Networking Basics

06b - IPv4 Address Resolution and IPv6 Neighbor Discovery

Wolfgang Tremmel
academy@de-cix.net

DE CIX

Networking Basics

DE-CIX Academy

01 - Networks, Packets, and Protocols

02 - Ethernet, 02a - VLANs

03 - IP, 03a - Routing, 03b - Global routing,

04a - UDP, 04b - TCP, 04c - ICMP, 04d - Traceroute

05 - Uni-, Broad-, Multi-, and Anycast

06a - Domain Name System (DNS)

06b - IPv4 ARP and IPv6 ND

07a - Simple Mail Transfer Protocol (SMTP)

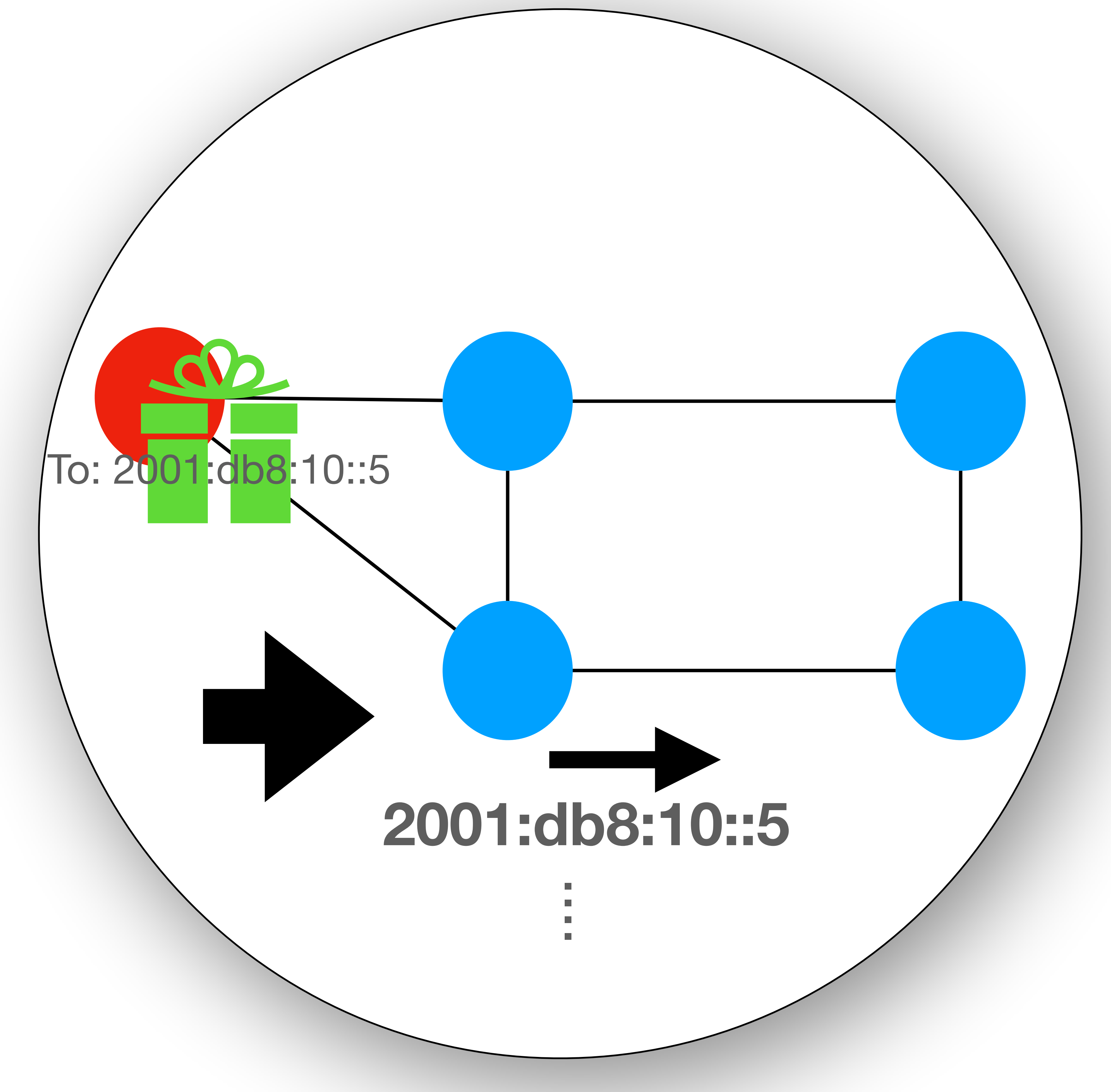
07b - Hypertext Transfer Protocol (HTTP)

