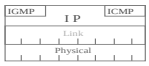
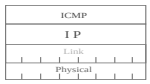
ICMP

- Internet Control Message Protocol



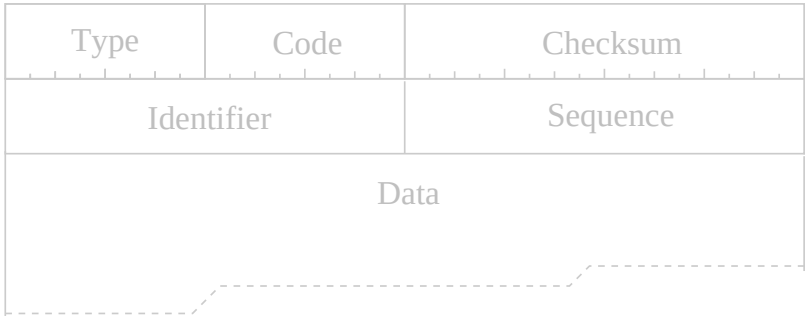


- 3 ICMP message types
  - Requests
  - Replies
  - Error Reports

ICMP format

- Type
  - eg: echo
    - request to return data
  - echo reply
    - reply to earlier echo
- Code
  - Minor variations of type

ICMP Requests & Replies



- Type
  - Select particular request or reply
- Code
  - 0 (unused)
- Identifier
  - Used to match reply with request (port)
- Sequence
  - Used to match reply with request (counter)

ICMP Requests

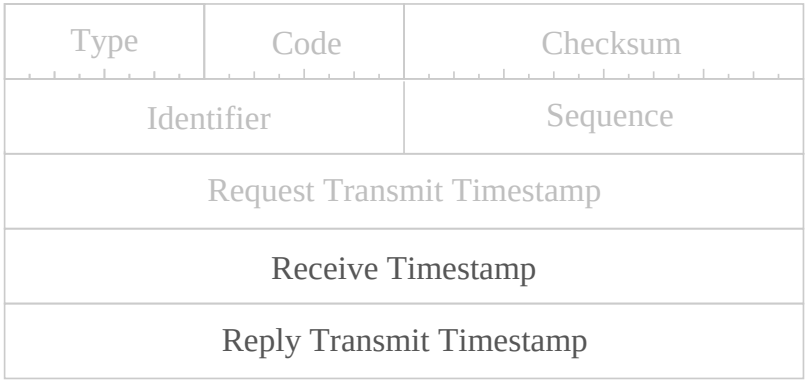
Request	Type	Reply Type	RFC
Echo	8	0	792
Timestamp	13	14	792
Information	15	16	792
Address Mask	17	18	950
Router Solicitation	10	9	1256

ICMP Echo

- Data can be anything
- Same Data returned in ICMP Echo Reply

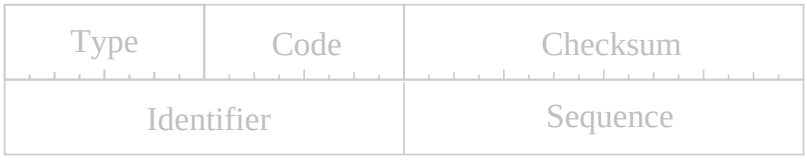
```
02:27:06.105890 192.168.192.23 > 192.168.192.22: icmp: echo request seq 0
                        4500 0054 3e8d 0000 ff01 7b9c c0a8 c017
                        c0a8 c016 0800 b620 0642 0000 0abe 0d3f
                        379d 0100 0809 0a0b 0c0d 0e0f 1011 1213
                        1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
                        2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
                        3435 3637
02:27:06.106051 192.168.192.22 > 192.168.192.23: icmp: echo reply seq 0
                        4500 0054 86dc 0000 ff01 334d c0a8 c016
                        c0a8 c017 0000 be20 0642 0000 0abe 0d3f
                        379d 0100 0809 0a0b 0c0d 0e0f 1011 1213
                        1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
                        2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
                        3435 3637
02:27:07.110189 192.168.192.23 > 192.168.192.22: icmp: echo request seq 1
02:27:07.110337 192.168.192.22 > 192.168.192.23: icmp: echo reply seq 1
02:27:08.120183 192.168.192.23 > 192.168.192.22: icmp: echo request seq 2
02:27:08.120328 192.168.192.22 > 192.168.192.23: icmp: echo reply seq 2
```

