SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

AN AUTONOMOUS INSTITUTION

Accredited by NBA–AICTE and Accredited by NAAC–UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

Department of Information Technology

19IT503 – Internet of Things

UNIT – 4 IPv6 TECHNOLOGIES FOR THE IOT

Motivations

Internet Protocol Version 6 (IPv6) is a newer version of the network layer protocol that is designed to coexist (but not directly interwork) with IPv4. In the long term, IPv6 is expected to replace IPv4, but that will not happen overnight. IPv6 provides improved internetworking capabilities compared to what is presently available with IPv4.

Challenges in IPv4
The current IPv4 has been in use for over 30 years, but it exhibits some challenges in supporting emerging demands for

- address space cardinality,
- high-density mobility,
- multimedia, and
- strong security.

IPv6 offers the potential of achieving scalability, reachability, end-to-end interworking, quality of service (QoS), and commercial grade robustness that is needed for contemporary and emerging web services, data services, mobile video, and Internet of things (IoT) applications.

Need for IPv6
IP was designed as a packet-based technology (protocol) in the late 1970s–early 1980s for the purpose of connecting computers that were in separate geographic locations.

Starting in the early 1990s, developers realized that the communication needs of the twenty-first century needed a protocol with some new features and capabilities, while at the same time retaining the useful features of the existing protocol.

IPv6 was initially developed in the early 1990s because of the anticipated need for more end-system addresses based on anticipated Internet growth, encompassing mobile phone deployment, smart home appliances, and billions of new users in developing countries.

Technologies and applications such as voice over IP (VoIP), "always-on access" (e.g., cable modems), broadband and/or ethernet-to-the-home, converged networks, evolving ubiquitous computing applications, and IoT will be driving this need even more in the next few years.

While the basic function of the network layer internetworking protocol is to move information

across networks, IPv6 has more capabilities built into its foundation than IPv4.

ADDRESS CAPABILITIES

IPv4 Addressing and Issues

- The current IPv4 naming scheme was developed in the 1970s and had capacity for about 4.3 billion addresses, which were grouped into 255 blocks of 16 million addresses each.

- In IPv4, addresses consist of four octets. With IPv4, the 32-bit (4 byte) address can be represented as AdrClass|netID|hostID. The network portion can contain either a network ID or a network ID and a subnet.

- Since the IPv4 address has 32 bits, there are nominally 2^{32} different IP addresses (as noted, approximately 4.3 billion nodes, if all combinations are used).

- Hence, there are 4,294,967,296 unique values, which can be considered as a sequence of 256 "/8s," where each "/8" corresponds to 16,777,216 unique address values.

- The problem is that during the 1980s, many public, registered addresses were allocated to firms and organizations without any consistent control.

- As a result, organizations have more addresses that they actually might need, giving rise to the present dearth of available "registerable" layer 3 addresses. Furthermore, not all IP addresses can be used due to the fragmentation described above. Therefore many address spaces are unused.

- Even worse, rise of population and rise of connected devices on internet such as mobile, tab are ever increasing, it creates address dearth.

- A temporary and pragmatic approach to alleviate the dearth of addresses, network address translation (NAT) mechanisms are employed by organizations and even home users.

- NAT is used to alleviate this issue by map the internal addresses to an external public address when the private-to-public network boundary is crossed

- This, however, imposes a number of limitations, particularly since the number of registered public addresses available to a company is almost invariably much smaller (as small as 1) than the number of internal devices requiring an address.

- A number of protocols cannot travel through a NAT device, and hence the use of NAT implies that many applications (e.g., VoIP, Videoconferencing, video-ondemand/IPTV) cannot be used effectively in all instances.

- The need for obligatory use of NAT disappears with IPv6.

IPv6 Address Space

One of the major modifications in the addressing scheme in IPv6 is a change to the basic types of addresses and how they are utilized.

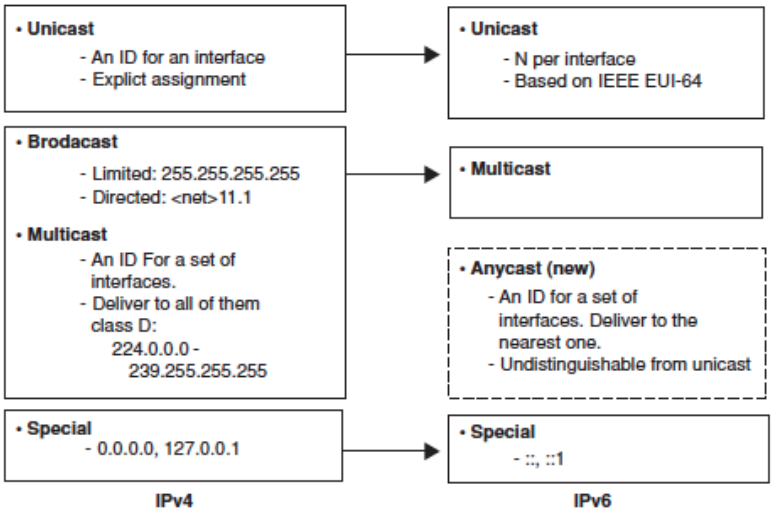
- Unicast addresses are utilized for a majority of traditional (enterprise) communications, as was the case in IPv4.

- Broadcast as a specific addressing type has been eliminated; in its place support for multicast addressing has been expanded and made a required part of the protocol.

- A new type of addressing called anycast has also been implemented. In addition, there are a number of special IPv6 addresses.

Logically, one can interpret the types of transmissions as follows3:

- Unicast transmission: "send to this one specific address"
- Multicast transmission: "send to every member of this specific group"
- Anycast transmission: "send to any one member of this specific group." Typically (motivated by efficiency goals), the transmission occurs to the closest (in routing terms) member of the group. Generally one interprets anycast to mean "send to the closest member of this specific group.



Address scheme

IPv6 address consists of 128 bits, rather than 32 bits as with IPv4 addresses; the number of bits correlates to the address space are

IP Version	Size of Address Space
IPv6	128 bits, which allows for 2^{128} or 340,282,366,920,938,463,463,374,607,431,768,211,456 (3.4×10^{38}) possible addresses
IPv4	32 bits, which allows for 2^{32} or 4,294,967,296 possible addresses

The use of 128 bits provides multiple levels of hierarchy and flexibility in designing hierarchical addressing and routing. The IPv4-based Internet currently lacks this flexibility.

The IPv6 address is represented as eight groups of 16 bits each, separated by the ":" character. Each 16-bit group is represented by 4 hexadecimal digits, that is, each digit has a value between 0 and f (0,1, 2, . . . a, b, c, d, e, f with a = 10, b = 11, and so on, to f = 15).