The Problem

IP Routing

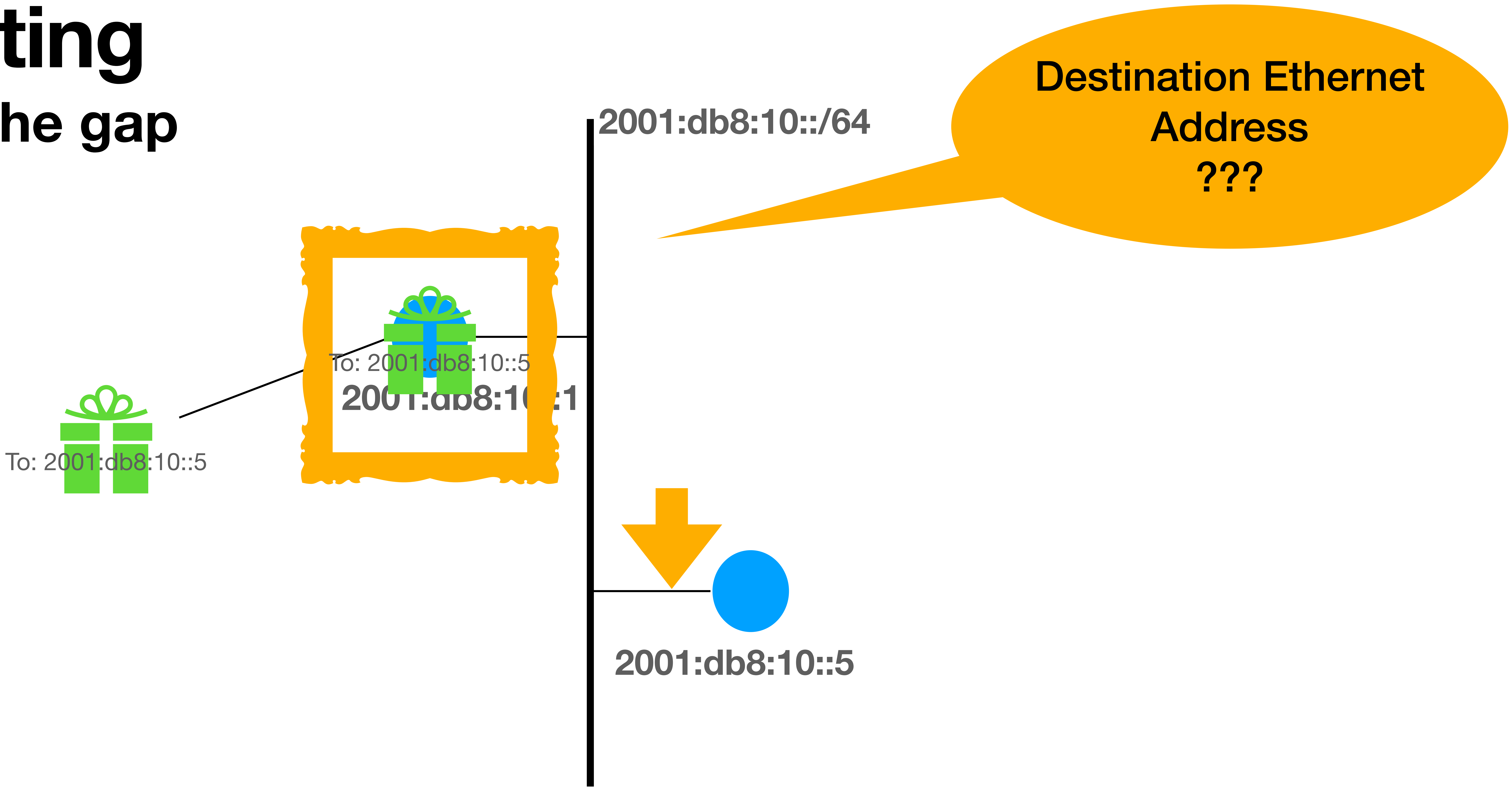
How a *router* works

- All IP packets have a *destination IP address*
- Depending on the destination IP address a *next hop* is chosen
- For this, each router has a large lookup-table
  - This is called the *routing table*
- It contains not single IP addresses, but prefixes
- This is quite the same for IPv4 and IPv6



IP Routing

Jumping the gap



The Problem

How to find the Ethernet address to a given IP address?