ICMP Timestamp Req/Reply



- Reply contains 3 timestamps
  - Time request transmitted
    - inserted by sender of request, copied into reply
  - Time request received
    - inserted by receiver of request
  - Time reply transmitted
    - inserted by sender of reply
- Times in milliseconds since midnight (UT)
  - Or other representation with MSB set

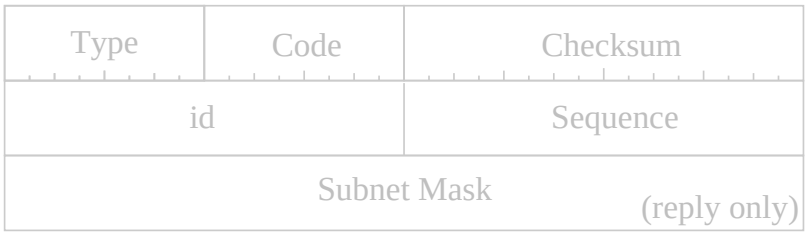
ICMP Information Req/Reply



- Note: No data in packet
  - IP address fields are the data
- Request sent with 0's in network part of addresses
  - Means this network
    - Only for this packet type...
- Reply fills in complete addresses
  - Allows host to discover network number
- Discover all the problems with this...

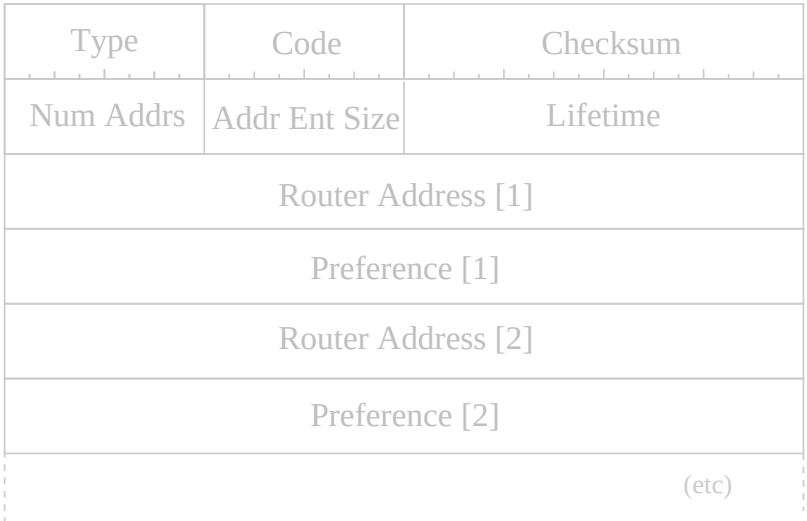
Address Mask Request

- Node wants to discover subnet mask



  - Send address mask request
    - usually broadcast
  - Receive address mask reply
    - usually many
- But no guarantee
  - that all hosts correctly configured
  - Not all replies the same
  - No useful information
- Address Mask messages deprecated

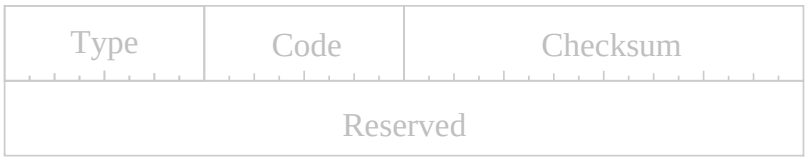
Router Advertisement (v4)



- Router gives all its addresses
  - On the subnet it sends the message
- And a preference for each address
  - Relative to all router addresses on the net

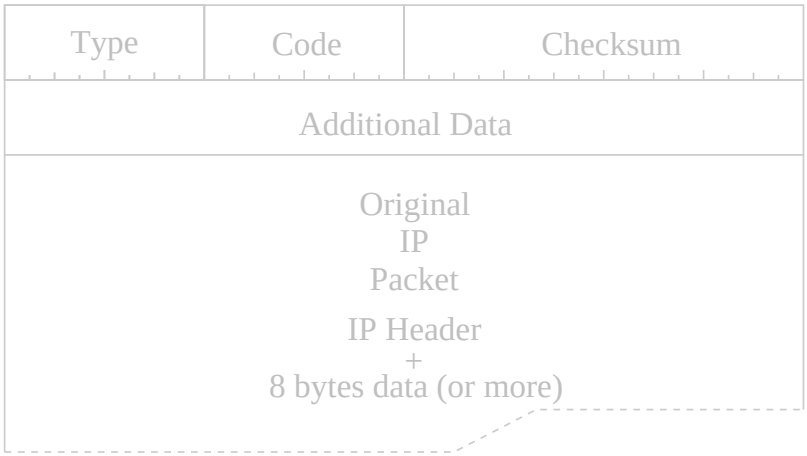
Router Advertisement (v4)