Example Address of IPv6

3223:0ba0:01e0:d001:0000:0000:d0f0:0010

An abbreviated format exists to designate IPv6 addresses when all endings are 0. For example

3223:0ba0:: is the abbreviated for the above.

Similarly, only one 0 is written, removing 0's in the left side, and four 0's in the middle of the address. For example the address

3223:ba0:0:0:0:0::1234

Special IPv6 addresses, as follows

- Auto-return or loopback virtual address. This address is specified in IPv4 as the 127.0.0.1 address. In IPv6, this address is represented as ::1.
- Not specified address (::). This address is not allocated to any node since it is used to indicate absence of address.
- IPv6 over IPv4 dynamic/automatic tunnel addresses. These addresses are designated as IPv4-compatible IPv6 addresses and allow the sending of IPv6 traffic over IPv4 networks in a transparent manner. They are represented as, for example, ::156.55.23.5.
- IPv4 over IPv6 addresses automatic representation. These addresses allow for IPv4-only nodes to still work in IPv6 networks. They are designated as "mapped from IPv4 to IPv6 addresses" and are represented as ::FFFF:, for example ::FFFF.156.55.43.3.

IPv6 Protocol Overview

IPv6 is a connectionless datagram protocol used primarily for addressing and routing packets between hosts.

- Connectionless means that a session is not established before exchanging data.
- Unreliable means that delivery is not guaranteed.
- IPv6 always makes a best-effort attempt to deliver a packet. An IPv6 packet might be lost, delivered out of sequence, duplicated, or delayed.

IPv6 basic protocol capabilities include the following:

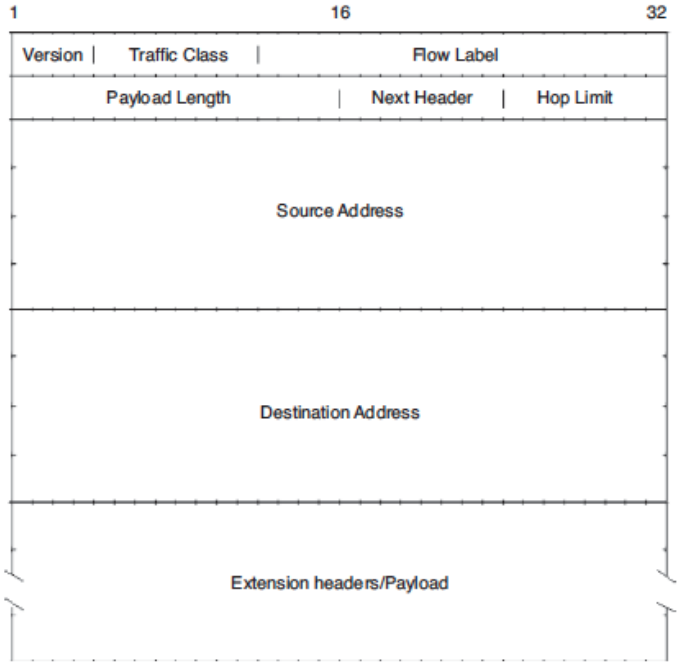
- Addressing
- Anycast
- Flow Labels
- ICMPv6
- Neighbor discovery (ND)

Packet format

An IPv6 packet, also known as an IPv6 datagram, consists of an IPv6 header and an IPv6 payload.

The IPv6 header consists of two parts, the IPv6 base header and optional extension headers.

Functionally, the optional extension headers and upper-layer protocols, for example TCP, are considered part of the IPv6 payload.



Changes from IPv4 to IPv6 Packet format

- The IP header in IPv6 has been streamlined and defined to be of a fixed length 40 bytes
- In IPv6, header fields from the IPv4 header have been removed, renamed, or moved to the new optional IPv6 extension headers.
- The IPv4 “type of service” is equivalent to the IPv6 “traffic class” field. The “total length” field has been replaced with the “payload length” field.
- The functionality provided by the “time to live (TTL4)” field has been replaced with the “hop limit” field.
- The “protocol” field has been replaced with the “next header type” field.
- The “header checksum” field was removed, which has the main advantage of not having each relay spend time processing the checksum.
- The “options” field is no longer part of the header as it was in IPv4. Options are specified in the optional IPv6 extension headers.

Fields in IPv6 Packet

- Every packet has IPv6 header and an IPv6 payload. The IPv6 header consists of two parts, the IPv6 base header and optional extension headers
- The header size is fixed 40 bytes.
- Version – 4 bits - Indicates version of IP which is 6

- Traffic class - 8 bits - Differentiated Services and Priority field used for Explicit congestion notification.
- Flow Label - 20 bits - Defines how traffic is handled and identified.
- Payload length – 16 bits - The size of the payload in octets, including any extension headers, as well as the upper-layer Protocols.
- Next Header – 8 bits - Identifies the header immediately following the IPv6 header (Extension header). Whereas In some cases it indicates the protocols contained within upper-layer packet, such as TCP, UDP.
- Hop limit – 8 bits - Identifies the number intermediate nodes on which the packet is allowed to travel before being discarded by a router. The hop limit is set by the sending host and is used to prevent packets from endlessly circulating on an IPv6 internetwork.

The IPv6 specification defines a number of extension headers

- Routing header—Similar to the source routing options in IPv4. The header is used to mandate a specific routing.
- AH—A security header that provides authentication and integrity.
- Encapsulating security payload (ESP) header—A security header that provides authentication and encryption.
- Fragmentation header—The Fragmentation Header is similar to the fragmentation options in IPv4.
- Destination options header—Header that contains a set of options to be processed only by the final destination node. MIPv6 is an example of an environment that uses such a header.
- Hop-by-hop options header—A set of options needed by routers to perform certain management or debugging functions.

New Features of IPv6

Auto Configuration (Plug and Play)

- Auto-configuration is a new characteristic of the IPv6 protocol that facilitates network management and system set-up tasks by users.

- Auto-configuration facilitates initialization of user devices: after connecting a device to an IPv6 network, one or several IPv6 globally unique addresses are automatically allocated.

- With this new feature IPv6 node generates addresses without the use of a DHCP for IPv6 (DHCPv6) server.

- The stateless auto-configuration protocol does not require a server component because there is no state to maintain.

- Stateless auto-configuration is also described as serverless. The acronym SLAAC is also used; it expands to stateless address auto-configuration.

- The host generates its own address using a combination of the information that it possesses and the information that is supplied by the router.

- To ensure that allocated addresses (granted either by manual mechanisms or by auto-configuration) are unique in a specific link, the link duplicated address detection algorithm is used. It removes duplicate address.

IPv6 Tunneling

- IPv6 tunneling is a technique for establishing a "virtual link" between two IPv6 nodes for transmitting data packets as payloads of IPv6 packets.