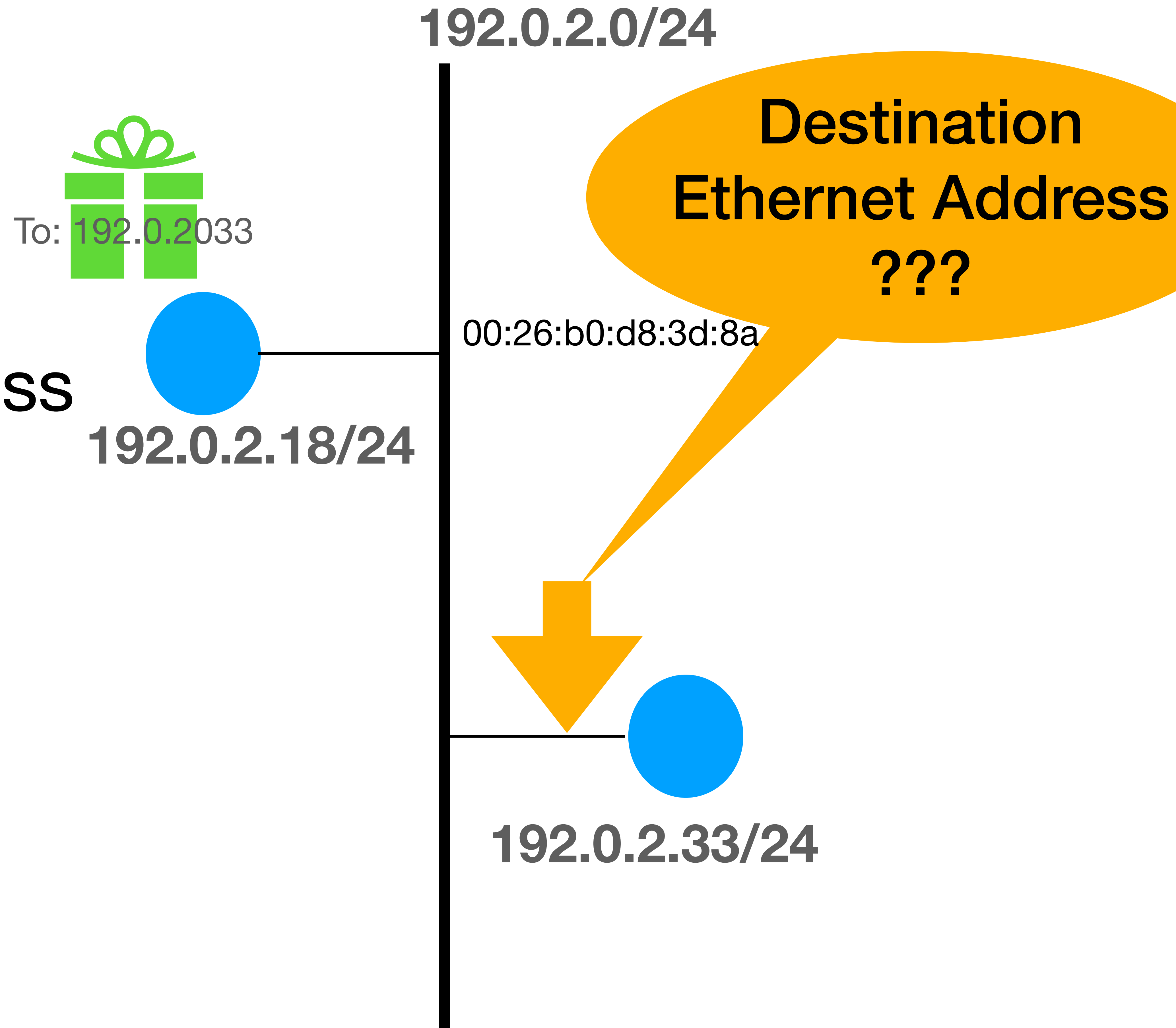
- We need a method to request an Ethernet address
- For a given IP target address
- On a LAN the router is directly connected
    - IPv4: ARP - Address Resolution Protocol
    - IPv6: ND - Neighbor Discovery
- Both request the Ethernet address for a given IP target address while providing the Ethernet address of the sender.

IPv4 - ARP
(Address Resolution Protocol)

Example

Requesting an Ethernet address for an IPv4 address

- My host has the IPv4 address 192.0.2.18/24
  - and the Ethernet address 00:26:b0:d8:3d:8a
- I want to send packets to 192.0.2.33
  - but I do not know the targets Ethernet address
- So how can I find it out?



ARP Request

Using the broadcast ability of Ethernet

- We need a protocol for this
  - The protocol for IPv4 is named ARP - Address Resolution Protocol
- ARP was designed for not only Ethernet
- In principle we *broadcast* a request to all systems on the Ethernet
- We send our own Ethernet and own and targets IPv4 address
- And ask for the targets Ethernet address

ARP Request

192.0.2.18 asking for 192.0.2.33

- ARP was designed not only for Ethernet, so we first have to fill out:
  - Hardware type "Ethernet"
  - Protocol type "IPv4" (0x800)
  - Hardware address length (48 bits = 6 octets)
  - Protocol address length (32 bits = 4 octets for IPv4)
- Operation = 1 (this is a request)

Octet	0	1
0	Hardware type (Ethernet = 1)	
2	Protocol type (IPv4 = 0x800)	
4	HW addr len (6)	Prot. addr len (4)
6	Operation (1 = request)	
8	Sender Ethernet address	
10		
12		
14	Sender IPv4 address	
16		
18	Target Ethernet address (this is what we want to know)	
20		
22		
24	Target IPv4 address	
26		

ARP Request

192.0.2.18 asking for 192.0.2.33

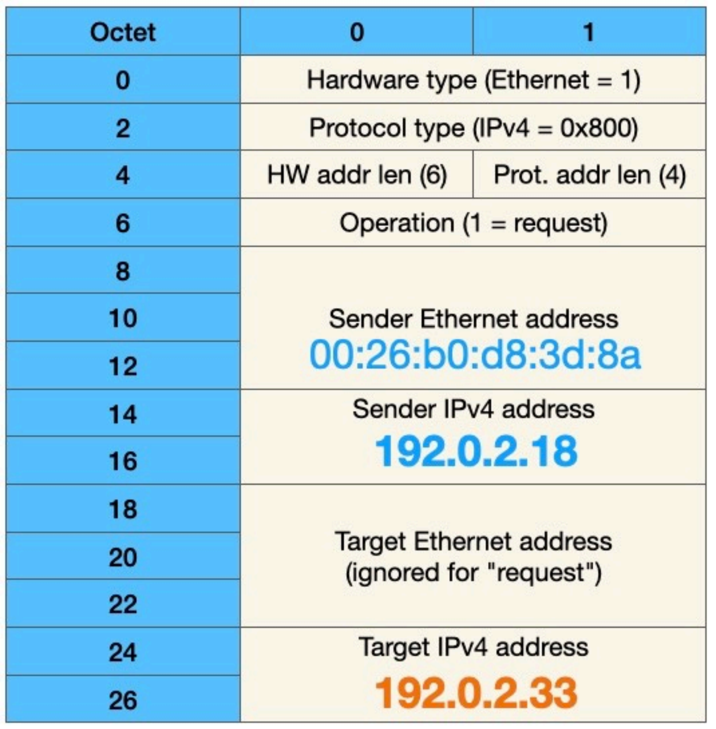
- Then we can add the request-specific values:
  - Requestor Ethernet address
  - Requestor IPv4 address
  - Target IPv4 address
- Target Ethernet address stays empty as this is what we want to know