- Router also gives a lifetime (seconds)
- Num Addrs
  - Number of Addresses in message
- Addr Ent Size
  - Size in 32 bit words of each entry (2)
- Solicitation



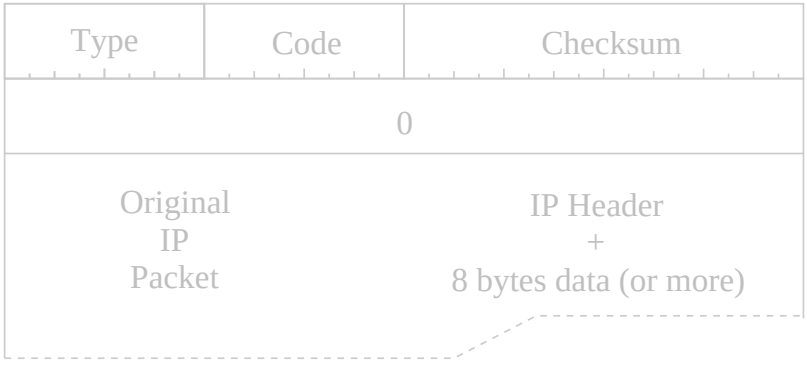
  - Request routers to send Router Advert

ICMP errors



- Additional Data
  - Sometimes indicates where error occurred
- Original Packet
  - Allows source to recognise
    - which packet had problem
  - 8 data bytes is usually transport header
    - or part of it

ICMP Unreachable



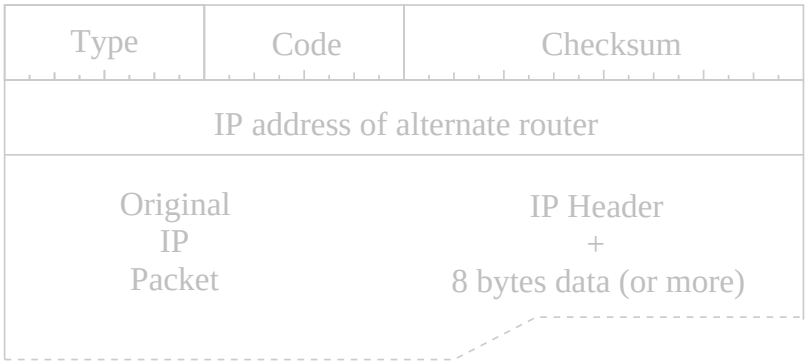
- Packet could not be delivered
- No additional data
  - Except modern Fragmentation Required: MTU
- Code: Why...

Network Host Protocol
Port Need to Fragment Source Route Fail
Net Unknown Host Unknown Source Isolated
Net Prohibited Host Prohibited Bad TOS (Net)
Bad TOS (Host) Admin Prohibit

ICMP Source Quench

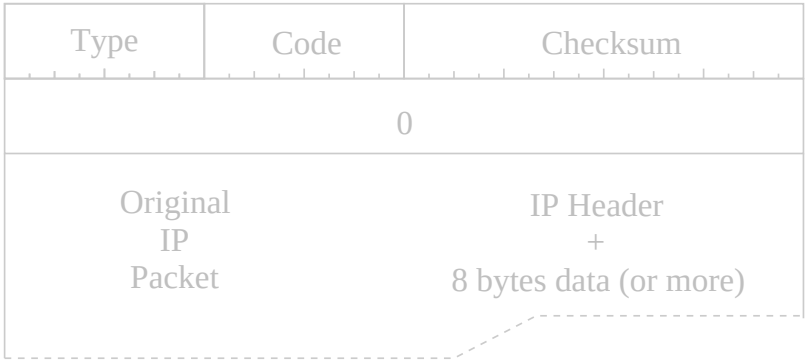
- Used to tell source host to send less packets
  - To destination of returned packet
- Same format as Unreachable
  - Code always 0
- Not used
  - No authentication
  - No reason for host to trust sender

