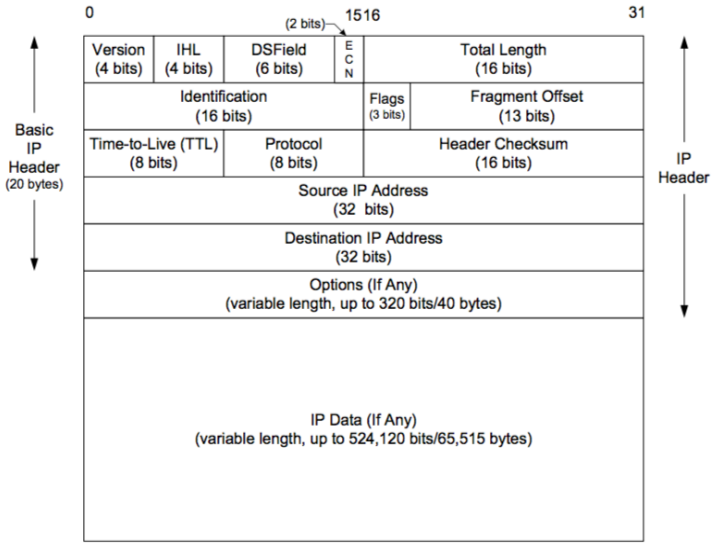
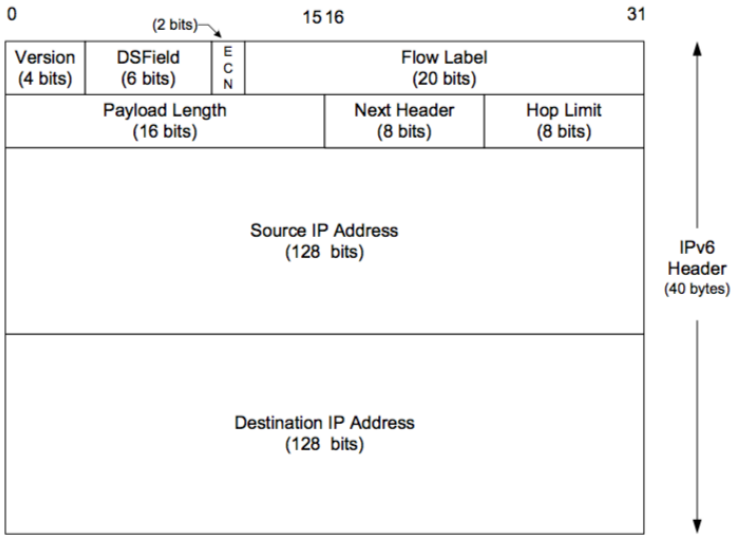
- Chapter 5
- The Internet Protocol (IP)
- 5.1 Introduction
  - IP is the workhorse protocol of the TCP/IP protocol suite
  - IP provides a best-effort, connectionless datagram delivery service
    - “best-effort” - there are not guarantees that an IP datagram gets to its destination successfully
    - when something goes wrong, such as a router temporarily running out of buffers, IP has a simple error-handling algorithm: throw away some data (usually the last datagram that arrived).
      - any required reliability must be provided by the upper layers
  - connectionless
    - means that IP does not maintain any connection state information about related datagrams within the network elements (i.e. within the routers); each datagram is handled independently from all others
    - means IP datagrams can be delivered out of order
  - IPv4 header



    - the header is of variable size, limited to fifteen 32-bit words
  - IPv6 header

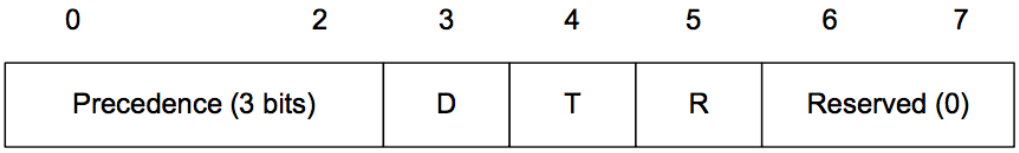


    - next header field is used to indicate the presence and types of additional extension headers that follow the IPv6 header, forming a daisy chain of headers that my include special extensions or processing directives
    - application data follows the header chain, usually immediately following a transport-layer header
- 5.2 IPv4 and IPv6 Headers
  - normal size of the IPv4 header is 20bytes, unless options are present
  - IPv6 header is twice as large but never has any options
    - it may have extension headers, which provide similar capabilities,as we shall see later
  - bytes are transmitted in big endian byte ordering required for all binary integers in the TCP/IP headers as they traverse a network
    - also called network byte order
- 5.2.1 IP Header Fields
  - Version Field
    - contains the version number of the IP datagram
    - 4 for IPv4 and 6 for IPv6
  - Internet Header Length (IHL)
    - is the number of 32-bit words in the IPv4 header, including any options
    - there is no such field in IPv6 because the header length is fixed at 40 bytes
  - Differentiated Services Field (DS Field), and the last 2 bits are the Explicit Congestion Notification (ECN) field or indicator bits
    - these fields are used for special processing of the datagram when it is forwarded
  - Total Length field
    - total length of the IPv4 datagram in bytes
    - when IPv4 is fragmented into multiple smaller fragments, each of which itself is an independent IP datagram, the Total Length field reflects the length of the particular fragment
    - IPv6 does support fragmentation in the header, and the length is instead given by the Payload Length field
  - Payload Length field (IPv6)
    - field measures the length of the IPv6 datagram not including the length of the header; extension headers, however, are included in the Payload Length field.
  - Identification field (IPv4)
    - helps identify each datagram sent by an IPv4 host.
      - internal counter is incremented each time a datagram is sent
  - Time-to-Live field (IPv4) - sets an upper limit on the number of routers through which a datagram can pass
    - initialized by the sender to some value and decremented by 1 every router that forwards the datagram
    - when this field reaches 0, the datagram is thrown away, and the sender is notified with an ICMP message
    - this prevents packets from getting caught in the network forever should an unwanted routing loop occur
  - Protocol Field (IPv4)
    - contains a number indicating the type of data found in the payload portion of the datagram
    - 17 for UDP and 6 for TCP
    - provides demultiplexing feature so that the IP protocol can be used to carry payloads of more than one protocol type

- Next Header field (IPv6)
  - generalizes the Protocol field from IPv4
  - used to indicate the type of header following the IPv6 header
  - may contain any values defined for the IPv4 Protocol field, or any of the values associated with the IPv6 extension headers
- Header Checksum (IPv4)
  - calculated over the IPv4 header only
  - means that the payload of the IPv4 datagram is not checked for correctness by the IP protocol
  - algorithm used in computing the checksums and is sometimes known as the Internet checksum must change as a result of decrementing TTL field.
- Source IP Address
  - contains the source IP address of the sender of the datagram
- Destination IP Address
  - contains the destination IP address of where the datagram is destined
  - 32 bit values for IPv4 and 128 bit values for IPv6
  - usually identify a single interface on a computer, although multicast and broadcast addresses violate this rule