Octet	0	1
0	Hardware type (Ethernet = 1)	
2	Protocol type (IPv4 = 0x800)	
4	HW addr len (6)	Prot. addr len (4)
6	Operation (1 = request)	
8	Sender Ethernet address 00:26:b0:d8:3d:8a	
10		
12		
14	Sender IPv4 address 192.0.2.18	
16		
18	Target Ethernet address (this is what we want to know)	
20		
22		
24	Target IPv4 address 192.0.2.33	
26		

Sending an ARP request

Broadcast!

- ARP request is put as payload into an Ethernet frame
- Destination MAC is ff:ff:ff:ff:ff:ff (broadcast address)
- Source MAC is senders Ethernet address
- Ethertype is 0x806 (ARP)

Preamble				SFD	Destination MAC Address	Source MAC Address	Ethertype	Payload	Checksum
10101010	10101010	10101010	10101010	10101011	ff:ff:ff:ff:ff:ff	00:26:b0:d8:3d:8a	0x806		32 Bits 4 Octets

Octet	0	1
0	Hardware type (Ethernet = 1)	
2	Protocol type (IPv4 = 0x800)	
4	HW addr len (6)	Prot. addr len (4)
6	Operation (1 = request)	
8	Sender Ethernet address 00:26:b0:d8:3d:8a	
10		
12		
14	Sender IPv4 address 192.0.2.18	
16		
18	Target Ethernet address (ignored for "request")	
20		
22		
24	Target IPv4 address 192.0.2.33	
26		

Broadcast

Disadvantages

- ARP requests are sent via Ethernet broadcast
- Every station on the LAN segment receives and needs to process this
  - Imagine a large IXP LAN with more than 1000 stations
- Each station checks if the target IPv4 address in the ARP request equals its own
  - This costs CPU cycles
  - If no, the ARP request is discarded
  - If yes, it sends an ARP reply

ARP Reply

192.0.2.33 answering to 192.0.2.18

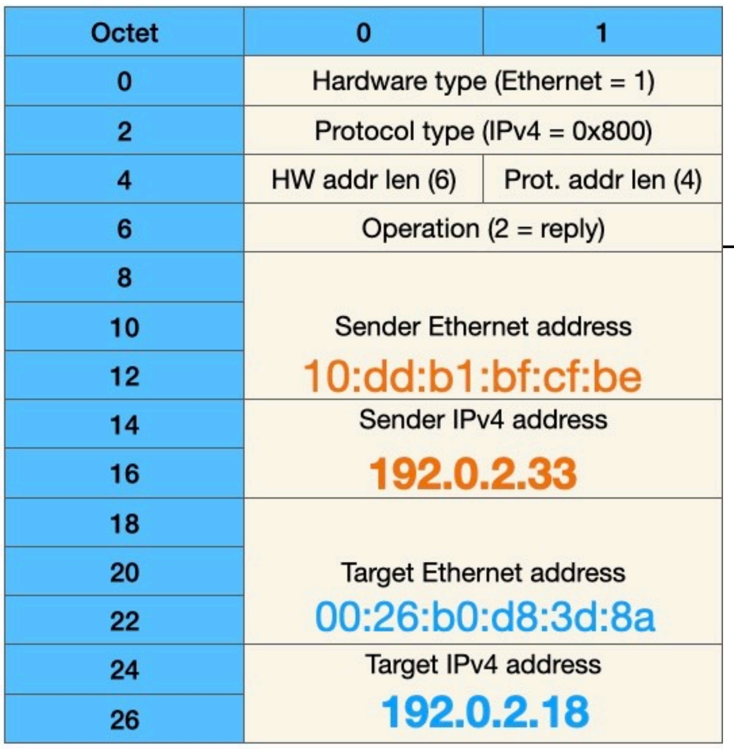
- ARP reply fills out:
  - Sender/orignal Target Ethernet address
  - Sender/original Target IPv4 address
  - Requestor Ethernet address
  - Requestor IPv4 address

Octet	0	1
0	Hardware type (Ethernet = 1)	
2	Protocol type (IPv4 = 0x800)	
4	HW addr len (6)	Prot. addr len (4)
6	Operation (2 = reply)	
8	Sender Ethernet address 10:dd:b1:bf:cf:be	
10		
12		
14	Sender IPv4 address **192.0.2.33**	
16		
18	Target Ethernet address 00:26:b0:d8:3d:8a	
20		
22		
24	Target IPv4 address **192.0.2.18**	
26		