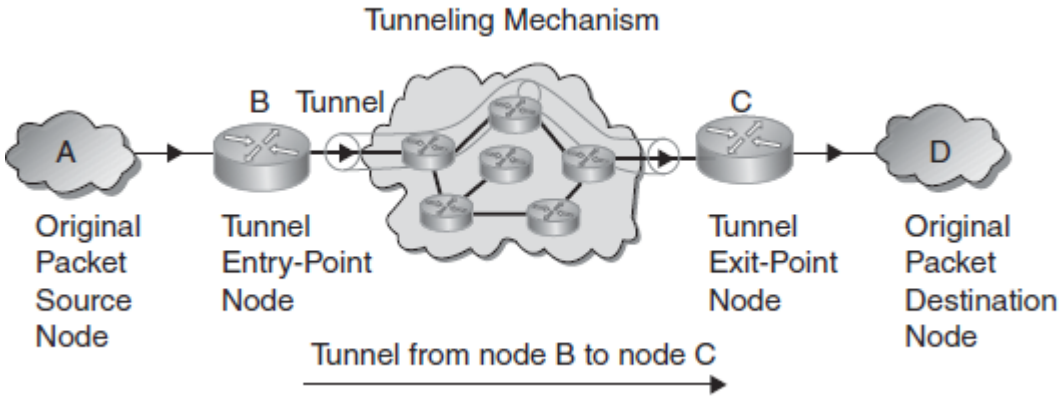
- From the perspective of the two nodes, this "virtual link," called an IPv6 tunnel , appears as a point-to-point link on which IPv6 acts like a link-layer protocol.

- The two IPv6 nodes support specific roles. One node encapsulates original packets received from other nodes or from itself and forwards the resulting tunnel packets through the tunnel.

- The other node decapsulates the received tunnel packets and forwards the resulting original packets toward their destinations, possibly itself.

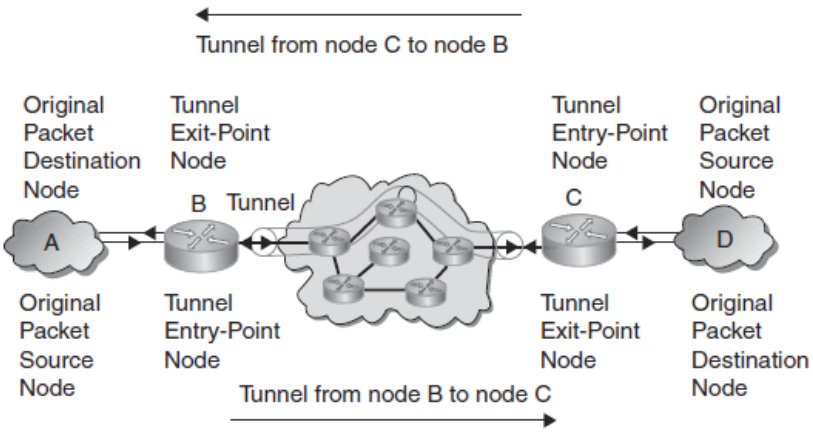
- The encapsulator node is called the tunnel entry-point node, and it is the source of the tunnel packets. The decapsulator node is called the tunnel exit point, and it is the destination of the tunnel packets.

- An IPv6 tunnel is a unidirectional mechanism—tunnel packet flow takes place in one direction between the IPv6 tunnel entry-point and exit-point nodes.



Tunneling - Unidirectional

- Bidirectional tunneling is achieved by merging two unidirectional mechanisms, that is, configuring two tunnels, each in opposite direction to the other—the entry-point node of one tunnel is the exit-point node of the other tunnel.



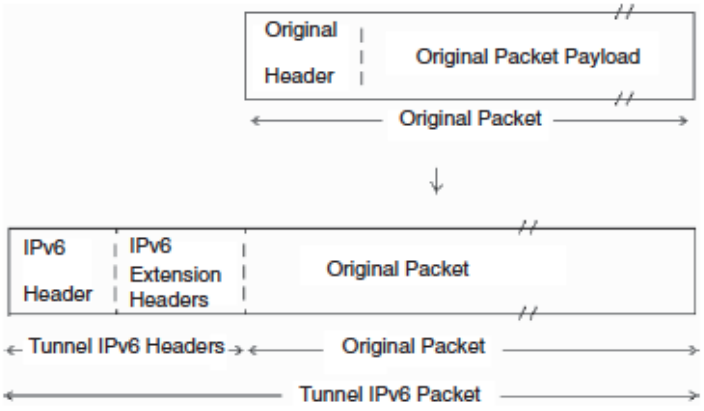
Tunneling - Bidirectional

Tunneling Mechanism

- IPv6 encapsulation entails prepending an IPv6 header to the original packet, and, optionally, a set of IPv6 extension headers, that are collectively called tunnel IPv6 headers.

- The encapsulation takes place in an IPv6 tunnel entry-point node, as a result of an original packet being forwarded onto the virtual link represented by the tunnel. The original packet is processed during forwarding according to the forwarding rules of the protocol of that packet.

- At encapsulation, the source field of the tunnel IPv6 header is filled with an IPv6 address of the tunnel entry-point node and the destination field with an IPv6 address of the tunnel exit point. Subsequently, the tunnel packet resulting from encapsulation is sent toward the tunnel exit-point node.



Encapsulation of Original Packet into another IPv6 Packet

- Upon receiving an IPv6 packet destined to an IPv6 address of a tunnel exit-point node, its IPv6 protocol layer processes the tunnel headers.

- The tunnel exit-point node, which decapsulates the tunnel packets, and the destination node, which receives the resulting original packets.

IPsec IN IPv6

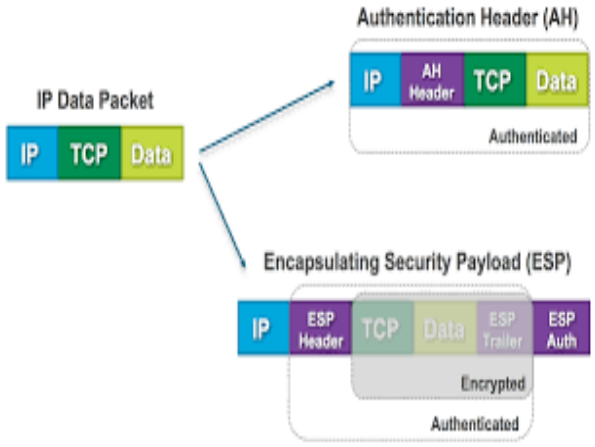
The IP security (IPSec) is an Internet Engineering Task Force (IETF) standard suite of protocols between 2 communication points across the IP network that provide data authentication, integrity, and confidentiality.

IPsec is a group of protocols that are used together to set up encrypted connections between devices. It helps keep data sent over public networks secure.