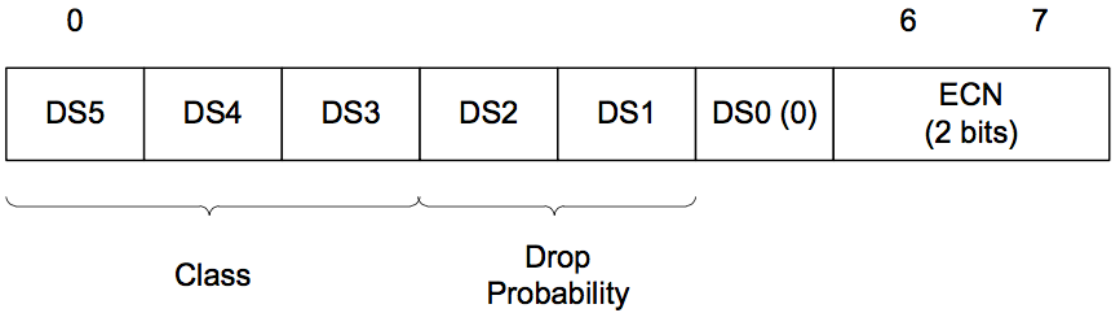
- 5.2.2 The Internet Checksum
  - 16-bit mathematical sum used to determine, with reasonably high probability, whether a received message or portion of a message matches the one sent.
  - Internet checksum algorithm is not the same as the common cyclic redundancy check (CRC) which offers stronger protection
  - Compute value
    - value of the datagram's Checksum field is first set to 0.
    - Then, 16-bit one's complement sum of the header is calculated (the entire header is considered a sequence of 16-bit words)
    - The 16-bit one's complement of this sum is then stored in the Checksum field to make the datagram ready for transmission.
  - When an IPv4 datagram is received, a checksum is computed across the whole header, including the value of the Checksum field itself.
    - assuming there are no errors, the computed checksum value is always 0
  - For any nontrivial packet or header, the value of the Checksum field in the packet can never be FFFF
    - if it were the sum (prior to the final one's complement operation at the sender) would have been 0—something that never happens with any legitimate IPv4 header.
  - When the header is found to be bad (the computed checksum is nonzero), the IPv4 implementation discards the received datagram.
    - up to the higher layers to somehow detect the missing datagram and retransmit if necessary
- 5.2.2.1 Mathematics of the Internet Checksum
- Abelian group
  - for the combination of a set and an operator to be a group, several properties need to be obeyed: closure, associativity, existence of an identity element, and existence of inverses.
  - the number 0000 is deleted from consideration for the group
- 5.2.3 DS Field and ECN (Formerly Called the Tos Byte or IPv6 Traffic Class)
  - Differentiated Services - is a framework and set of standards aimed at supporting differentiated classes of service on the internet.

- IP datagrams that are marked in certain ways (by having some of these bits set according to predefined patterns) may be forwarded differently than other datagrams.
- DS Field - a number is placed into the field termed the Differentiated Services Code Point (DSCP).
  - a “code point’ refers to a particular predefined arrangement of bits with agreed-upon meaning.
  - typically, datagrams have a DSCP assigned to them when they are given to the network infrastructure that remains unmodified during delivery.
    - However, policies (such as how many high-priority packets are allowed to be sent in a period time) may cause a DSCP in a datagram to be changed during delivery
- ECN field
  - is used for marking a datagram with congestion indicator when passing through a router that has a significant amount of internally queued traffic
  - both bits are set by persistently congested ECN-aware routers when forwarding packets
  - The use case envisioned for this function is that when a marked packet is received at the destination, some protocol (such as TCP) will notice that the packet is marked and indicate this fact back to the sender, which would then slow down, thereby easing congestion before a router is forced to drop traffic because of overload.
- original uses for the ToS (Type of Service) and Traffic Class bytes are not widely supported the structure of the DS Field has been arranged to provide some backward compatibility with them
- Original structure of Type of Service field



  - The original IPv4 Type of Service and IPv6 Traffic Class field structures
  - Precedence subfield was used to indicate which packets should receive higher priority (larger values means higher priority)
    - values range from 000 (routine) to 111 (network control) with increasing priority
    - based on a call preemption scheme called Multilevel Precedence and Preemption (MLPP)
  - D, T, and R subfields refer to delay, throughput, and reliability.
    - a value of 1 in these fields corresponds to a desire for low delay, high throughput, and high reliability, respectively.
- DS Field
  - precedence values have been taken into account so as to provide a limited form of backward compatibility
  - 6-bit DS Field holds the DSCP, providing support for 64 distinct code points
    - the particular value of the DSCP tells the router the forwarding treatment or special handling the datagram should receive.
    - the various forwarding treatments are expressed as per-hop behavior (PHB), so the DSCP value effectively tells a router which PHB to apply to the datagram
    - default value is 0, which corresponds to routine, best-effort Internet traffic
    - The 64 possible DSCP values are broadly divided into a set of pools for various uses.

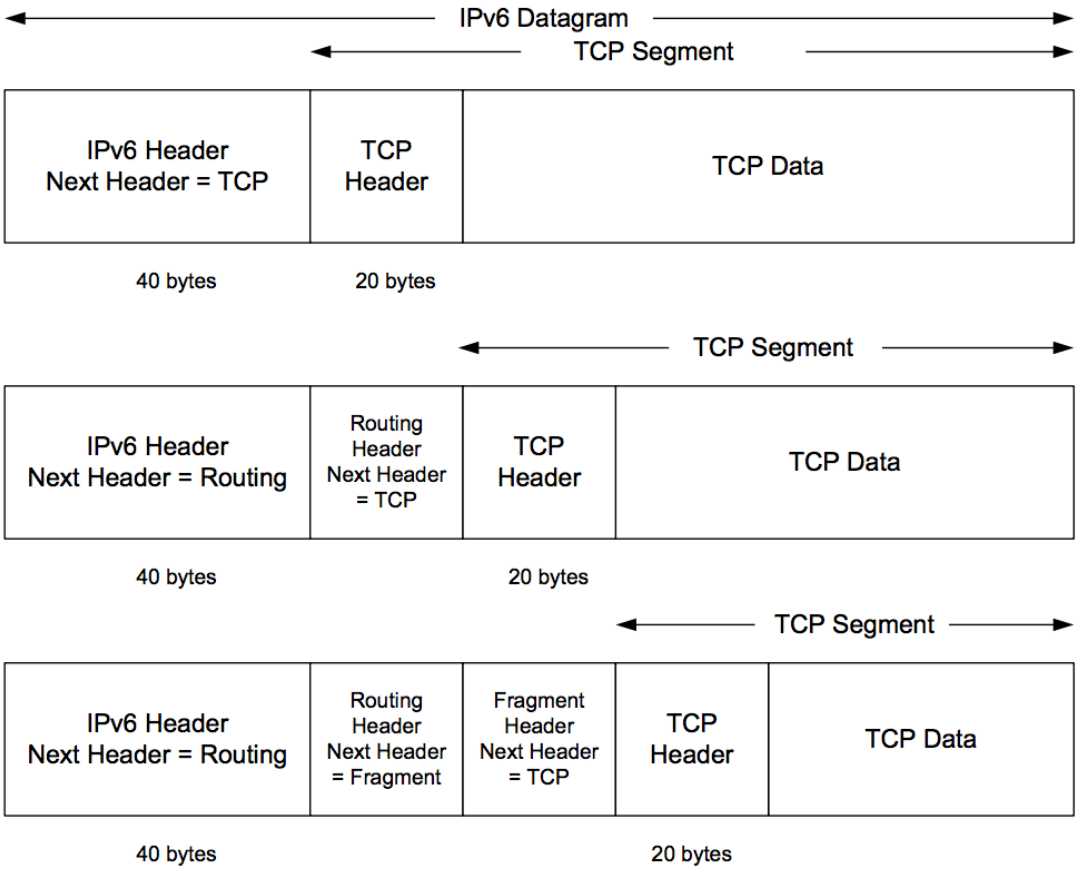
- DSCPs
  - ending in 0 are subject to standardized use
  - ending in 1 are for experimental/local use
  - ending in 01 are intended initially for experimentation or local use but with eventual intent toward standardization