Sending an ARP reply

Unicast to requestor

- ARP reply is put as payload into an Ethernet frame
- Destination MAC is requestors Ethernet Address
- Source MAC is senders Ethernet address
- Ethertype is 0x806 (ARP)

Preamble				SFD	Destination MAC Address	Source MAC Address	Ethertype	Payload	Checksum
10101010	10101010	10101010	10101010	10101011	00:26:b0:d8:3d:8a	10:dd:b1:bf:cf:be	0x806		32 Bits 4 Octets

Octet	0	1
0	Hardware type (Ethernet = 1)	
2	Protocol type (IPv4 = 0x800)	
4	HW addr len (6)	Prot. addr len (4)
6	Operation (2 = reply)	
8	Sender Ethernet address 10:dd:b1:bf:cf:be	
10		
12		
14	Sender IPv4 address 192.0.2.33	
16		
18	Target Ethernet address 00:26:b0:d8:3d:8a	
20		
22		
24	Target IPv4 address 192.0.2.18	
26		

ARP

Which layer?

- ARP works directly on Ethernet
- It is also not restricted to IPv4
- So I think it belongs to Layer 3
- Although not to the "Internet" layer

Layer	Name
5	Application

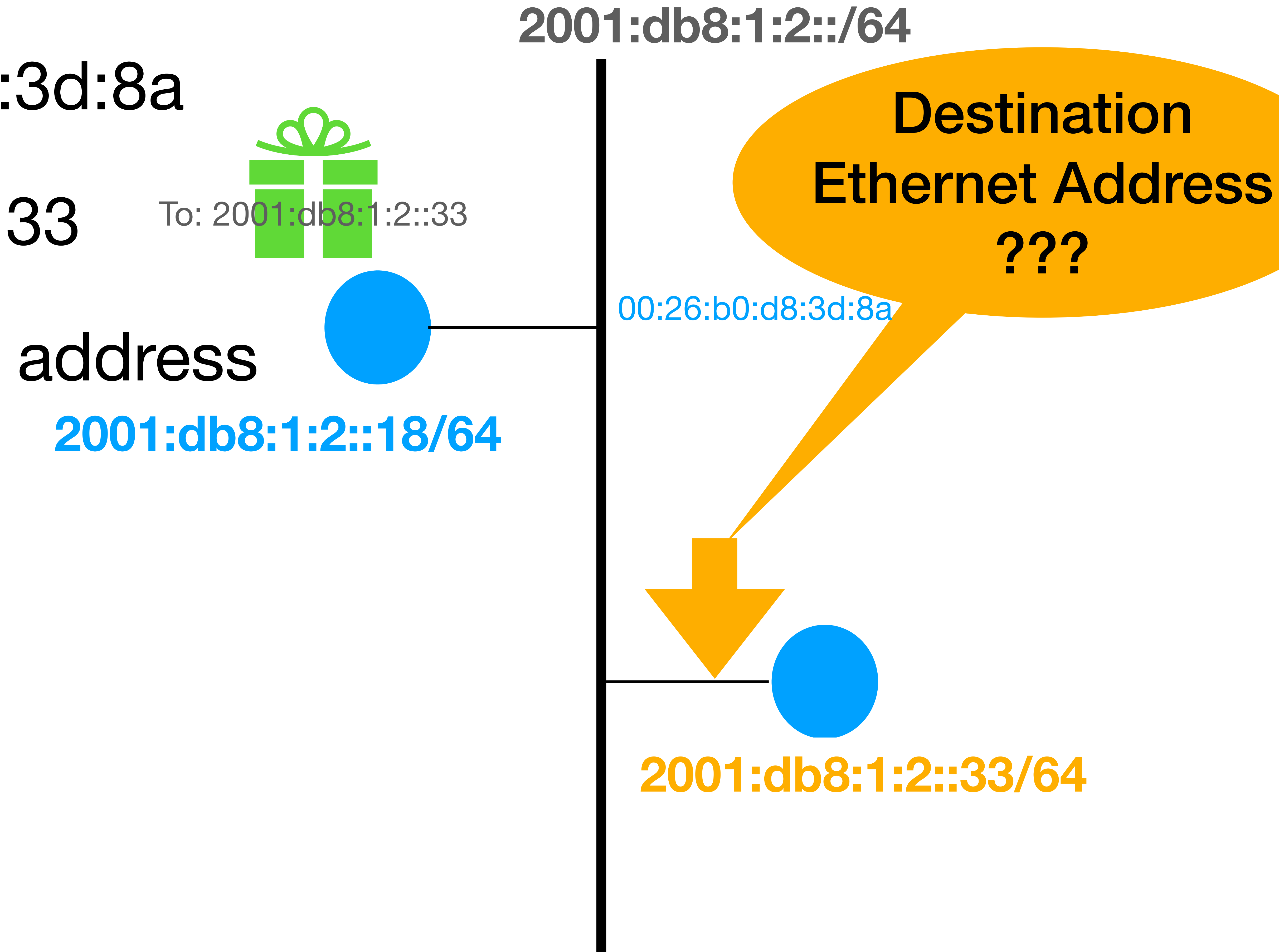
Layer	Name
3	ARP
2	Link
1	Physical

IPv6 - ND (Neighbor Discovery)

Example

Requesting an Ethernet address for an IPv6 address

- My host has the IPv6 address 2001:db8:1:2::18/64
  - and the Ethernet address 00:26:b0:d8:3d:8a
- I want to send packets to 2001:db8:1:2::33
  - but I do not know the targets Ethernet address
- So how can I find it out?