ICMP Redirect



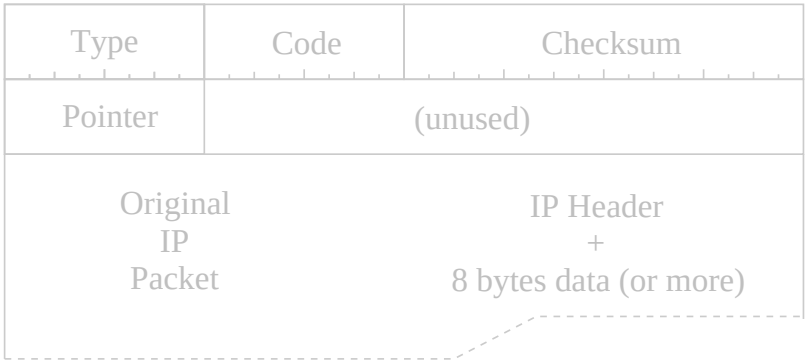
- If host chooses wrong router
  - hosts have no information to make choice
- router still forwards packet
- router also returns redirect
  - next packet to destination should be sent to the alternate router address specified
- Code:
  - host or net; always, or for TOS used

ICMP Time Exceeded



- TTL decremented to 0
- Code 0
  - In Transit
    - while being forwarded
- Code 1
  - In reassembly queue
    - waiting for later fragments to arrive
    - only first fragment (offset == 0)

ICMP Parameter Problem



- Error of some kind in headers
  - Packet was discarded
    - No ICMP error sent if packet forwarded
- Pointer is offset to the byte of the header where the error was detected

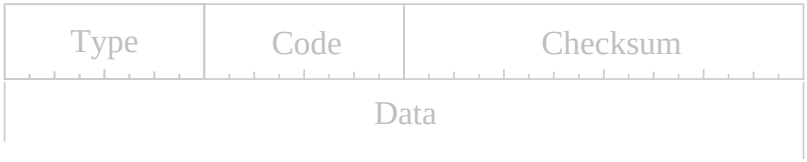
ICMP Error Rule

- Never send ICMP error packet if
  - packet with the problem was an ICMP error packet
- Error types
  - Unreachable (3) Source Quench (4)
  - Redirect (5) Time Exceeded (11)
  - Param Prob (12)
- Not Errors
  - Echo/Reply (8/0) Time/Reply (13/14)
  - Info/Reply (15/16) Mask/Reply (17/18)
  - Router Solicit/Advert (10/9)
- What about type 6?

ICMPv6

- Entirely new protocol
  - ICMP: Protocol 1
  - ICMPv6: Protocol 58
- ICMP Error Message
  - Types 0..127
- ICMP Information messages (Requests/Replies)
  - Types 128..255
- Error messages return as much of original packet as fits
  - So ICMP packet will not be bigger than 1280 bytes

ICMPv6



- Just the same as ICMP (for IPv4)

ICMPv6 Error messages

- Destination Unreachable Type 1
- Packet Too Big Type 2
- Time Exceeded Type 3
- Parameter Problem Type 4

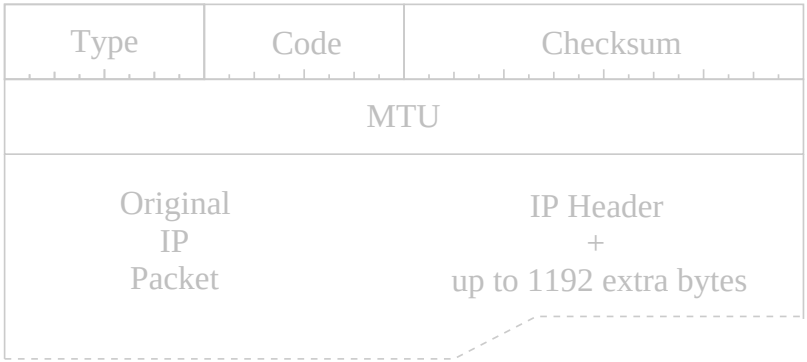
- Deleted
  - Source Quench
- Moved to Informational
  - Redirect

