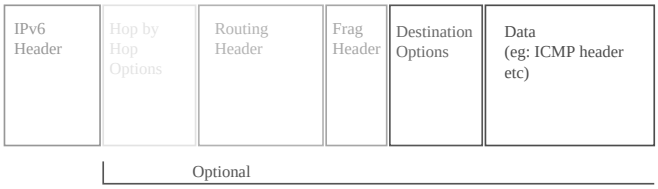
IPv6 Header Chains



- Each Header (before data) contains a "Next Header" field
  - Specifies the type of the header that follows
  - The first header is always an IPv6 header
- Each Header is of fixed length, or provides a mechanism to allow its length to be determined
  - Length field in most headers

Header Processing

- Headers are processed "left to right" through the packet
  - Unrecognised header is an error,
    - ICMP error report
    - Drop packet
- Routers look only at IPv6 header, and Hop-by-Hop Options
  - Easy to tell if HBH is present
    - Next Header field in IPv6 Header (0)

Options Header

- Header Length
  - Counts number of 64 bit words after the first
  - All (new) IPv6 headers are a multiple of 8 bytes
- Each option contains Type Length (and Data)
  - Can skip unknown options

IP Options

- Some options for destination node
- Some options for routers along path
- Some options for some routers along path

- IPv4 options
  - In IPv4 header
  - All routers must examine all options
    - Only way to find options that need processing

- Many IPv4 options in RFC791
  - Source Routing
  - Route record
  - Timestamp record
- Almost none in use now

IPv6 Options

- Multiple Option headers possible

- Options go in header that will be seen by appropriate node

  - All routers look at options in "Hop By Hop"
    - Special case
  - When Dest Addr in IPv6 header is reached
    - Routing options processed
    - Routing header processed
    - New IPv6 dest addr inserted
  - When routing header runs out
    - Final destination
    - Destination options processed

Using options

- Using IP header option in IPv4
  - Causes every router to have to examine options
  - Causes packet to not be hardware forwarded
  - Slows processing - sometimes changes routing
    - 2nd class packet
  - Hence options not used

- 40 byte max option space
  - Not enough for interesting source routes
  - not enough to record routes, or timestamps
  - hence options not used

- IPv6 suffers neither of those problems

IPv6 Addresses

- 128 bits
- Like IPv4, divided into
  - network identifier
  - host identification on that network

    Eg:
    172.30.2.60 netmask 255.255.255.0

    172.30.2.0 is the network identifier
    0.0.0.60 is the host identifier on that network
- Netmask can divide anywhere, not just at one of the dots

IPv6 Prefix

- The "network part" of an IPv6 is the prefix
- Prefix identified by a length, rather than a mask
  - written /nn (as in /32 or /57)
- Prefix is never longer than 64
  - On a standard network
- That is, 64 bits are always available to identify a host on the network

IPv4 Host Configuration

- Host must be configured with
  - IP (IPv4) Address
  - Netmask
  - Address of a router to use
  - (other info usually, eg: DNS server)
- Manual configuration
  - Difficult
  - Error Prone
- DHCP
  - Dynamic Host Configuration Protocol
  - Successor to BOOTP

DHCP

- Will examine DHCP in more detail later
- However ...
  - DHCP requires a server
    - That server must be configured
      - OK for large nets (university, company)
      - Not great for small nets (home, shop)
  - DHCP server
    - can implement any address assignment policy desired
      - SMOP ("simple matter of programming")
  - eg: assign addresses only to "known" hosts
  - or assign addresses to anyone who asks

IPv6 Autoconfiguration

- The 64 bit host part is big enough to contain a MAC address
  - Not an accident
- Host is shipped with MAC address available from hardware
  - PROM or similar
- If host can learn the prefix of the local network, it can create its own IPv6 address by simply putting its MAC address in the low 64 bits
  - MAC address is unique (enough), so the address formed should be unique too.

IPv6 Packet Example

```
0:10:a4:f:41:cf 0:1:3:40:8a:e5 86dd

 3ffe:8001:2:181:210:a4ff:fe0f:41cf >
  3ffe:8001:2:181:201:3ff:fe40:8ae5
     icmp6: echo request
```

- Note similarity between MAC and IPv6 addresses
  - Both source and destination

```
                    00:10:a4       :0f:41:cf
3ffe:8001:2:181:02 10:a4 ff:fe 0f:41 cf
```

Generating IPv6 Addresses

```
00:10:a4      :0f:41:cf
02 10:a4 ff:fe 0f:41 cf
```

- FFFE is the defined way to transform a 48 bit MAC address into a 64 bit MAC address (EUI-64)
  - FFFE is inserted between the 3rd and 4th octets of 48 bit address
- 02 in the first address octet is the "locally defined" bit
  - In MAC address bit set indicates a locally defined address
  - In IPv6 address, bit is inverted