IPsec is often used to set up VPNs, and it works by encrypting IP packets, along with authenticating the source where the packets come from.

IPsec provides network-level security where the application data is encapsulated within the IPv6 packet. IPsec itself is a set of two protocols:

- ESP, which provides integrity and confidentiality and AH, which provides integrity.
- IPsec utilizes the AH and/or ESP header to provide security.

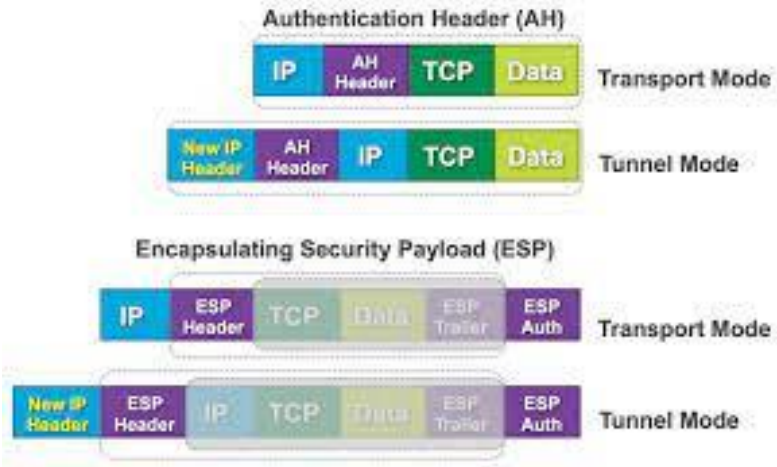


IPsec, with ESP, offers integrity and data origin authentication, confidentiality, and optional anti-replay features; in addition, ESP provides limited traffic flow confidentiality.

Both the AH and ESP header may be employed as follows

- Tunnel mode - The protocol is applied to the entire IP packet. This method is needed to ensure security over the entire packet, where a new IPv6 header and an AH or ESP header are wrapped around the original IP packet.

- Transport mode—The protocol is just applied to the transport layer (i.e., TCP, UDP, ICMP) in the form of an IPv6 header and AH or ESP header, followed by the transport protocol data (header, data).



Header Compression Schemes

Implementation of IPv6 gives rise to concerns related to expanded packet headers. The packet header size doubled from 20 bytes in IPv4 to at least 40 bytes in IPv6.

The use of network-layer encryption mechanism nearly doubles IP operational overhead. Header Compression scheme is, therefore, of interest because of expanded packet headers.

Currently, the use of HC in commercial networks is generally rare, but wireless and video applications (especially in an IPv6 environment) may well drive future deployment of the technology.

HC algorithms can reduce the performance and throughput impact of expanded IPv6 packet headers and protocol-imposed overhead.

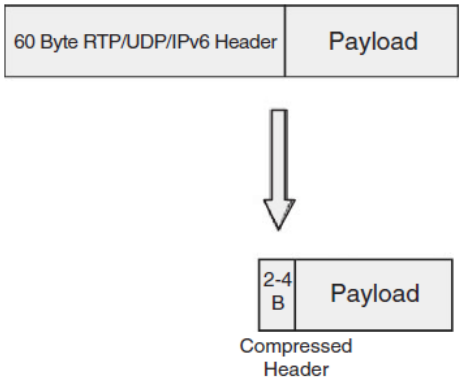


FIGURE 7.9 HC for IPv6.

Traditionally, compression is applied to layer 3 (IP) and several layer 4 protocol headers; for example, RTP/UDP/IPv6 headers can be compressed from 60 bytes to 2–4 bytes.

HC algorithms can also reduce the additional overhead introduced by network-layer encryption mechanisms (e.g., IPsec).

Compression algorithms that address encryption/decryption have the ability to:
(i) compress inner headers before encryption and
(ii) compress outer ESP/IP headers after encryption.

Two compression protocols emerged from the IETF in recent years

- Internet protocol header compression (IPHC)
  A scheme designed for low bit error rate (BER) links. It provides compression of TCP/IP, UDP/IP, RTP/UDP/IP, and ESP/IP header; enhanced compression of RTP/UDP/IP (ECRTP) headers is also used.

- Robust header compression (ROHC)
  It is a scheme designed for wireless links that provides greater compression compared to IPHC at the cost of greater implementation complexity. This is more suitable for high BER, long RTT links and supports compression of ESP/IP, UDP/IP, and RTP/UDP/IP header.

Compression is applied over a link between a source node (i.e., compressor) and a destination node (i.e., decompressor). HC algorithms make use of protocol inter-packet header field redundancies to improve overall efficiency.

Both compressor and decompressor store header fields of each packet stream and associate each stream with a context identifier (CID).

Upon reception of a packet with an associated context, the compressor removes the IPv6 header fields from packet header and appends a CID.

Upon reception of a packet with a CID, the decompressor inserts IPv6 header fields back into packet header and transmits packet.

Quality of Service in IPv6

QoS is supported in IPv6. The IPv6 header has two QoS-related fields:

20-bit flow label, usable in IntServ-based environments. In IntServ environments, performance guarantees to traffic and resource reservations are provided on per-flow basis. A guaranteed and controlled load service capability is supported. IntServ approaches have scalability issues;

8-bit traffic class indicator usable in DiffServ-based environments. DiffServ environments are more common. The traffic class field may be used to set specific precedence or differentiated services code point (DSCP) values. DiffServ classifies all the network traffic into classes. Two distinct types (per hop behaviors) are supported:

Expedited forwarding (EF): aims at providing QoS for the class by minimizing jitter and is generally focused on providing stricter guarantees;

Assured forwarding (AF): inserts at most four classes with at most three levels of packets dropping categories.

There are no signaling protocol for resource allocation (admission control) and QoS mechanisms control. The following priority levels are typical, but variances are possible:

- Level 0—No specify priority
- Level 1—Background traffic (news)
- Level 2—Unattended data transfer (email)
- Level 3—Reserved
- Level 4—Attended bulk transfer (FTP)
- Level 5—Reserved
- Level 6—Interactive traffic (Telnet, Windowing)
- Level 7—Control traffic (routing, network management)

MOBILE IPv6

MIPv6 specifies a protocol that allows nodes to remain reachable while moving around in the IPv6 Internet. An entity that implements the MIPv6 protocol is a MIPv6 entity.

Mobile node (MN)

- A mobile node is a handheld equipment with roaming capabilities. It can be a cell phone, a personal digital assistant (PDA), a laptop, etc.

Home network

- The home network of a mobile device is the network within which the device receives its identifying IP address (home address). In other words, a home network is a subnet to which a mobile node belongs to as per its assigned IP address. Within the home network, there is no need of mobile IP.