ICMPv6 Unreachable

- Same format as for IPv4
  - Except more of failed packet expected to be included

- Only 5 codes
  - No Route
  - Communication Admin Prohibited
  - Going Out of Scope
  - Address Unreachable
  - Port Unreachable

ICMPv6 Packet Too Big



- Replaces IPv4 Unreachable: Need Fragmentation
- MTU specifies the MTU that was too small for packet
  - Allows PMTU Discovery
- Separate code from Unreachable
  - allows unreachables to be blocked
  - still allow "too big" to be received

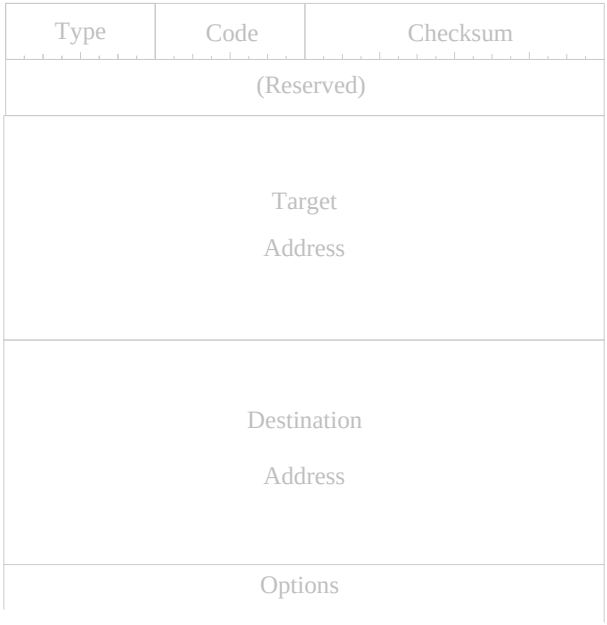
ICMPv6 Errors Concluded

- Time Exceeded
  - Identical to IPv4 ICMP (with more data)
- Parameter Problem
  - Similar to IPv4 parameter problem
    - 32 bit pointer
    - code indicates general problem
      - 0 bad header field
      - 1 unrecognised next header
      - 2 unrecognised option

ICMPv6 Information

- Echo Request/Reply
  - Identical to IPv4 Echo
- No: Address Mask, Timestamp, Info Request
- Router Solicitation
- Router Advertisement
- Neighbour Solicitation
- Neighbour Advertisement
- Redirect
  - Much more structured messages
    - NS/NA new

ICMPv6 Redirect

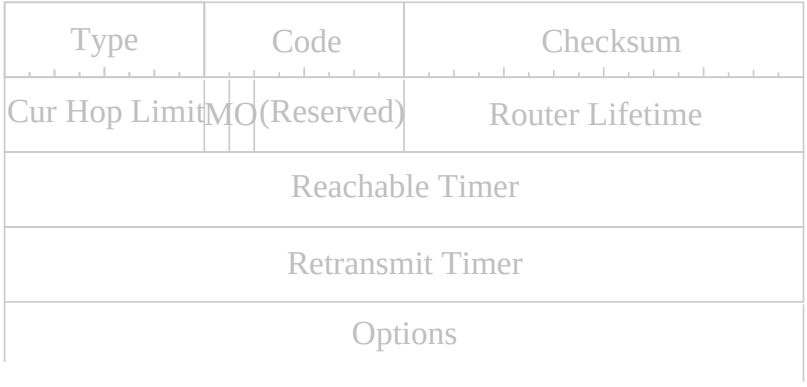


ICMPv6 Redirect

- Target Address
  - Address of Router (Link Local)
    - Or of destination host
- Destination Address
  - Address that is being redirected
- Options:

  - Target Link Layer Address
  - Redirected Header

ICMPv6 Router Advertisement



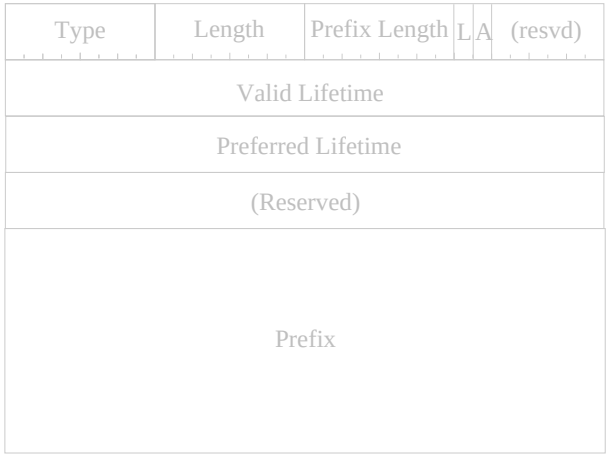
- Announce sender as a router
- Provide information to nodes on link
- Always sent from Link Local Address
  - Hop Limit == 255