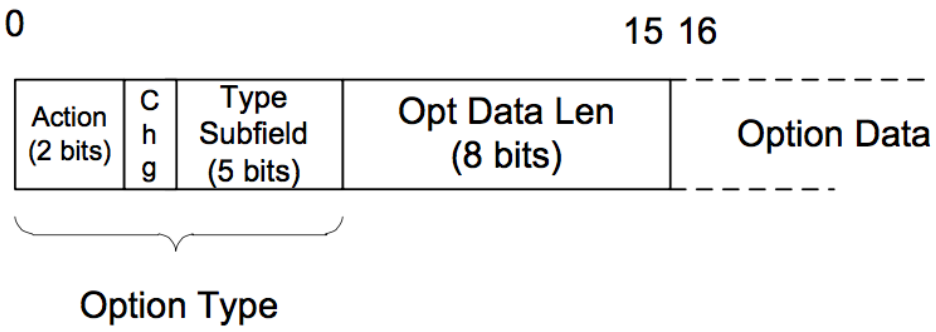
- DS field structure
  - class - portion of the DS Field contains the first 3 bits and is based on the earlier definition of the Precedence subfield of the Type of Service field.
  - a router is to first segregate traffic into different classes
    - traffic within a common class may have different drop probabilities, allowing the router to decide what traffic to drop first if it is forced to discard traffic
  - The 3 bit class selector provides for eight defined code points (called the class selector code points) that correspond to PHBs with a specified minimum set of features providing similar functionality to the earlier IP precedence capability
    - these are called class selector compliant PHBs. They are intended to support partial backward compatibility with the original definition given for the IP Precedence subfield
    - Code points of the form xxx000 always map to such PHBs, although other values may also map the the same PHBs.
- Assured Forwarding (AF)
  - group provides forwarding of IP packets in a fixed number of independent AF classes, effectively generalizing the precedence concept.
  - Traffic from one class is forwarded separately from other classes.
  - within a traffic class, a datagram is assigned a drop precedence
    - datagrams of higher drop precedence in a class are handled preferentially (i.e. forwarded with higher priority)
  - combining the traffic class and drop precedence, the name AFij corresponds to assured forwarding class i with drop precedence j.
- Expedited Forwarding (EF)
  - provides the appearance of an uncongested network—that is, EF traffic should receive relatively low delay, jitter, and loss
  - Intuitively, this requires the rate of EF traffic going out of a router be at least as larges as the rate coming in.
  - EF traffic will only ever have to wait in a router queue behind other EF traffic

- 5.2.4 IP Options
- IPv4
  - most options are no longer practical or desirable because of the limited size of the header or concerns regarding security

- IPv6
  - most of the options have been removed or altered and are not an integral part of the basic IPv6 header
  - instead they are placed after the IPv6 header in one or more extension headers
- An IP router that receives a datagram containing options is usually supposed to perform special processing on the datagram
- Options, if present are carried in IPv4 packets immediately after the basic IPv4 header. Options are identified by an 8-bit option Type field. This field is subdivided into three subfields: Copy(1 bit), Class(2 bits), and Number(5 bits). Options 0 and 1 are a single byte long, and most others are variable in length. Variable options consist of 1 byte of type identifier, 1 byte of length, and the option itself.
- 5.3 IPv6 Extension Headers
- special functions such as those provided by options in IPv4 can be enabled by adding extension headers that follow the IPv6 header. The routing and timestamp functions from IPv4 are supported this way, as well as some other functions such as fragmentation and extra-large packets that were deemed to be rarely used for most IPv6 traffic.
- IPv6 header is fixed at 40 bytes, and extension headers are added only when needed
  - design and construction of high performance routers can be simpler than IPv4
- Extension headers, along with headers of higher-layer protocols such as TCP or UDP, are chained together with the IPv6 header to form a cascade of headers
- Next Header field - in each header indicates the type of each subsequent header, which could be an IPv6 extension header or some other type



- The value 59 indicates the end of the header chain.
- IPv6 implementation must be prepared to process extension headers in the order in which they are received
- Destination options header can be used twice
  - first time for options pertaining to the destination IPv6 address contained in the IPv6 header
  - second time for options pertaining to the final destination of the datagram
- Destination IP Address - In some cases the field in the IPv6 header changes as the datagram is forwarded to its ultimate destination
- 5.3.1 IPv6 Options
- IPv6 provides a more flexible and extensible way of incorporating extensions and options as compared to IPv4.
  - options from IPv4 that ceased to be useful because of space limitations in the IPv4 header appear in IPv6 as variable-length extension headers or options encoded in special extension headers that can accommodate today's much larger Internet.
  - Options if present
    - options headers are capable of holding more than one option
    - each of these options is encoded as type-length-value (TLV) sets, according to the format shown.



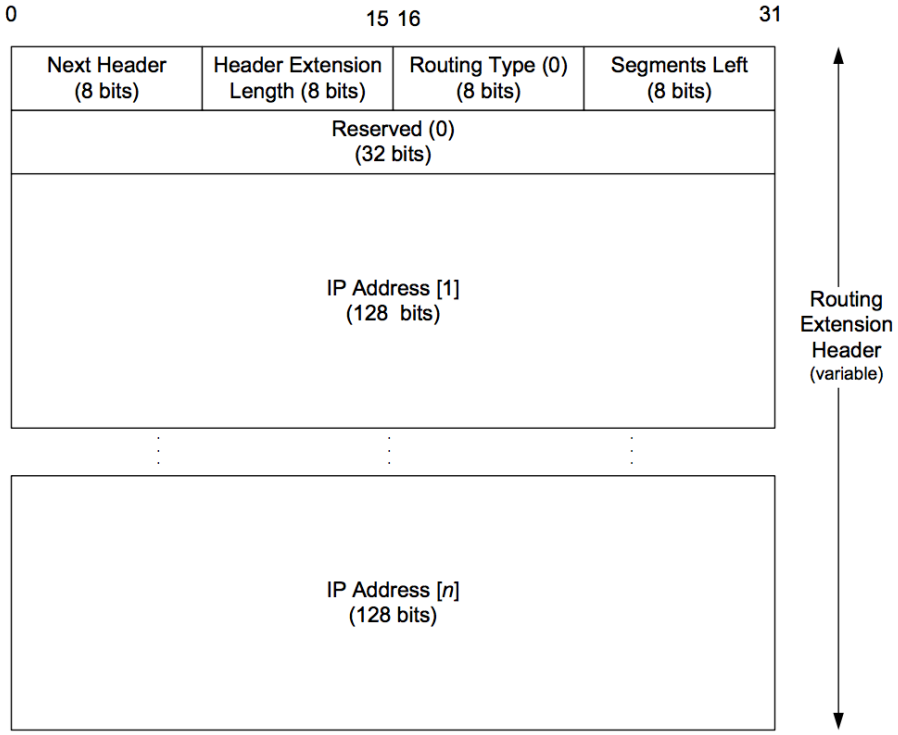
  - TLV
    - Consists of 2 bytes followed by a variable length number of data bytes
    - First byte
      - indicates the type of the option and includes three subfields
        - first subfield - action
          - gives the action to be taken by an IPv6 node attempting to process the option that does not recognize the 5-bit option type subfield
          - indicate whether an IPv6 node should forward or drop the datagram if the option is not recognized, and whether a message indicating the datagram's fate should be sent back to the sender
        - second subfield - change
          - bit is set to 1 when the option data may be modified as the datagram is forwarded
        - third subfield - type subfield
  - two groups
    - Hop-by-Hop Options
      - those relevant to every router along a datagram's path
      - are the only ones that need to be processed by every router a packet encounters