Home agent (HA)

- The HA stores information about all mobile nodes whose permanent home address is in the network assigned to the HA. The HA maintains a location directory of the mobile handsets belonging permanently to the home network, and acts as a router for delivery of datagrams to the MH, when it is away from home.

Correspondent node (CN):

- A peer node with which an MN is communicating. The CN may be either mobile or stationary. A CN does not necessarily require MIPv6 support, but it does require IPv6 support.

Foreign agent (FA)

- The foreign agent is a router in a foreign network that functions as the point of attachment for a mobile node (MN) when it roams to the foreign network. The packets from the home agent are sent to the foreign node which delivers it to mobile node.

Foreign network

- The foreign network is the current subnet to which the mobile node is visiting. Itis different from home network. In other words, a foreign network is the network in which a mobile node is operating when away from its home network.

Care-of-address (COA)

- It is an address that identifies the mobile node's current location. The packets sent to the MN are delivered to COA. COA is typically associated with the mobile node's foreign agent (FA).

Key Mechanism

Mobile IP is associated with the following three basic mechanisms

- Discovering the care-of-address
- Registering the care-of-address
- Tunneling to the care-of-address

Discovering the care-of-address

Each mobile node uses a discovery protocol to identify the respective home and foreign agents. The discovery of the care-of-address consists of the following important steps.

1.Mobile agents advertise their presence by periodically broadcasting the agent advertisement messages.

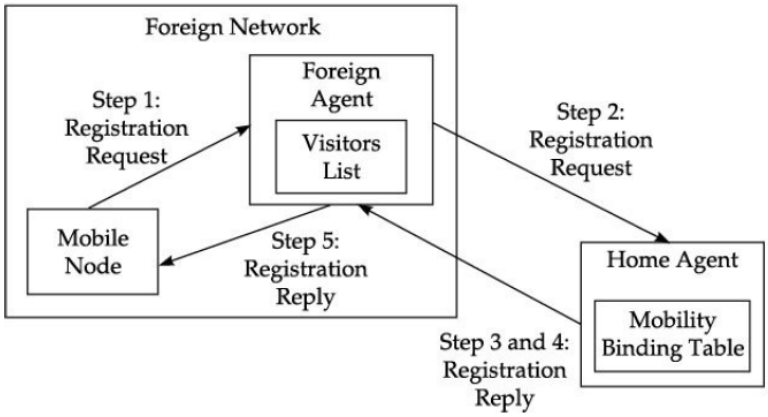
Foreign agents send messages to advertise the available care-of addresses.

2. The mobile node receiving the agent advertisement message observes whether the message is from its own home agent and determines whether it is on the home network or on a foreign network.

3. If a mobile node does not wish to wait for the periodic advertisement, it can send out agent solicitation messages that will be responded to by a mobility agent.

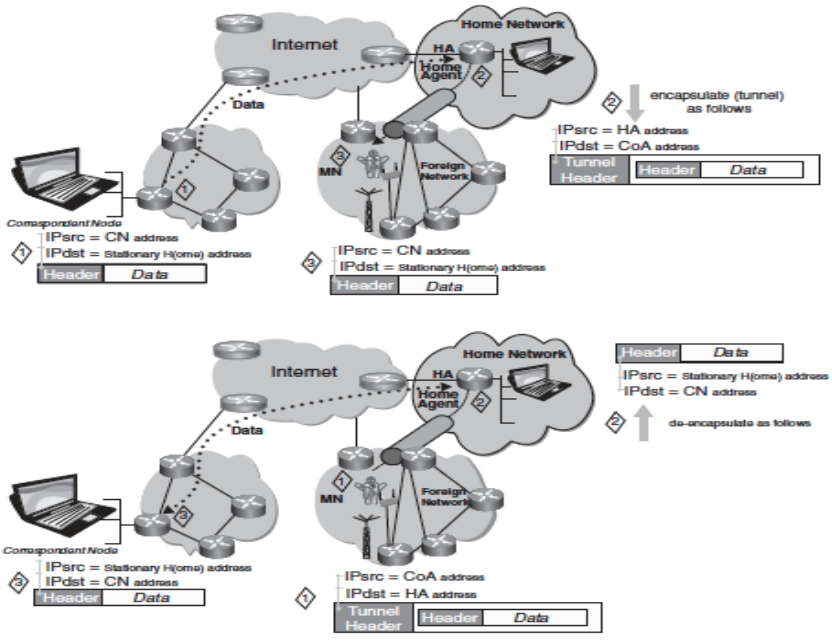
Registering the care-of-address

- While a node has moved to a different network, if the mobile node obtains a care-of-address from a foreign agent, then this address should be registered with the home agent.
- If the MN travels to a FN, it registers with the FA by sending a registration request message which includes the permanent IP address of the mobile host and the IP address of its HA.
- The FA in turn performs the registration process on behalf of the mobile host by sending a Registration Request containing the permanent IP address of the MN and the IP address of the FA (COA) to the HA.
- When the HA receives the Registration Request, it updates the mobility binding by associating the COA of the MN with its home address.
- The HA then sends an acknowledgement to the FA.
- The FA in turn updates its visitors list by inserting the entry for the MN and relays the reply to the mobile node.



Tunnelling to the care-of-address

- Tunnelling takes place to forward an IP datagram from the home agent to a care-of-address. This involves carrying out the following steps:
- When a home agent receives a packet addressed to a mobile host, it forwards the packet to the care-of-address using IP-within-IP (encapsulation).
- Using IP-within-IP, the home agent inserts a new IP header in front of the IP header of any datagram.
- Destination address is set to the care-of-address.
- Source address is set to the home agent's address.
- After stripping out the first header, IP processes the packet again.



Tunnelling to the care-of-address

Two approaches are used for tunneling to COA. They are Bidirectional tunneling and direct tunneling.

Bidirectional tunneling

In this approach, the HA plays a crucial role, although this implies that the network traffic to this node can be high; however, the CN has no requirements related to mobility support—also, the

MNs have no direct visibility related to the CN.

Direct routing (aka route optimization)

In the mobile IP protocol, all the data packets to the mobile node go through the home agent. Because of this there will be heavy traffic between HA and CN in the network, causing latency to increase. Therefore, the following route optimization needs to be carried out to overcome this problem.

- Enable direct notification of the corresponding host
- Direct tunnelling from the corresponding host to the mobile host
- Binding cache maintained at the corresponding host

TABLE 4.1 Messages Transmitted in Optimized Mobile IP

Message type	*Description*
1. Binding request	If a node wants to know the current location of a mobile node (MN), it sends a request to home agent (HA).
2. Binding acknowledgement	On request, the node will return an acknowledgement message after getting the binding update message.
3. Binding update	This is a message sent by HA to CN mentioning the correct location of MN. The message contains the fixed IP address of the mobile node and the care-of-address. The binding update can request for an acknowledgement.
4. Binding warning	If a node decapsulates a packet for a mobile node (MN), but it is not the current foreign agent (FA), then this node sends a binding warning to the home agent (HA) of the mobile node (MN).