ICMPv6 RA Information

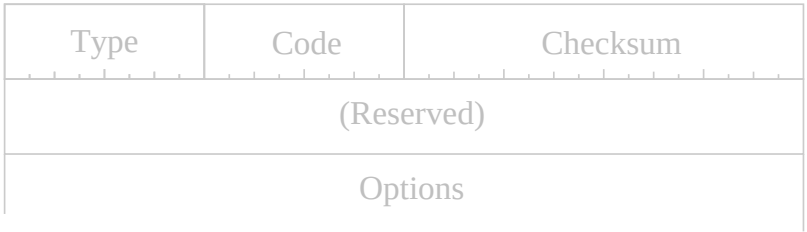
- Fixed Fields:
  - Cur Hop Limit
    - Hop Limit hosts should use
  - M
    - Managed Configuration
  - O
    - Other stateful config
  - Reserved
    - Now contains router preference (2 bits)
  - Router Lifetime
    - seconds for default router list
  - Reachable Time
    - How long confirmation implies reachable
  - Retrans Timer
    - Neighbour Solicitation retransmit timer

ICMPv6 RA Information

- Options:
  - Source Link Layer Address
  - MTU
  - Prefix Information



ICMPv6 Router Solicitation



- Options:
  - Source Link Layer Address

ICMPv6 Information

- Echo Request/Reply
- Router Solicitation
- Router Advertisement
- Redirect
- Neighbour Solicitation
- Neighbour Advertisement

ICMPv6 echo

```
11:52:37.819986 0:6:5b:da:45:ad 0:10:5a:76:e9:b4 86dd 70:
    fe80::206:5bff:feda:45ad > fe80::210:5aff:fe76:e9b4:
        icmp6: echo request
                        (len 16, hlim 64)

             6000 0000 0010 3a40 fe80 0000 0000 0000
             0206 5bff feda 45ad fe80 0000 0000 0000
             0210 5aff fe76 e9b4 8000 ed19 6ec5 0001
             3f17 7d15 000c 82d1


11:52:37.820127 0:10:5a:76:e9:b4 0:6:5b:da:45:ad 86dd 70:
    fe80::210:5aff:fe76:e9b4 > fe80::206:5bff:feda:45ad:
        icmp6: echo reply
                        (len 16, hlim 64)

             6000 0000 0010 3a40 fe80 0000 0000 0000
             0210 5aff fe76 e9b4 fe80 0000 0000 0000
             0206 5bff feda 45ad 8100 ec19 6ec5 0001
             3f17 7d15 000c 82d1
```

Neighbour Discovery

- Find Mac Address for address on Link
- Also
  - Duplicate Address Detection
  - Neighbour Unreachability Detection
- First:
  - The IPv4 method...

Address Resolution Protocol

- Have IP address from application/transport
- Want MAC (link layer address) so packet can be transmitted
- How to translate?

ARP Possibilities

- Could embed IP address in MAC address
  - using defined value for the extra bytes
    - (DECNET 4 used that method)
- With 3 Mbps ethernet
  - 16 bit addresses
  - embed ethernet address in IP address
    - simply extract it when needed
- Build a DNS like server mechanism (database) and send a query

ARP Solution

- Simply ask!
- Broadcast packet sent to the link layer
  - "Who has the IP address a.b.c.d?"
  - Broadcast because when this was invented,
    - multicast was not defined
    - Means that every node
      - on the link layer
      - receives the packet

ARP Examples

```
18:11:54 arp who-has 172.30.0.77
             tell 172.30.0.9

18:11:54 arp reply
         172.30.0.77 is-at 0:10:a4:f:41:cf


18:11:55 arp who-has 172.30.0.161
             tell 172.30.0.77

18:11:55 arp reply
         172.30.0.161 is-at 0:1:3:40:8a:e5
```