    - Destination Options
        - those relevant only to the recipient
- 5.3.1.1 Pad1 and PadN
- IPv6 options are aligned to 8-byte offsets, so options that are naturally smaller are padded with 0 bytes to round out their lengths to the nearest 8-bytes
- Pad1
    - (type 0) is the only option that lacks Length and Value fields.
    - It is simply 1 byte long and contains the value 0
- PadN
    - (type 1) inserts 2 or more bytes of padding into the options area of the header using the format shown.
    - For n bytes of padding, the Opt Data Len field contains the value (n-2)
- 5.3.1.2 IPv6 Jumbo Payload
- In some TCP/IP networks, such as those used to interconnect supercomputers, the normal 64KB limit on the IP datagram size can lead to unwanted overhead when moving large amounts of data
- IPv6 Jumbo Payload option - specifies an IPv6 datagram with payload size larger than 65,535 bytes, called jumbogram.
    - this option need not be implemented by nodes attached to links with MTU sizes below 64KB
    - the jumbo payload option provides a 32-bit field for holding the payload size for datagrams with payloads of sizes between 65,535 and 4,294,967,295
- jumbogram
    - when formed for transmission, its normal Payload Length field is set to 0.
- when the jumbo payload option is used, TCP must be careful to use the length value from the option instead of the regular Length field in the base header
- larger payloads can lead to an increased chance of undetected error
- 5.3.1.3 Tunnel Encapsulation Limit
- Tunneling - refers to the encapsulation of one protocol in another that does not conform to traditional layering
    - example - IP datagrams may be encapsulated inside the payload portion of another IP datagram
- tunneling can be used to form virtual overlay networks
    - one network (e.g., the Internet) acts as a well-connected link layer for another layer of IP
- Tunnels can be nested in the sense that datagrams that are in a tunnel may themselves be placed in a tunnel, in a recursive fashion
- When sending an IP datagram, a sender does not ordinarily have much control over how many tunnel levels are ultimately used for encapsulation.
    - using this option, however, a sender can specify this limit
- A router intending to encapsulate an IPv6 datagram into a tunnel first checks for the presence and value of the Tunnel Encapsulation Limit option.
    - If the limit value is 0, the datagram is discarded and an ICMPv6 Parameter Problem message is sent to the source of the datagram
    - If the limit is nonzero, the tunnel encapsulation is permitted
        - the newly formed (encapsulating) IPv6 datagram must include a Tunnel Encapsulation Limit option whose value is 1 less than the option value in the arriving datagram
- 5.3.1.4 Router Alert
- option indicates that the datagram contains information that needs to be processed by a router

- 5.3.1.5 Quick-Start
- option is used in conjunction with the experimental Quick Start procedure for TCP/IP specified in …
- suggested for only private networks and not the global internet
- includes a value encoding the sender's desired transmission rate in bits per a second, a QS TTL value, and some additional information
- routers along the path may agree that supporting the desired rate is acceptable, in which case they decrement the QS TTL and leave the rate request unchanged when forwarding the containing datagram.
- When they disagree (i.e., wish to support a lower rate), they can reduce the number to an acceptable rate.
- Routers that do not recognize the QS option do not decrement the QS TTL.
- A receiver provides feedback to the sender, including the difference between the received datagram's IPv4 TTL or IPv6 Hop Limit field and its QS TTL, along with the resulting rate that may have been adjusted by the routers along the forwarded path.
  - This information is used by the sender to determine its sending rate
- Comparison of the TTL values is used to ensure that every router along the path participates in the QS negotiation
  - if any routers are found to be decrementing the IPv4 TTL field and not modifying the QS TTL value, QS is not enabled
- 5.3.1.6 CALIPSO
- option is used for supporting the Common Architecture Label IPv6 Security Option in certain private networks
- provides a method to label datagrams with a security-level indicator, along with some additional information
- intended for multilevel secure networking environments (e.g., government, military, and banking) where the security level of all data must be indicated by some form of label
- 5.3.1.7 Home Address
- option holds the "home" address of the IPv6 node sending the datagram when IPv6 mobility options are in use
- Mobile IP - specifies a set of procedures for handling IP nodes that may change their point of network attachment without losing their higher-layer network connections
  - it has a concept of a node's "home" which is derived from the address prefix of its typical location
  - when roaming away from home, the node is generally assigned a different IP address.
  - This option allows the node to provide its normal home address in addition to its (presumably temporarily assigned) new address while traveling.
  - the home address can be used by other IPv6 nodes when communicating with the mobile node.
- If the home address option is present, the destination options header containing it must appear after a Routing header and before the Fragment, Authentication, and ESP headers, if any of them is also present
- 5.3.2 Routing Header
- provides a mechanism for the sender of an IPv6 datagram to control, at least in part, the path the datagram takes through the network
- two different versions of the routing extension header have been specified
  - type 0 (RH0)
    - deprecated because of security concerns
  - type 2 (RH2)

- defined in conjunction with Mobile IP

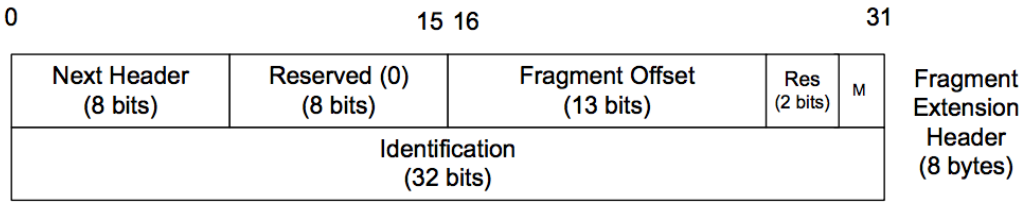
- RH0 header



- specifies one or more IPv6 nodes to be “visited” as the datagram is forwarded
- generalizes the loose Source and Record Route options from IPv4
- supports the possibility of routing identifies other than IPv6 addresses
- for standardized routing on IPv6 addresses RH0 allows the sender to specify a vector of IPv6 addresses for nodes to be visited
- The header contains an 8-bit Routing Type identifier and an 8-bit Segments Left field.
  - The type identifier for IPv6 addresses is 0 for RH0 and 2 for RH2
- Segments Left field
  - indicates how many route segments remain to be processed that is, the number of explicitly listed intermediate nodes still to be visited before reaching the final destination
- The block of address starts with a 32-bit reserved field set be the sender to 0 and ignored by receivers
  - the addresses are non multicast IPv6 address to be visited as the datagram is forwarded
- A routing header is not processed until it reaches the node whose address is contained in the Destination IP Address field of the IPv6 header
  - at this time, the Segments Left field is used to determine the next hop address from the address vector, and this address is swapped with the Destination IP Address field in the IPv6 header
  - Thus, as the datagram is forwarded, the segments Left field grows smaller and the list of addresses in the header reflects the node addresses that forwarded the datagram.
- Commands