Neighbor Discovery

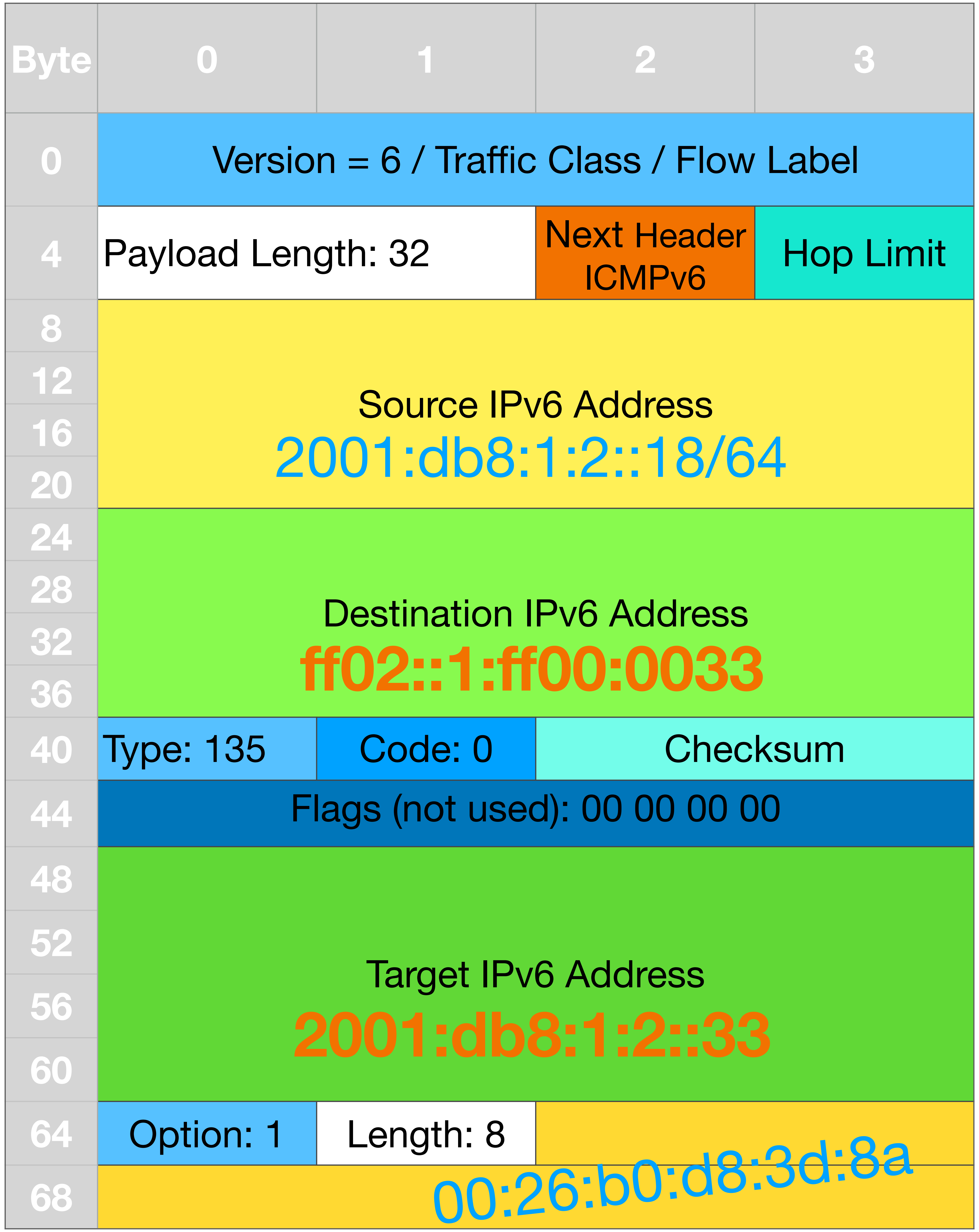
No broadcast in IPv6

- IPv6 no longer has broadcast ability, it uses multicast instead (send to a specific group of stations)
- Each IPv6 station, joins a number of multicast groups based on its IPv6 address(es)
- This is called the "solicited-node multicast address" (see RFC4291)
  - This multicast address is FF02:0:0:0:0:1:FFxx:xxxx
  - With xx:xx:xx being the last 6 bytes of the IPv6 address in question
  - There is also a mapping from IPv6 multicast addresses to Ethernet multicast addresses
  - So in IPv6 a broadcast to all is not necessary. The multicast address to send a neighbor discovery to can be calculated.