Protocol Details

IPv6 Protocol Extensions

MIPv6 defines a new IPv6 protocol, using the mobility header.

TABLE 8.4 Mobility Header Messages

Message	Description
HoTi HoT Care-of test init Care-of test	These messages are used to perform the return-routability procedure from the MN to a CN
BU	Message is used by an MN to notify a CN or the MN's HA of its current binding. The BU sent to the MN's HA to register its primary CoA is marked as a "home registration"
BA	Message is used to acknowledge receipt of a BU, if an acknowledgement was requested in the BU, the BU was sent to an HA, or an error occurred
BRR	Message is used by a CN to request an MN to re-establish its binding with the CN. This message is typically used when the cached binding is in active use, but the binding's lifetime is close to expiration. The CN may use, for instance, recent traffic and open transport layer connections as an indication of active use
Binding error	Message is used by the CN to signal an error related to mobility, such as an inappropriate attempt to use the home address destination option without an existing binding

New IPv6 ICMP Messages

MIPv6 also introduces four new ICMPv6 message types, two for use in the dynamic HAAD mechanism and two for renumbering and mobile configuration mechanisms.

- HAAD request. The ICMP HAAD request message is used by an MN to initiate the dynamic HAAD mechanism. The MN sends the HAAD request message to the MIPv6 HA anycast address for its own home subnet prefix.
- HAAD reply. The ICMP HAAD reply message is used by an HA to respond to an MN that uses the dynamic HAAD mechanism.
- Mobile prefix solicitation. The ICMP mobile prefix solicitation message is sent by an MN to its HA while it is away from home. The purpose of the message is to solicit a mobile prefix advertisement from the HA, which will allow the MN.
- Mobile prefix advertisement. An HA will send a mobile prefix advertisement to an MN to distribute prefix information about the home link while the MN is traveling away from the home network.

Mobile IPv6 Security

MIPv6 incorporates a number of security features.

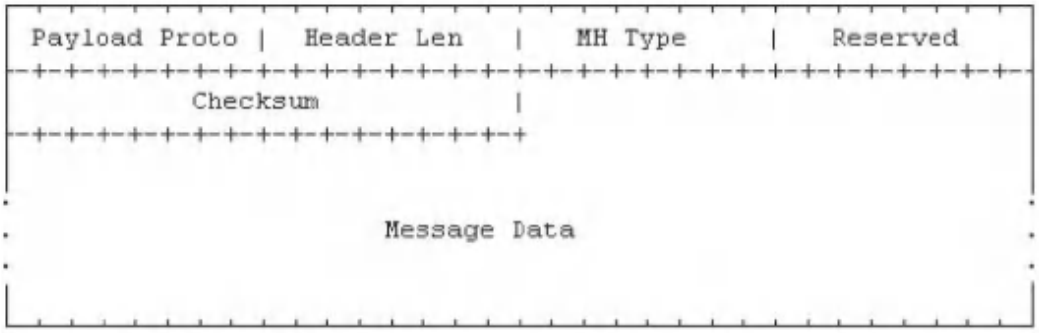
These include the protection of BUs both to HAs and to CNs, the protection of mobile prefix discovery, and the protection of the mechanisms that MIPv6 uses for transporting data packets:

- BUs are protected by the use of IPsec extension headers, or by the use of the binding authorization data option (this option employs a binding management key, Kbm, which can be established through the return-routability procedure).
- Mobile prefix discovery is protected through the use of IPsec extension headers.
- Mechanisms related to transporting payload packets—such as the home address destination option and type 2 routing header—have been specified in a manner that restricts their use in attacks.

New IPv6 Protocol, Message Types, and Destination Option

Mobility Header

The mobility header is an extension header used by MNs, CNs, and HAs in all messaging related to the creation and management of bindings.



Mobility Header Fields

Payload Proto	8-bit selector. Identifies the type of header immediately following the mobility header. Uses the same values as the IPv6 next header field. This field is intended to be used by a future extension.
Header Len	8-bit unsigned integer, representing the length of the mobility header in units of 8 octets, excluding the first 8 octets.
MH Type	8-bit selector. Identifies the particular mobility message in question.
Reserved	8-bit field reserved for future use. The value must be initialized to zero by the sender and must be ignored by the receiver.
Checksum	16-bit unsigned integer. This field contains the checksum of the mobility header.
Message Data	A variable length field containing the data specific to the indicated mobility header type.

Mobility Message Type

Binding refresh request (BRR) Message	The BRR message requests a mobile node to update its mobility binding. This message is sent by correspondent nodes. The BRR message uses the MH Type value 0.
Home test init (HoTI) message	A mobile node uses the HoTI message to initiate the return-routability procedure and request a home keygen token from a correspondent node. The Home test init message uses the MH type value 1. This message is tunneled through the home agent when the mobile node is away from home. Such tunneling should employ IPsec ESP in tunnel mode between the HA and the mobile node. This protection is indicated by the IPsec security policy database.
Care-of test init (CoTI) message	A mobile node uses the CoTI message to initiate the return-routability procedure and request a care-of keygen token from a correspondent node. The Care-of test init message uses the MH type value 2.
Home test (HoT) message	The HoT message is a response to the Home test init message and is sent from the correspondent node to the mobile node. The HoT message uses the MH type value 3.
Care-of test (CoT) message	The CoT message is a response to the CoT Init message and is sent from the correspondent node to the mobile node. The CoT message uses the MH type value 4.
Binding update (BU) message	The BU message is used by a mobile node to notify other nodes of a new CoA for itself. The BU uses the MH type value 5.
Binding acknowledgement (BA) message	The BA is used to acknowledge receipt of a BU. The BA has the MH type value 6.
Binding error (BE) message	The BE message is used by the correspondent node to signal an error related to mobility, such as an inappropriate attempt to use the home address destination option without an existing binding. The BE message uses the MH type value 7.

Mobility Message Type

Two important messages are the BU message and the BA message.

Binding Update (BU)

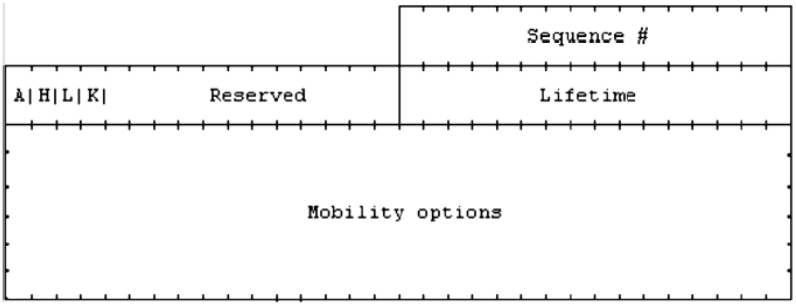
The BU message is used by an MN to notify other nodes of a new CoA it has acquired.