- we can arrange to include a Routing header with a single command-line option to the ping6 command in Windows XP
  - C:\> ping6 -r -s 2001:db8::100 2001:db8::1
    - command arranges to use the source address 2001:db8::100 when sending a ping request to 2001:db8::1
    - -r option arranges for a Routing header (RH0) to be included
- RH0 has been deprecated because of a security concern that allows RH0 to be used to increase the effectiveness of DoS attacks
  - RH0 allows the same address to be specified in multiple locations within the Routing header
    - this can lead to traffic being forwarded many times between two or more hosts or routers along a particular path
    - The potentially high traffic loads that can be created along particular paths in the network can cause disruption to other traffic flows competing for bandwidth across the same path

- RH2 remains as the sole routing header supported by IPv6
  - equivalent to RH0 except it has room for only a single address and uses a different value in the Routing Type field
- 5.3.3 Fragment Header
- Fragment header is used by an IPv6 source when sending a datagram larger than the path MTU of the datagram's intended destination
- IPv4
  - any host or router can fragment a datagram if it is too large for the MTU on the next hop, and fields within the second 32-bit word of the IPv4 header indicate the fragmentation information
- IPv6
  - only the sender of the datagram is permitted to perform fragmentation, and in such cases a Fragment header is added
- Fragment header
  - includes the same information as found in the IPv4 header, but the Identification field is 32 bits instead of 16 bits that are used for IPv4



  - Reserved field
    - zero and ignored by receivers
  - 2-bit Res field
    - zero and ignored by receivers
  - Fragment Offset - field indicates where the data that follows the Fragment header is located, as a positive offset in 8-byte units, relative to the "fragmentable part" of the original IPv6 datagram
  - M bit field
    - if set to 1, indicates that more fragments are contained in the datagram

    - if set to 0, indicates that the fragment contains the last bytes of the original datagram
- The datagram serving as input to the fragmentation process is called the “original packet”
  - consists of two parts
    - “unfragmentable part”
      - includes the IPv6 header and any included extension headers required to be processed by intermediate nodes to the destination
    - “fragmentable part”
      - constitutes the remainder of the datagram (i.e., Destination options header, upper-layer headers, and payload data)
- when the original packet is fragmented, multiple fragment packets are produced, each of which contains a copy of the unfragmentable part of the original packet, but for which each IPv6 header has the Payload Length field altered to reflect the size of the fragment packet it describes.
- following the fragmentable part, each new fragment packet contains a Fragment header with an appropriately assigned Fragment Offset field (e.g. the first fragment contains offset 0)
- each fragment contains a copy of the original packet’s Identification field
- the last fragment has its M (More Fragments) bit field set to 0
- figure example
  - larger packet has been fragmented into three smaller packets, each containing a Fragment header
  - IPv6 headers Payload Length field is modified to reflect the size of the data and the newly formed fragment header
  - The fragment header in each fragment contains a common identification field
  - sender ensures that no distinct original packets are assigned the same field value within the expected lifetime of a datagram network
  - offset - field in the fragment header is given 8-byte units, so fragmentation is performed at 8-byte boundaries, which is why the first and second fragments contain 1448 data bytes instead of 1452
  - all but the last fragment is a multiple of 8 bytes
  - receiver must ensure that all fragments of an original datagram have been received before performing reassembly.
  - fragments may arrive out of order at the receiver but are reassembled in order to form a datagram that is given to other protocols for processing
- windows 7
  - construction of IPv6 fragment
  - ping -1 3952 ff01::2
  - figure 1
    - ping program generates ICMPv6 packets containing 3960 IPv6 payload bytes in the example
    - these packets are fragmented to produce three packet fragments, each of which is small enough to fit in the Ethernet MTU size of 1500 bytes
  - figure 2
    - second fragment of an ICMPv6 Echo Request contains 1448 IPv6 payload bytes including the 8-byte Fragment header
    - The presence of the Fragment header indicates that overall datagram was fragmented at the source
    - Offset field of 181 indicates that this fragment contains data starting at byte offset 1448
    - More Fragments - bit field being set indicates that other fragments are needed to reassemble the datagram

    - All fragments from the same original datagram contain the same Identification field
  - figure 3
    - the last fragment of the first ICMPv6 Echo Request datagram has an offset of 2896 and payload length of 1072 bytes
    - More Fragments - bit field being set to 0 indicates that this is the last fragment, and the original datagram's total payload length is 2896 + 1064 = 3960 bytes
- 5.4 IP Forwarding
- If the destination is directly connected to the host or on a shared network, the IP datagram is sent directly to the destination—a router is not required or used
  - otherwise, the host sends the datagram to a single router (called the default router) and lets the router deliver the datagram to its destination
- most hosts today can be configured to be routers as well as hosts, and many home networks use an Internet-connected PC to act as a router
- a host never forwards datagrams it does not originate, whereas routers do
- IP protocol can receive a datagram either from another protocol on the same machine (TCP, UDP, etc.) or from a network interface
- IP layer has some information in memory, usually called routing table or forwarding table, which it searches each time it receives a datagram to send
- when a datagram is received from a network interface, IP first checks if the destination IP address is one of its own IP addresses or some other address for which it should receive traffic such as as an IP broadcast or multicast address
  - if so the datagram is delivered to the protocol module specified by the Protocol field in the IPv4 header or Next Header field in the IPv6 header
- If the datagram is not destined for one of the IP addresses being used locally by the IP module
  - (1) if the IP layer was configured to act as a router, the datagram is forwarded (that is, handled as an outgoing datagram)
  - (2) the datagram is silently discarded
  - under some circumstances (e.g., no route is known in case 1) , an ICMP message may be sent back to the source indicating an error condition
- 5.4.1 Forwarding Table
- IP protocol standards do not dictate the precise data required to be in a forwarding table, as this choice is left up to the implementer of the IP protocol
- key pieces of information are generally required to implement the forwarding table for IP
  - each entry in the routing or forwarding table contains the following information fields, at least conceptually
    - Destination
      - contains a 32-bit field (128-bit field for IPv6) used for matching the result of a masking operation
      - destination can be as simple as zero, for a "default route" covering all destinations, or as long as the full length of an IP address, in the case of a "host route" that describes only a single destination
    - Mask
      - Contains a 32-bit field (128-bit field for IPv6) applied as a bitwise AND mask to the destination IP address of a datagram being looked up in the forwarding table
      - Masked result is compared with the set of destinations in the forwarding table entries
    - Next-hop
      - Contains the 32-bit IPv4 address or 128-bit IPv6 address of the next IP entity (router or host) to which the datagram should be sent

    - The next-hop entity is typically on a network shared with the system performing the forwarding lookup, meaning the two share the same network prefix
  - Interface
    - contains an identifier used by the IP layer to reference the network interface that should be used to send the datagram to its next hop.
    - For example
      - could refer to a host's 802.11 wireless interface
      - a wired Ethernet interface
      - PPP interface associated with a serial port
    - this field is used in selecting which source IP address to use on the outgoing datagram
- IP forwarding is performed on a hop-by-hop basis
- routers and hosts do not contain the complete forwarding path to any destination (except, of course, those destinations that are directly connected to the host or router)
- IP forwarding provides the IP address of only the next-hop entity to which the datagram is sent
- It is assumed that the next hop is really "closer" to the destination than the forwarding system is, and that the next-hop router is directly connected to the forwarding system
- also generally assumed that no "loops" are constructed between the next hops so that the datagram does not circulate around the network until its TTL or hop limit expires
- The job of ensuring correctness of the routing table is given to one or more routing protocols
- 5.4.2 IP Forwarding Actions
- When the IP layer in a host or router needs to send an IP datagram to a next-hop router or host, it first examines the destination IP address (D) in the datagram
- using the value D, the following longest prefix match algorithm is executed on the forwarding table

