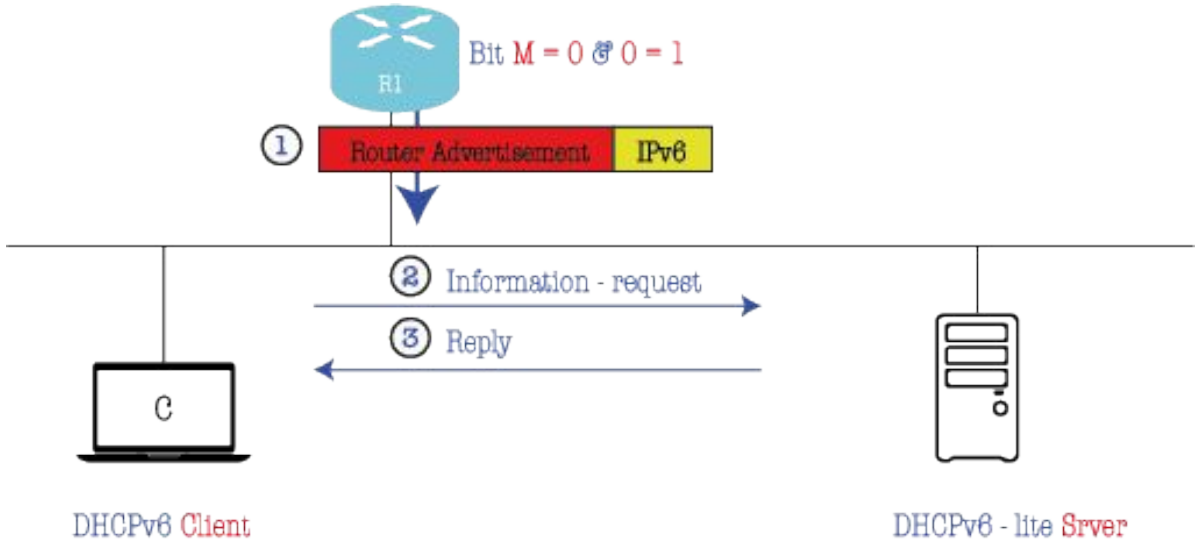
Statefull DHCPv6

SAPIENZA
UNIVERSITÀ DI ROMA



Stateless DHCPv6

SAPIENZA
UNIVERSITÀ DI ROMA



Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
  - Security consideration
- IPv6 Addressing
  - type of addresses
  - unicast
  - multicast
  - subnetting
- Neighbor Discovery
  - basics
  - messages
  - host conceptual procedures and data structures
  - Router Discovery
  - Neighbor Discovery
  - Security Threats
- Host Autoconfiguration
  - SLAAC
  - DHCPv6
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

Overview

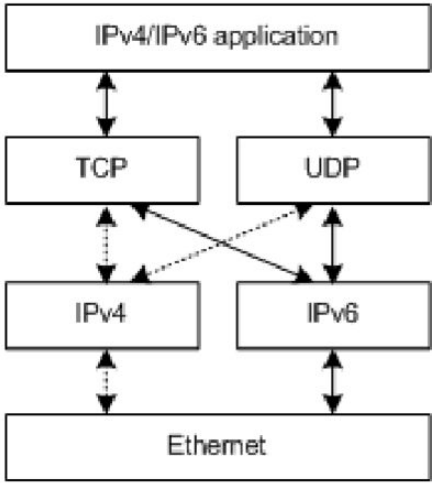
SAPIENZA
UNIVERSITÀ DI ROMA

- The key to a successful IPv6 transition is compatibility with the large installed base of IPv4 hosts and routers
- Two mechanism to manage transition
  - **dual IP layer** (also known as dual stack)
    - complete support for both IPv4 and IPv6 in hosts and routers
  - **tunneling** of IPv6 over IPv4
    - establishing point-to-point tunnels by encapsulating IPv6 packets within IPv4