Acknowledge (A) - The acknowledge (A) bit is set by the sending MN to request a BA be returned upon receipt of the BU.

Home registration (H) - The home registration (H) bit is set by the sending MN to request that the receiving node should act as this node's HA.

Link-local address compatibility (L) – This bit is set when the home address reported by the MN has the same interface identifier as the MN's link-local address.

Key management mobility capability (K) - If this bit is cleared, the protocol used for establishing the IPsec SAs between the MN and the HA does not survive movements; it may then have to be rerun.



Reserved - These fields are unused. They must be initialized to zero by the sender and must be ignored by the receiver.

Sequence number - A 16-bit unsigned integer used by the receiving node to sequence BUs and by the sending node to match a returned BA with this BU.

Lifetime. 16-bit unsigned integer. The number of time units remaining before the binding must be considered expired. A value of zero indicates that the binding cache entry for the MN must be deleted.

Mobility options. Variable-length field of such length that the complete mobility header is an integer multiple of 8 octets long. This field contains zero or more Type/Length/Value (TLV)-encoded mobility options.

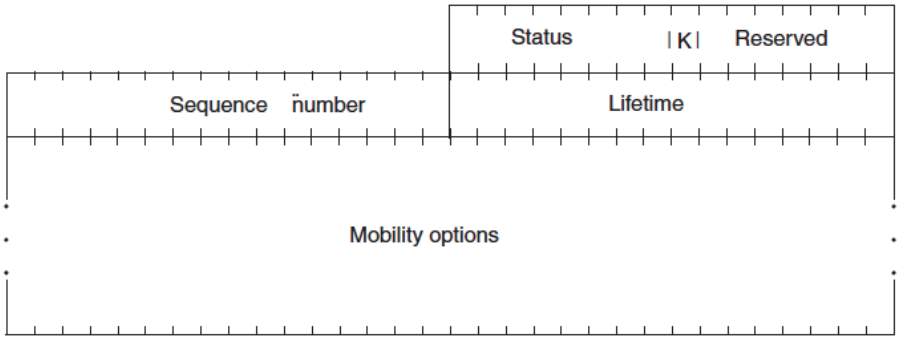
Binding Acknowledgement

The BA message is used to acknowledge the receipt of a BU.

Key management mobility capability (K) - If this bit is cleared, the protocol used by the HA for establishing the IPsec SAs between the MN and the HA does not survive movements (it may then have to be rerun).

Reserved - These fields are unused. They must be initialized to zero by the sender and must be ignored by the receiver.

Status - 8-bit unsigned integer indicating the disposition of the BU. Values of the status field less than 128 indicate that the BU was accepted by the receiving node. Values greater than or equal to 128 indicate that the BU was rejected by the receiving node.



Sequence number - The sequence number in the BA is copied from the sequence number field in the BU. It is used by the MN in matching this BA with an outstanding BU.

Lifetime - The granted lifetime, in time units of 4s, for which this node should retain the entry for this MN in its binding cache.

Mobility options. Variable-length field of such length that the complete mobility header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options.

Modifications to IPv6 Neighbor Discovery

Existing Protocols modified with the following options

- Modified Router Advertisement Message
- Modified Prefix Information Option
- New Advertisement Interval Option
- New HA Information Option
- Changes to Sending Router Advertisements

Modified Router Advertisement Message

- MIPv6 modifies the format of the router advertisement message by the addition of a single flag bit to indicate that the router sending the advertisement message is serving as an HA on this link.

Modified Prefix Information Option

- MIPv6 requires knowledge of a router's global address in building an HA list as part of the dynamic HAAD mechanism.
- MIPv6 extends neighbor discovery defined in RFC 2461 to allow a router to advertise its global address by the addition of a single flag bit in the format of a prefix information option for use in router advertisement messages.

New Advertisement Interval Option

- MIPv6 defines a new advertisement interval option, used in router advertisement messages to advertise the interval at which the sending router sends unsolicited multicast router advertisements.

New HA Information Option

- MIPv6 defines a new HA information option, used in router advertisements sent by an HA to advertise information specific to this router's functionality as an HA.

Changes to Sending Router Advertisements

- The basic NDP specification limits routers to a minimum interval of 3s between sending unsolicited multicast router advertisement messages from any given network interface.
- This option used for faster movement detection, that is to increase the rate at which unsolicited router advertisements are sent.
- MIPv6 may send unsolicited multicast router advertisements more frequently.

Requirements for Various IPv6 Nodes

MIPv6 imposes specific requirements on the functions provided by different types of IPv6 nodes

IPv6 nodes with support for route optimization

- The node must be able to validate a home address option using an existing binding cache entry.
- The node should be able to interpret ICMP messages.
- The node must be able to send Binding Error messages.
- The node must be able to process Mobility Headers.
- The node must be able to participate in a return-routability procedure.
- The node must be able to process BU messages.
- The node must be able to return a BA.
- The node must be able to maintain a Binding Cache of
- the bindings received in accepted BUs.

IPv6 routers

- Every IPv6 router should be able to send an advertisement interval option in each of its router advertisements, to aid movement detection by MNs.
- The use of this option in router advertisements should be configurable
- Every IPv6 router should be able to support sending unsolicited multicast router advertisements at a fast rate (the used rate should then be configurable)
- Each router should include at least one prefix with the router address (R) bit set and with its full IP address in its router advertisements

IPv6 routers that serve as an HA

- Every HA must be able to maintain an entry in its binding cache for each MN for which it is serving as the HA
- Every HA must be able to intercept packets (using proxy neighbor discovery) addressed to an MN for which it is currently serving as the HA, on that MN's home link, while the MN is away from home
- Every HA must be able to encapsulate such intercepted packets in order to tunnel them to the primary CoA for the MN indicated in its binding in the HA's binding cache
- Every HA must support decapsulating reverse tunneled packets sent to it from an MN's home address.
- Every HA must also check that the source address in the tunneled packets corresponds to

the currently registered location of the MN
- The node must be able to process mobility headers.
- Every HA must be able to return a BA in response to a BU
- Every HA must maintain a separate HA list for each link on which it is serving as an HA