1. Search the table for all entries for which the following property holds: $(D \wedge m_j) = d_j$, where m_j is the value of the mask field associated with the forwarding entry e_j having index j, and d_j is the value of the destination field associated with e_j. This means that the destination IP address D is bitwise ANDed with the mask in each forwarding table entry (m_j), and the result is compared against the destination in the same forwarding table entry (d_j). If the property holds, the entry (e_j here) is a "match" for the destination IP address. When a match happens, the algorithm notes the entry index (j here) and how many bits in the mask m_j were set to 1. The more bits set to 1, the "better" the match.
2. The best matching entry e_k (i.e., the one with the largest number of 1 bits in its mask m_k) is selected, and its next-hop field n_k is used as the next-hop IP address in forwarding the datagram.

- If no matches in the forwarding table are found, the datagram is undeliverable
  - If the undeliverable datagram was generated locally (on this host), a "host unreachable" error is normally returned to the application that generated the datagram
- In some circumstances, more than one entry may match an equal number of 1 bits. This can happen, for example, when more than one default route is available
  - The end-system behavior in such cases is not set by standards and is instead specific to the operating system's protocol implementation
  - a common behavior is for the system to simply choose the first match

    - more sophisticated systems may attempt to load balance or split traffic across the multiple routes
- 5.4.3 Examples
- 5.4.3.1 Direct Delivery
- Example
    - our windows xp host (with IPv4 address S and MAC address S) which we will just call S, has an IP datagram to send to our Linux host (IPv4 address D, MAC address D), which we will call D.
    - These systems are interconnected using a switch
    - Both hosts are on the same Ethernet
    - When the IP layer in S receives a datagram to send from one of the upper layers such as TCP or UDP, it searches its forwarding table
    - The datagram is encapsulated in a lower-layer frame destined for the target host D
    - If the lower-layer address of the target host is unknown, the ARP protocol (IPv4) or Neighbor Solicitation (IPv6) operation may be invoked at this point to determine the correct lower layer address, D.
    - Once known, the destination address in the datagram is D's IPv4 address (10.0.0.9), and D is placed in the Destination IP Address field in the lower layer header
    - The switch delivers the frame to D based solely on the link-layer address D; it pays no attention to the IP addresses
- 5.4.3.2 Indirect Delivery
- Example
    - Windows host has an IP datagram to send to the host ftp.uu.net, whose IPv4 address is 192.48.96.9
    - First, the windows machine searches its forwarding table but does not find a matching prefix on the local network.
    - It uses its default route entry (which matches every destination, but with no 1 bits at all)
    - default entry indicates that the appropriate next-hop gateway is 10.0.0.1.
        - Typical scenario for a home network
    - IP addresses correspond to the source and destination hosts as before, but the lower-layer addresses do not
    - Lower-layer addresses determine which machines receive the frame containing the datagram on a per-hop basis
    - In this example the lower-layer address needed is the Ethernet address of the next-hop router R1's a-side interface, the lower-layer address corresponding to IPv4 address 10.0.0.1
        - this is accomplished by ARP on the network interconnection S and R1
    - Once R1 responds with its a-side lower-layer address, S sends the datagram to R1
    - Delivery from S to R1 takes place based on processing only the lower-layer headers (more specifically, the lower-layer destination address)
    - Upon receipt of the datagram, R1 checks its forwarding table
    - When R1 receives the datagram, it realizes that the datagram's destination IP address is not one of its won, so it forwards the datagram
    - Forwarding table is searched and the default entry is used.
    - The default entry in this case has a next hop within the ISP servicing the network, 70.231.159.254
    - This address happens to be within SBC's DSL network called by the some-what cumbersome name adsl-70-231-159-254.dsl.snfc21.sbcglobal.net

- Because the router is in the global Internet and the Windows machine's source address is the private address 10.0.0.100
- R1 performs Network Address Translation (NAT) on the datagram to make it routable on the internet
  - results in the datagram having a new source address 70.231.132.85
  - which corresponds to R1's b-side interface
- Networks that do not use private addressing avoid the last step and the original source address remains unchanged
- When R2 receives the datagram, it goes through the same steps that the local router R1 did (except for the NAT operation)
- If the datagram is not destined for one of its own IP addresses, the datagram is forwarded
- In this case the router usually has not only a default route but several others, depending on its connectivity to the rest of the Internet and its own local policies
- IPv6 forwarding varies only slightly from conventional IPv4 forwarding
- Aside from larger addresses IPv6 uses a slightly different mechanism (Neighbor Solicitation messages) to ascertain the lower-layer address of its next hop
- IPv6 has both link-local addresses and global addresses
- global addresses behave like regular IP addresses, link-local addresses can be used only on the same link
- because all the link-local addresses share the same IPv6 prefix (fe80::/10), a multihomed host may require user input to determine which interface to use when sending a datagram destined for a link local destination
- traceroute program
  - program lists each of the IP hops traversed while sending a series of datagrams to the destination ftp.uu.net (192.48.96.9)
  - The traceroute program uses a combination of UDP datagrams (with increasing TTL over time) and ICMP messages (used to detect each hop when the UDP datagrams expire) to accomplish its task
  - Three UDP packets are sent at each TTL value, providing there round-trip-time measurements to each hop
  - traceroute carries IP information
  - Multiprotocol Label Switching - form of link-layer network capable of carrying multiple network-layer protocols
    - use it for traffic engineering purposes

- **5.4.4 Discussion**
- Operation of IP unicast forwarding
  - (1)
    - Most of the hosts and routers in this example used a default route consisting of a single forwarding table entry of this form: mask 0, destination 0, next hop <some IP address>
    - Indeed, most hosts and most routers at the edge of the Internet can use a default route for everything other than destinations on local networks because there is only one interface available that provides connectivity to the rest of the Internet
  - (2)

    - the source and destination IP addresses in the datagram never change once in the regular Internet. This is always the case unless either source routing is used, or when other functions (such as NAT, as in the example) are encountered along the data path
    - Forwarding decisions at the IP layer are based on the destination address
  - (3)
    - A different lower-layer header is used on each link that uses addressing, and the lower-layer destination address (if present) always contains the lower-layer address of the next hop
    - Therefore, lower-layer headers routinely change as the datagram is moved along each hop toward its destination
    - Lower-layer addresses are normally obtained using ARP for IPv4 and ICMPv6 Neighbor Discovery for IPv6
- **5.5 Mobile IP**
- discussed conventional ways that IP datagrams are forwarded through the Internet, as well as private networks that use IP.
  - one assumption of the model is that host's IP address shares a prefix with its nearby hosts and routers
  - if such a host should move its point of network attachment, yet remain connected to the network at the link layer, all of its upper-layer (e.g. TCP) connections would fail because either its IP address would have to be changed or routing would not deliver packets to the (moved) host properly
- Mobile IP (MIPv6 more flexible and easier to explain deployed in the smartphone market)
  - based on the idea that a host has a "home" network but may visit other networks from time to time
  - while at home
    - ordinary forwarding is performed, according to the algorithms discussed in this chapter
  - when away from home
    - host keeps the IP address it would ordinarily use at home
      - special routing and forwarding tricks are used to make the host appear to the network, and to the other systems with which it communicates, as though it is attached to its home network
    - scheme depends on a special type of router called a "home agent" that helps provide routing for mobile nodes
- MIPv6
  - most of the complexity involves signaling messages and how they are secured
    - messages use various forms of the Mobility extension header
    - special protocol of its own
- **5.5.1 The Basic Model: Bidirectional Tunneling**
- mobile node (MN) - a host that might move
- correspondent nodes (CNs) - hosts that a mobile node might communicate with
- Home address (HoA) - The MN is given an IP address chosen from the network prefix used in its home network