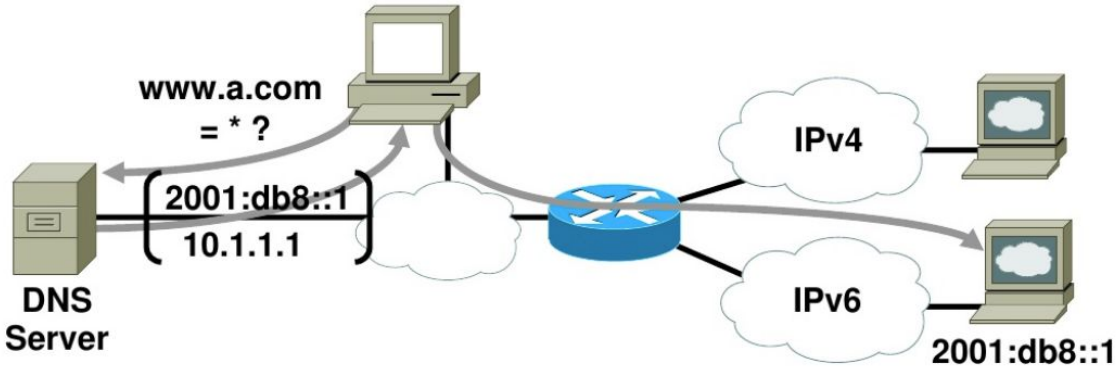
Dual IP Layer Operation

SAPIENZA UNIVERSITÀ DI ROMA

- The most straightforward way for IPv6 nodes to remain compatible with IPv4-only nodes is by providing a complete IPv4 implementation (IPv6/IPv4 nodes)
  - Address Configuration
    - nodes may be configured with both IPv4 and IPv6 addresses (e.g., DHCPv4 + SLAAC)
  - DNS
    - A new resource record type named "AAAA" has been defined for IPv6 addresses
    - IPv6/IPv4 nodes must provide resolver libraries capable of dealing with IPv4 "A" records as well as IPv6 "AAAA" records



Dual IP Layer Operation



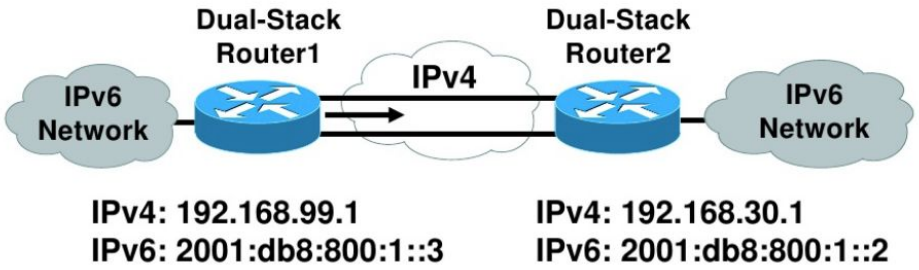
Configured Tunneling Mechanisms

SAPIENZA
UNIVERSITÀ DI ROMA

- Tunneling provides a way to utilize an existing IPv4 routing infrastructure to carry IPv6 traffic
  - while the IPv6 infrastructure is being deployed, the existing IPv4 routing infrastructure can be used to carry IPv6 traffic
- IPv6 datagrams forwarded over regions of IPv4 routing topology by encapsulating them within IPv4 packets
- Tunneling can be used in a variety of ways
  - Router-to-Router
  - Host-to-Router
  - Host-to-Host
  - Router-to-Host

Manually Configured IPv6 over IPv4 Tunnel

SAPIENZA UNIVERSITÀ DI ROMA




```
router1#

interface Tunnel0
 ipv6 enable
 ipv6 address 2001:db8:c18:1::3/127
 tunnel source 192.168.99.1
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip
```

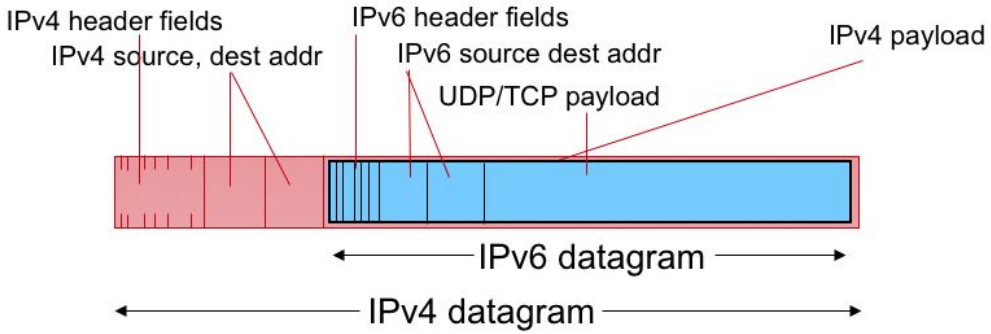
```
router2#

interface Tunnel0
 ipv6 enable
 ipv6 address 2001:db8:c18:1::2/127
 tunnel source 192.168.30.1
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

Configured Tunneling Mechanisms

SAPIENZA UNIVERSITÀ DI ROMA

- The underlying mechanisms for tunneling are
  - The **entry node of the tunnel**
    - creates an encapsulating IPv4 header and transmits the encapsulated packet
  - The **exit node of the tunnel**
    - receives the encapsulated packet,
    - reassembles the packet if needed,
    - removes the IPv4 header,
    - processes the received IPv6 packet
- The determination of which packets to tunnel is usually made by routing information on the encapsulator (e.g., via a routing table)

Encapsulation



- the encapsulator also has to handle some more complex issues:
  - Determine when to fragment and when to report an ICMPv6 "packet too big" error back to the source
  - How to reflect ICMPv4 errors from routers along the tunnel path back to the source as ICMPv6 errors.

Encapsulation