ICMPv6 Neighbor Solicitation

Request

- This is a standard IPv6 packet
  - With ICMPv6 as payload
- Source IPv6 address is requestor
- Destination IPv6 address is calculated multicast group
- Requested address goes into ICMPv6 part
- Source Ethernet address goes into an ICMPv6 option

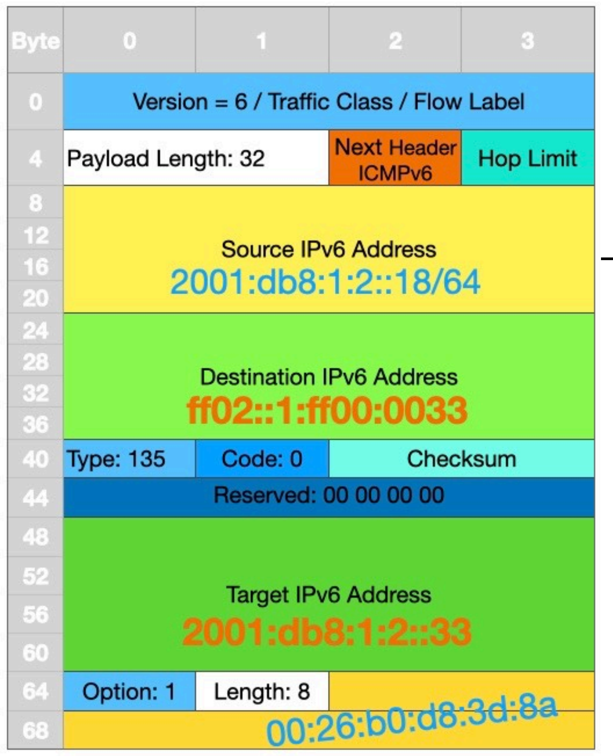


Byte	0	1	2	3
0	Version = 6 / Traffic Class / Flow Label			
4	Payload Length: 32		Next Header ICMPv6	Hop Limit
8	Source IPv6 Address 2001:db8:1:2::18/64			
12				
16				
20				
24	Destination IPv6 Address ff02::1:ff00:0033			
28				
32				
36				
40	Type: 135	Code: 0	Checksum	
44	Flags (not used): 00 00 00 00			
48	Target IPv6 Address 2001:db8:1:2::33			
52				
56				
60				
64	Option: 1	Length: 8	00:26:b0:d8:3d:8a	
68				

Sending an ICMPv6 multicast packet

Ethernet Multicast!

- ICMPv6 request is put as payload into an Ethernet frame
- Destination MAC is a multicast group calculated from the IPv6 address
  - 33:33: plus last part of IPv6 address
- Source MAC is senders Ethernet address
- Ethertype is 0x86dd (IPv6)

Preamble				SFD	Destination MAC Address	Source MAC Address	Ethertype	Payload	Checksum
10101010	10101010	10101010	10101010	10101011	33:33:ff:00:00:33	00:26:b0:d8:3d:8a	0x86dd		32 Bits 4 Octets

Byte	0	1	2	3
0	Version = 6 / Traffic Class / Flow Label			
4	Payload Length: 32		Next Header ICMPv6	Hop Limit
8	Source IPv6 Address 2001:db8:1:2::18/64			
12				
16				
20				
24	Destination IPv6 Address ff02::1:ff00:0033			
28				
32				
36				
40	Type: 135	Code: 0	Checksum	
44	Reserved: 00 00 00 00			
48	Target IPv6 Address 2001:db8:1:2::33			
52				
56				
60				
64	Option: 1	Length: 8	00:26:b0:d8:3d:8a	
68				