- Care-of-address (CoA) - When the MN travels to a visited network, it is given additional address
- Home agent (HA) - traffic from communication between a CN and MN is routed through HA
  - special type of router deployed in the network infrastructure like other important systems (e.g., routers and Web servers)
- Binding for the MN - the association between an MN's HoA and its CoA
- Basic model works in cases where an MN's CNs do not engage in the MIPv6 protocol
  - this model is used for network mobility when an entire network is mobile
- binding update - message sent to the HA when the MN (or mobile network router) attaches to a new point in the network, and it receives it CoA
- binding acknowledgment - the response of the HA upon receiving the binding update message
- bidirectional tunneling - traffic between the MN and CNs is thereafter routed through the MN's HA using a two-way form of IPv6 packet tunneling
- Encapsulating Security Payload (ESP) - IPsec uses ESP to ordinarily protect the messages
  - doing so ensures that an HA is not fooled into accepting a binding update from a fake MN
- **5.5.2 Route Optimization (RO)**
- bidirectional tunneling makes MIPv6 work in a relatively simple way, and with CNs that are not Mobile-IP-aware
  - routing can be extremely inefficient, especially if the MN and CNs are near each other but far away from the MN's HA
- route optimization (RO) - can be used, provided it is supported by the various nodes
  - optimization to improve the inefficient routing that may occur in basic MIPv6
- when RO is used, involves a correspondent registration whereby an MN notifies its CNs of its current CoA to allow routing to take place without help from the HA
- RO operates in two parts
  - one part involves establishing and maintaining the registration bindings
  - another involves the method used to exchange datagrams once all bindings are in place
- Return Ratability Procedure (RRP) - To establish a binding with its CNs, an MN must prove to each CN that it is the proper MN
  - messages that support RRP are not protected using IPsec as are the messages between an MN and its HA
  - although RRP is not as strong as IPsec, it is simpler and covers most of the security threats of concern to the designers of Mobile IP
- RRP uses the following mobility messages, all of which are subtypes of the IPv6 Mobility extension header
  - Home Test Init (HoTI)
  - Home Test (HoT)
  - Care-of Test Init (CoTI)
  - Care-of Test (CoT)

  - These messages verify to a CN that a particular MN is reachable both at its home address (HoTI and HoT messages) and at its care-of addresses (CoTI and CoT messages)
- Figure
  - To understand RRP, we take the simplest case of a single MN, its HA, and a CN as shown
  - The MN begins by sending both a HoTI and CoTI message to the CN
    - The HoTI message is forwarded through the HA on its way to the CN
  - The CN receives both messages in some order and responds with a HoT and CoT message to each, respectively.
  - The HoT message is sent to the MN via the HA
  - Inside these messages are random bit strings called tokens, which the MN uses to form a cryptographic key
    - The key is then used to form authenticated binding updates that are sent to the CN
    - If successful, the route can be optimized and data can flow directly between an MN and a CN
  - Once a binding has been established successfully, data may flow directly between an MN and its CNs without the inefficiency of bidirectional tunneling
    - This is accomplished using an IPv6 Destination option for traffic moving from the MN to a CN
    - Type 2 routing header (RH2) for traffic headed in the reverse direction
  - The packets include a Source IP Address field of the MN's CoA, which avoids problems associated with ingress filtering that might cause packets containing the MN's HoA in the Source IP Address field to be dropped
  - The MN's HoA, contained in the Home Address option, is not processed by routers, so it passes through to the CN without modification
  - On the return path, packets are destined for the MN's CoA
  - After successfully receiving a returning packet, the MN processes the extension header and replaces the destination IP address with the HoA contained in the RH2
  - The resulting packet is delivered to the rest of the MN's protocol stack, so applications "believe" they are using the MN's HoA instead of its CoA for establishing connections and other actions
- **5.5.3 Discussion**
- Issues with Mobile IP
  - designed to address a certain type of mobility in which a node's IP address may change while the underlying link layer remains more or less connected
- usage model requiring Mobile IP is more likely to be a large number of smartphones that use IP
- **5.6 Host Processing of IP Datagrams**
- routers do not ordinarily have to consider which IP addresses to place in the Source IP Address and Destination IP Address fields of the packets they forward, hosts must consider both
- accept traffic destined for a local IP address if it arrives on the wrong interface?
- **5.6.1 Host Models**
- Strong host model

    - a datagram is accepted for delivery to the local protocol stack only if the IP address contained in the Destination IP Address field matches one of those configured on the interface upon which the datagram arrived.
- weak host model
    - a datagram carrying a destination address matching any of the local addresses may arrive on any interface and is processed by the receiving protocol stack, irrespective of the network interface upon which it arrived
- Host models also apply to sending behavior
    - a host using the strong host model sends datagrams from a particular interface only if one of the interface's configured addresses matches the Source IP Address field in the datagram being sent
- Attraction of strong host model relates to a security concern
    - negative consequences if applications make access control decisions based on the source IP address
- **5.6.2 Address Selection**
- when a host sends an IP datagram, it must decide which of its IP addresses to place in the Source IP Address field of the outgoing datagram
    - and which destination address to use for a particular destination host if multiple addresses for it are known
- source address selection - procedure for obtaining the Source IP address of the datagram
- destination address selection - procedure for obtaining the Destination IP Address
- selection of default addresses is controlled by a policy table, present in each host
    - It is a longest-matching prefix lookup table, similar to a forwarding table used with IP routing.
    - For an address A, a lookup in this table produces a precedence value for A, P(A), and a label for A, L(A)
        - A higher precedence value indicates greater preference
    - These labels are used for groupings of similar types
- The table or one configured at a site based upon administrative configuration parameters, is used to drive the address selection algorithm
- Notation
    - CPL(A,B) - "common prefix length" is the length, in bits, of the longest common prefix between IPv6 addresses A and B, starting from the left-most-significant bit
    - S(A) - the scope of IPv6 address A mapped to a numeric value with larger scopes mapping to larger values
        - If A is link-scoped and B is global scope, then S(A) < S(B)
    - M(A) - maps an IPv4 address A to an IPv4-mapped IPv6 address
    - ^(A) - is the lifecycle of the address
        - delta(A) < delta(B) if A is a deprecated address (i.e., one whose use is discouraged) and B is a preferred address (i.e., an address preferred for active use)
    - Mobile IP notation only
        - H(A) is true if A is a home address
        - C(A) is true if A is a care-of address
- **5.6.2.1 The Source Address Selection Algorithm**