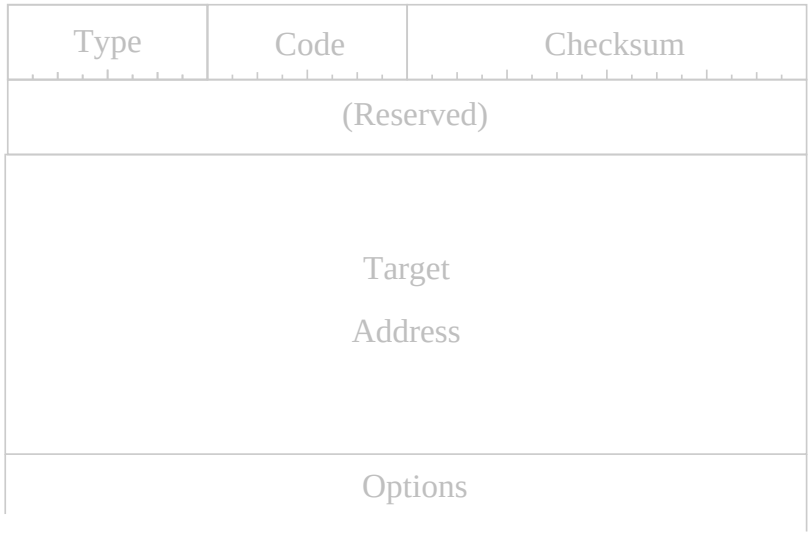
ARP Example (with link layer)

```
0:e0:29:1d:ec:c8 ff:ff:ff:ff:ff:ff 0806 60:
   0001 0800 0604 0001 00e0 291d ecc8 ac1e
   0040 0000 0000 0000 ac1e 0005 0a2b 5cfb
   0000 0000 0000 0000 0000 0000 0000

arp who-has 172.30.0.5 tell 172.30.0.64

0:c0:4f:9a:20:56 0:e0:29:1d:ec:c8 0806 60:
   0001 0800 0604 0002 00c0 4f9a 2056 ac1e
   0005 00e0 291d ecc8 ac1e 0040 0040 0040
   0040 0040 0040 0040 0040 0040 0040

arp reply 172.30.0.5 is-at 0:c0:4f:9a:20:56
```

IPv6 Neighbour Discovery

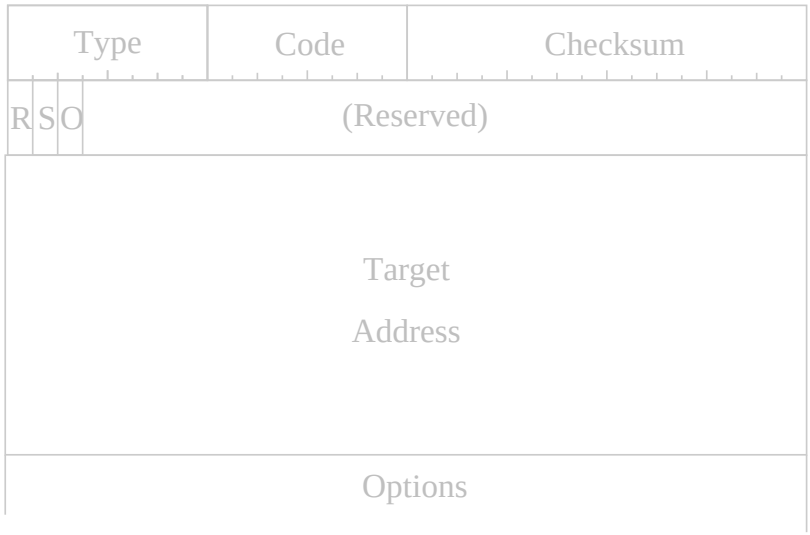
- Same general principles as ARP
- But
  - ICMP instead of link level
    - Means same protocol on all links
  - Multicast instead of Broadcast
    - Multicast group depends upon target address
    - Often only one node will receive request

IPv6 Neighbour Solicitation



- Target Address
  - The IPv6 address sought
- Options
  - Source Link Layer Address