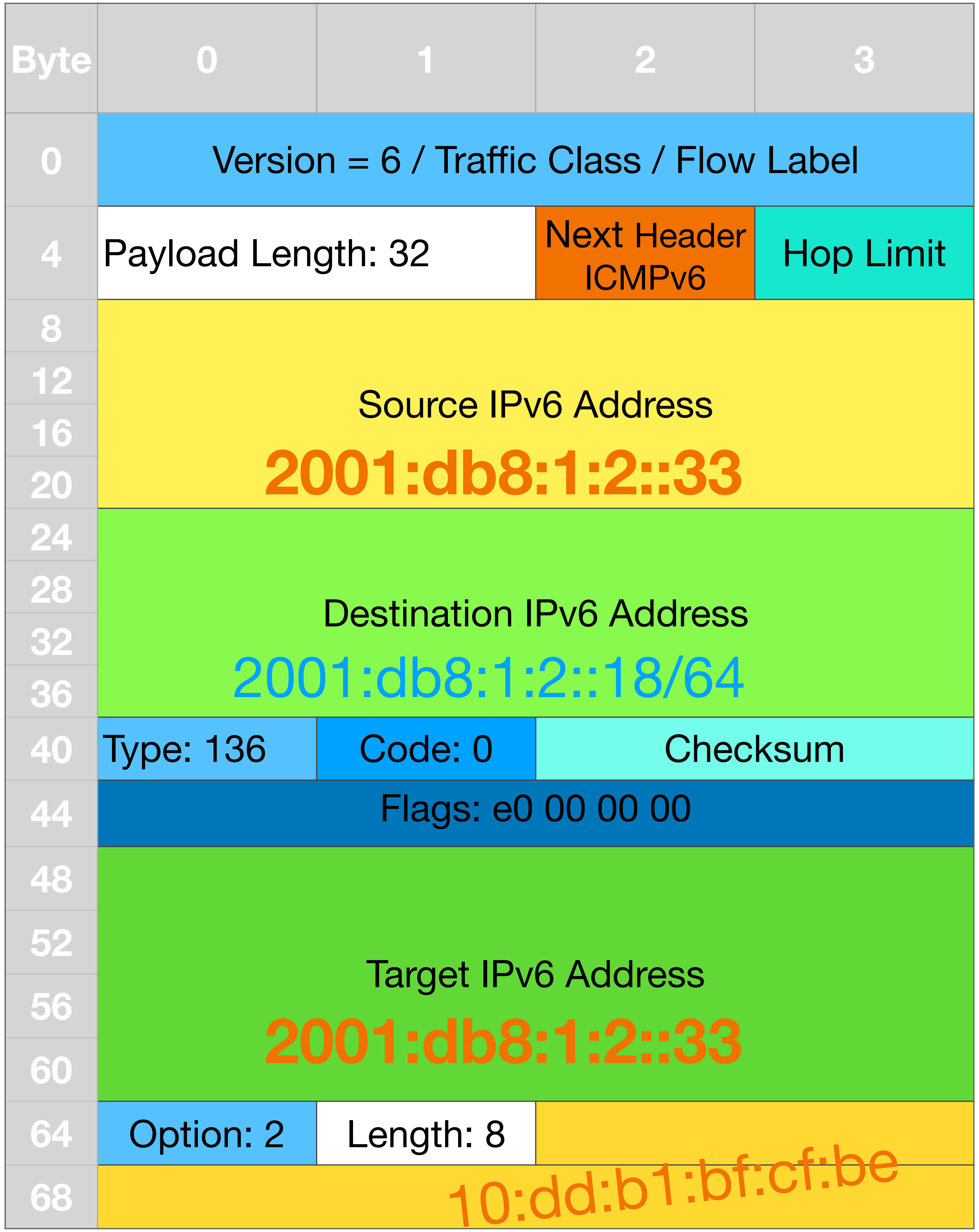
↑

ff02::1:ff00:0033

ICMPv6 Neighbor Solicitation

Reply

- Also a standard IPv6 packet
  - With ICMPv6 as payload
- Source IPv6 address is target
- Destination IPv6 address is requestor
- Flags "Router", "Solicited" and "Override" set
- Requested address goes into ICMPv6 part
- Targets Ethernet address goes into an ICMPv6 option



Byte	0	1	2	3
0	Version = 6 / Traffic Class / Flow Label			
4	Payload Length: 32		Next Header ICMPv6	Hop Limit
8	Source IPv6 Address 2001:db8:1:2::33			
12				
16				
20				
24	Destination IPv6 Address 2001:db8:1:2::18/64			
28				
32				
36				
40	Type: 136	Code: 0	Checksum	
44	Flags: e0 00 00 00			
48	Target IPv6 Address 2001:db8:1:2::33			
52				
56				
60				
64	Option: 2	Length: 8	10:dd:b1:bf:cf:be	
68				

Comparison

IPv4 and IPv6

	IPv4	IPv6
Communication	Broadcast	Multicast
Stations involved	all on LAN	limited group
Protocol	ARP (own protocol)	ICMPv6 (part of IPv6)
Winner		**IPv6!**

Conclusion

Learning Ethernet addresses

IPv4 and IPv6

- To send traffic across an Ethernet to an IP node, a sender needs to learn that nodes Ethernet address
- IPv4 and IPv6 have different methods to solve this
  - IPv4 uses ARP - Adress Resolution Protocol on top of Ethernet
  - IPv4 broadcasts ARP requests, and receives ARP replies
  - IPv6 uses ICMPv6 for Neighbor Discovery
  - IP6v sends ND requests to a Multicast Group it knows the receiver is in

Thank you!

academy@de-cix.net

Interested in more webinars? Please subscribe to our mailing list at https://lists.de-cix.net/wws/subscribe/academy

Links and further reading

Links and further reading

- Internet protocol - https://en.wikipedia.org/wiki/Internet_Protocol
- Protocol stack - https://en.wikipedia.org/wiki/Protocol_stack
- IP Network Model: https://en.wikipedia.org/wiki/Internet_protocol_suite
- IP Version Numbers https://www.iana.org/assignments/version-numbers/version-numbers.xhtml#version-numbers-1
- IPv4
  - IPv4 - https://en.wikipedia.org/wiki/IPv4
  - IPv4 address exhaustion - https://en.wikipedia.org/wiki/IPv4_address_exhaustion
  - Map of IPv4 addresses in 2006, 2011
- IPv6
  - IPv6 itself - https://en.wikipedia.org/wiki/IPv6
  - IPv6 header - https://en.wikipedia.org/wiki/IPv6_packet
  - IPv6 addresses - https://en.wikipedia.org/wiki/IPv6_address
  - First standard: RFC1884, current standard: RFC8200
- Routing
  - Packet forwarding - https://en.wikipedia.org/wiki/Packet_forwarding
  - Routing - https://en.wikipedia.org/wiki/Routing

Internet RFCs (Standards)

- IPv6:
  - IPv6 Addressing Architecture: RFC4291
  - IPv6 Neighbor Discovery: RFC4861
  - IPv6 Packets over Ethernet: RFC2464
- IPv4:
  - ARP: RFC826
  - IANA Allocation Guidelines for ARP: RFC5494
- How does something become RFC? https://www.rfc-editor.org/pubprocess/
- The IETF - Internet Engineering Task Force

https://de-cix.net/academy

Interested in more webinars? Please subscribe to our mailing list at https://lists.de-cix.net/wws/subscribe/academy