- CS(D) - candidate set of potential source addresses based on a particular destination address D
  - anycast, multicast, and the unspecified address are never in CS(D) for any D
- R(A) - indicate the rank of address A in the set CS(D)
  - R(A) > R(B) - means that A is preferred to B for use as a source address for reaching the machine with address D
  - R(A) *> R(B) - means to assign A a higher rank than B in CS(D)
- I(D) - indicates the interface selected (i.e., by the forwarding longest matching prefix algorithm described previously) to reach destination D
- @(i) - is the set of addresses assigned to interface i
- T(A) - is the Boolean true if A is a temporary address and false otherwise
- following rules are applied to establish a partial ordering between addresses A and B in CS(D) for destination D

1. Prefer same address: if A = D, R(A) *> R(B); if B = D, R(B) *> R(A).
2. Prefer appropriate scope: if S(A) < S(B) and S(A) < S(D), R(B) *> R(A) else R(A) *> R(B); if S(B) < S(A) and S(B) < S(D), R(A) *> R(B) else R(B) *> R(A).
3. Avoid deprecated addresses: if S(A) = S(B), { if Λ(A) < Λ(B), R(B) *> R(A) else R(A) *> R(B) }.
4. Prefer home address: if H(A) and C(A) and ¬(C(B) and H(B)), R(A) *> R(B); if H(B) and C(B) and ¬(C(A) and H(A)), R(B) *> R(A); if (H(A) and ¬C(A)) and (¬H(B) and C(B)), R(A) *> R(B); if (H(B) and ¬C(B)) and (¬H(A) and C(A)), R(B) *> R(A).
5. Prefer outgoing interface: if A ∈ @(I(D)) and B ∈ @(I(D)), R(A) *> R(B); if B ∈@(I(D)) and A ∈ @(I(D)), R(B) *> R(A).
6. Prefer matching label: if L(A) = L(D) and L(B) ≠ L(D), R(A) *> R(B); if L(B) = L(D) and L(A) ≠ L(D), R(B) *> R(A).
7. Prefer nontemporary addresses: if T(B) and ¬T(A), R(A) *> R(B); if T(A) and ¬T(B), R(B) *> R(A).
8. Use longest matching prefix: if CPL(A,D) > CPL(B,D), R(A) *> R(B); if CPL(B,D) > CPL(A,D), R(B) *> R(A).

- The partial ordering rules can be used to form a total ordering of all the candidate addresses in CS(D)
- Q(D) - the one with the largest rank, and is used by the destination selection algorithm
- Q(D) = emptySet(null) , no source could be determined for destination D.
- **5.6.2.2 The Destination Address Selection Algorithm**
- Q(D) - source address selected to reach the destination D
- U(B) - be the Boolean true if destination B is not reachable

- E(A) - indicate that destination A is reached using some “encapsulating transport” (e.g., tunneled routing.)
- following rules

1. Avoid unusable destinations: if U(B) or Q(B) = Ø, R(A) *> R(B); if U(A) or Q(A) = Ø, R(B) *> R(A).
2. Prefer matching scope: if S(A) = S(Q(A)) and S(B) ≠ S(Q(B)), R(A) *> R(B); if S(B) = S(Q(B)) and S(A) ≠ S(Q(A)), R(B)*>R(A).
3. Avoid deprecated addresses: if Λ (Q(A)) < Λ (Q(B)), R(B) *> R(A); if Λ (Q(B)) < Λ (Q(A)), R(A) *> R(B).
4. Prefer home address: if H(Q(A)) and C(Q(A)) and ¬(C(Q(B)) and H(Q(B))), R(A) *> R(B); if (Q(B)) and C(Q(B)) and ¬(C(Q(A)) and H(Q(A))), R(B) *> R(A); if (H(Q(A)) and ¬C(Q(A))) and (¬H(Q(B)) and C(Q(B))), R(A) *> R(B); if (H(Q(B)) and ¬C(Q(B))) and (¬H(Q(A)) and C(Q(A))), R(B) *> R(A).
5. Prefer matching label: if L(Q(A)) = L(A) and L(Q(B)) ≠ L(B), R(A) *> R(B); if L(Q(A)) ≠ L(A) and L(Q(B)) = L(B), R(B) *> R(A).
6. Prefer higher precedence: if P(A) > P(B), R(A) *> R(B); if P(A) < P(B), R(B) *> R(A).
7. Prefer native transport: if E(A) and ¬E(B), R(B) *> R(A); if E(B) and ¬E(A), R(A) *> R(B).
8. Prefer smaller scope: if S(A) < S(B), R(A) *> R(B) else R(B) *> R(A).
9. Use longest matching prefix: if CPL(A, Q(A)) > CPL(B, Q(B)), R(A) *> R(B); if CPL(A, Q(A)) < CPL (B, Q(B)), R(B) *> R(A).
10. Otherwise, leave rank order unchanged.

- these rules form a partial ordering between two elements of the set of possible destinations in the set of destinations SD(S) for source S
- highest-rank address gives the output for the destination address selection algorithm
- **5.7 Attacks Involving IP**
- Without authentication or encryption, IP spoofing attacks are possible
  - ingress filtering - an ISP checks the source addresses of its customers’ traffic to ensure that datagrams contain source addresses from an assigned IP prefix
- IPv6 and Mobile IP are relatively new compared to IPv4
  - all of their vulnerabilities have undoubtedly not yet been discovered
  - with the newer more flexible types of options headers, an attacker could have considerable influence on the processing of an IPv6 packet
- **5.8 Summary**
- IPv4 and IPv6 headers

  - discussing some of the related functions such as the Internet checksum and fragmentation
- IPv6
  - increases the size of addresses
  - improves upon IP's method of including options in packets by use of the extension headers
  - removes several of the noncritical fields from the IPv4 header
  - with the addition of this functionality, the IP header increases in size by only a factor of 2 even through the size of the addresses has increased fourfold
- IPv4 and IPv6 not directly compatible and share only the 4-bit Version field in common
- IP forwarding
  - describes the way IP datagrams are transported through single and multihop networks
  - performed on a hop-by-hop basis unless special processing takes place
  - destination IP address never changes as the datagram proceeds through all the hops, but the link-layer encapsulation and destination link-layer address change on each hop
- Forwarding tables and the longest prefix match algorithm
  - are used by hosts and routers to determine the best matching forwarding entry and determine the next hops along a forwarding path
  - in many circumstances, very simple tables consisting of only a default route, which matches all possible destinations equally, are adequate
- Mobile IP
  - using a special set of protocols for security and signaling Mobile IP establish secure bindings between a mobile node's home address and care-of address
  - these bindings may be used to communicate with a mobile node even when it is not at home
  - basic function involves tunneling traffic through a cooperating home agent, but this may lead to very inefficient routing
  - route optimization - feature allows a mobile node to talk directly with other remote nodes and vice versa
    - requires a mobile node's correspondent hosts to support MIPv6 as well as route optimization
- host model
  - strong or weak affects how IP datagrams are processed
  - Strong model
    - each interface is permitted to receive or send datagrams that use addresses associated with the interface
  - Weak model
    - less restrictive
    - permits communication in some cases where it would not otherwise be possible
      - may be vulnerable to certain kinds of attacks
  - host model also relates to how a host chooses which addresses to use when communicating
    - a set of address selection algorithms, for both source and destination addresses, was presented

- algorithms tend to prefer limited scope and permanent addresses