ICMPv6 Neighbour Advert



- R Sender is a router
- S Solicited
- O Override
- Options

ND Examples

```
11:52:36.832814 0:6:5b:da:45:ad 33:33:ff:76:e9:b4 86dd 86:

    fe80::206:5bff:feda:45ad > ff02::1:ff76:e9b4:

       icmp6: neighbor sol:
                 who has fe80::210:5aff:fe76:e9b4

       (src lladdr: 00:06:5b:da:45:ad)
                        (len 32, hlim 255)


11:52:36.833011 0:10:5a:76:e9:b4 0:6:5b:da:45:ad 86dd 86:

   fe80::210:5aff:fe76:e9b4 > fe80::206:5bff:feda:45ad:

       icmp6: neighbor adv:
                 tgt is fe80::210:5aff:fe76:e9b4(SO)

       (tgt lladdr: 00:10:5a:76:e9:b4)
                        (len 32, hlim 255)
```

ND Examples expanded

```
11:52:36.832814 0:6:5b:da:45:ad 33:33:ff:76:e9:b4 86dd 86:
    fe80::206:5bff:feda:45ad > ff02::1:ff76:e9b4:
       icmp6: neighbor sol:
                  who has fe80::210:5aff:fe76:e9b4
       (src lladdr: 00:06:5b:da:45:ad)
                        (len 32, hlim 255)

             6000 0000 0020 3aff fe80 0000 0000 0000
             0206 5bff feda 45ad ff02 0000 0000 0000
             0000 0001 ff76 e9b4 8700 091c 0000 0000
             fe80 0000 0000 0000 0210 5aff fe76 e9b4
             0101 0006 5bda 45ad


             6000 0000 0020 3aff
             fe80 0000 0000 0000 0206 5bff feda 45ad
             ff02 0000 0000 0000 0000 0001 ff76 e9b4

             8700 091c 0000 0000
             fe80 0000 0000 0000 0210 5aff fe76 e9b4
             0101 0006 5bda 45ad
```

ND Examples expanded

```
11:52:36.833011 0:10:5a:76:e9:b4 0:6:5b:da:45:ad 86dd 86:
   fe80::210:5aff:fe76:e9b4 > fe80::206:5bff:feda:45ad:
       icmp6: neighbor adv:
                  tgt is fe80::210:5aff:fe76:e9b4(SO)
       (tgt lladdr: 00:10:5a:76:e9:b4)
                        (len 32, hlim 255)

             6000 0000 0020 3aff fe80 0000 0000 0000
             0210 5aff fe76 e9b4 fe80 0000 0000 0000
             0206 5bff feda 45ad 8800 a8e1 6000 0000
             fe80 0000 0000 0000 0210 5aff fe76 e9b4
             0201 0010 5a76 e9b4


             6000 0000 0020 3aff
             fe80 0000 0000 0000 0210 5aff fe76 e9b4
             fe80 0000 0000 0000 0206 5bff feda 45ad

             8800 a8e1 6000 0000
             fe80 0000 0000 0000 0210 5aff fe76 e9b4
             0201 0010 5a76 e9b4
```

ICMP echo sequence

```
11:52:36.832814 0:6:5b:da:45:ad 33:33:ff:76:e9:b4 86dd 86:
    fe80::206:5bff:feda:45ad > ff02::1:ff76:e9b4:
       icmp6: neighbor sol:
                  who has fe80::210:5aff:fe76:e9b4
       (src lladdr: 00:06:5b:da:45:ad)
                        (len 32, hlim 255)

11:52:36.833011 0:10:5a:76:e9:b4 0:6:5b:da:45:ad 86dd 86:
   fe80::210:5aff:fe76:e9b4 > fe80::206:5bff:feda:45ad:
       icmp6: neighbor adv:
                  tgt is fe80::210:5aff:fe76:e9b4(SO)
       (tgt lladdr: 00:10:5a:76:e9:b4)
                        (len 32, hlim 255)

11:52:36.833050 0:6:5b:da:45:ad 0:10:5a:76:e9:b4 86dd 70:
   fe80::206:5bff:feda:45ad > fe80::210:5aff:fe76:e9b4:
       icmp6: echo request
                        (len 16, hlim 64)

11:52:36.833162 0:10:5a:76:e9:b4 0:6:5b:da:45:ad 86dd 70:
   fe80::210:5aff:fe76:e9b4 > fe80::206:5bff:feda:45ad:
       icmp6: echo reply
                        (len 16, hlim 64)
```