IPv6 MNs
- The node must maintain a BU list
- The node must support sending packets containing a home address option and follow the required Ipsec interaction
- The node must be able to perform IPv6 encapsulation and decapsulation
- The node must be able to process type 2 routing header
- The node must support receiving a binding error message
- The node must support receiving ICMP errors
- The node must support movement detection, CoA formation, and returning home
- The node must be able to process mobility headers
- The node must support the return-routability procedure
- The node must be able to send BUs
- The node must be able to receive and process BAs
- The node must support receiving a BRR by responding with a BU
- The node must support receiving mobile prefix advertisements and reconfiguring its home address based on the prefix information contained therein
- The node should support use of the dynamic HAAD mechanism
- The node must allow route optimization to be administratively enabled or disabled. The default should be enabled

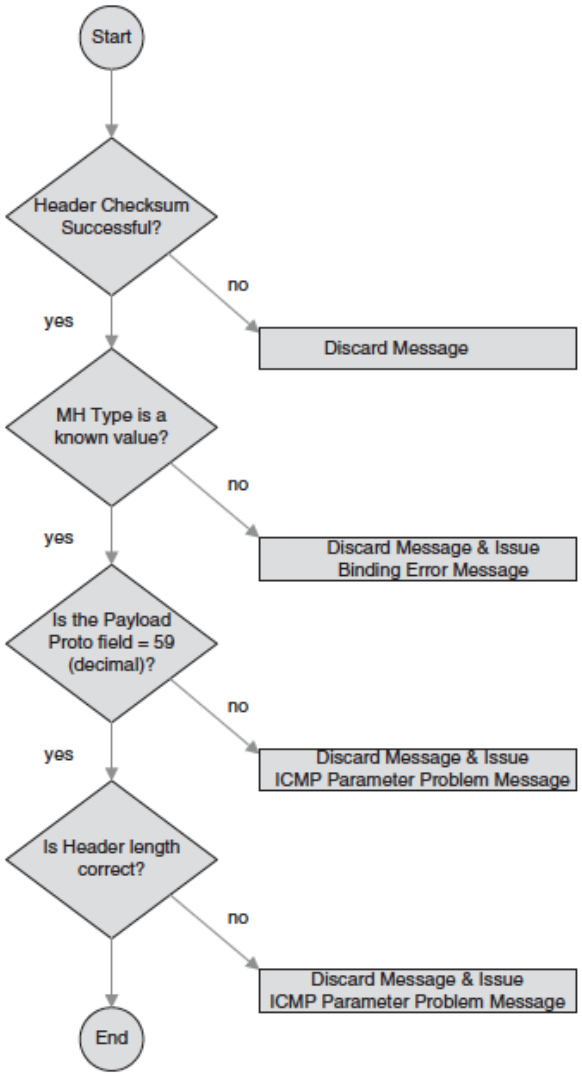
Correspondent Node Operation

CNs are required to support the following functionality:
- Processing mobility headers
- Packet processing
- Return-routability procedure
- Processing bindings
- Cache replacement policy

Processing mobility headers
Mobility header processing follows the process of Figure

Packet Processing Packet processing covers the following subactivities:
- Receiving packets with home address option
- Sending packets to an MN
- Sending binding error messages
- Receiving ICMP error messages

Receiving packets with home address option. The CN must process the option in a manner consistent with exchanging the home address field from the home address option into the IPv6 header and replacing the original value of the source address field there.



Processing mobility headers

Sending packets to an MN. Before sending any packet (except when sending an IPv6 neighbor discovery packet), the sending node should examine its binding cache for an entry for the destination address to which the packet is being sent.

Sending binding error messages. A binding error message is sent directly to the address that appeared in the IPv6 source address field of the offending packet (if the source address field does not contain a unicast address, the binding error message must not be sent).

Receiving ICMP error messages. When the CN has a binding cache entry for an MN, all traffic destined to the MN goes directly to the current CoA of the MN using a routing header. Any ICMP error message caused by packets on their way to the CoA will be returned in the normal manner to the CN.

Return-Routability Procedure

TABLE 8.7 Return-Routability Actions of the CN

Action	Description
Receiving HoTi messages	Upon receiving a HoTi message, the CN verifies that the packet does not include a home address destination option. Any packet carrying a HoTi message that fails to satisfy all of these tests must be silently ignored. Otherwise, in preparation for sending the corresponding HoT message, the CN checks that it has the necessary material to engage in a return-routability procedure. The CN must have a secret Kcn and a nonce; it does not have this material yet, it must produce it before continuing with the return-routability procedure
Receiving care-of test init messages	Upon receiving a HoTi message, the CN verifies that the packet does not include a home address destination option. Any packet carrying a care-of test init message that fails to satisfy all of these tests must be silently ignored. Otherwise, in preparation for sending the corresponding care-of test message, the CN checks that it has the necessary material to engage in a return-routability procedure
Sending HoT messages	The CN creates a home keygen token and uses the current nonce index as the home nonce index; it then creates a HoT message and sends it to the MN at the latter's home address
Sending care-of test messages	The CN creates a care-of keygen token and uses the current nonce index as the care-of nonce index; it then creates a care-of test message and sends it to the MN at the latter's CoA

Processing Bindings Messages related to bindings are as follows:
Receiving BUs. Before accepting a BU, the receiving node must validate the BU.

Requests to cache a binding. There is a need to process a valid BU that requests a node to cache a binding, for which the home registration (H) bit is not set in the BU. In this case, the receiving node should create a new entry in its binding cache for this home address, or update its existing binding cache entry for this home address, if such an entry already exists.

Requests to delete a binding. There is a need to process a valid BUthat requests a node to delete a binding when the home registration (H) bit is not set in the BU. Any existing binding for the given home address must be deleted. A binding cache entry for the home address must not be created in response to receiving the BU.

Sending BAs. A BA may be sent to indicate receipt of a BU. If the node accepts the BU and creates or updates an entry for this binding, the status field in the BA must be set to a value less than 128. Otherwise, the status field must be set to a value greater than or equal to 128.

Sending binding refresh requests (BRRs). If a binding cache entry being deleted is still in active use when sending packets to an MN, then the next packet sent to the MN will be routed normally to the MN's home link. Communication with the MN continues, but the tunneling from the home network creates additional overhead and latency in delivering packets to the MN.

Cache Replacement Policy

A node may maintain a separate timer for each entry in its binding cache. When creating or updating a binding cache entry in response to a received and accepted BU, the node sets the timer for this entry to the specified lifetime period; entries in a node's binding cache are deleted after the expiration of the lifetime specified in the BU from which the entry was created or last updated.

HA Node Operation

HA operations entail the following functions:

Maintaining the binding cache and the HA list
Processing mobility headers

Processing bindings
◦ Primary CoA registration
◦ Primary CoA de-registration

Packet processing
◦ Intercepting packets for an MN
◦ Processing intercepted packets
◦ Multicast membership control
◦ Stateful Address autoconfiguration
◦ Handling reverse tunneled packets
◦ Protecting return-routability packets

Dynamic HAAD
Sending prefix information to the MN