2. IPv6 – advanced functionalities

PA159: Net-Centric Computing I.

Eva Hladká

Faculty of Informatics Masaryk University

Autumn 2010

Lecture Overview I

1. Lecture Overview

2. Brief IPv6 Introduction
   - IPv6 Addresses
   - Path MTU discovery

3. IPv6 Neighbor Discovery Protocol in Detail
   - L2 address resolution
   - Duplicate Address Detection
   - Neighbor Unreachability Detection
   - Autoconfiguration
   - Summary

4. IPv6 Mobility Support in Detail
   - Return Routability Procedure

Lecture overview I

1. Lecture Overview
2. Brief IPv6 Introduction
   - IPv6 Addresses
   - Path MTU discovery
3. IPv6 Neighbor Discovery Protocol in Detail
   - L2 address resolution
   - Duplicate Address Detection
   - Neighbor Unreachability Detection
   - Autoconfiguration
   - Summary
4. IPv6 Mobility Support in Detail
   - Return Routability Procedure

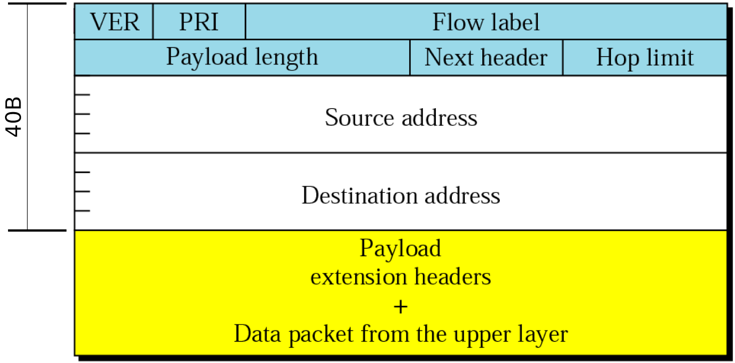
IP Protocol version 6 (IPv6) – Why a new protocol?

- *the master pulse for a new protocol proposal:* relatively fast exhaustion of IPv4 address space
- further reasons: the issues, that arose during IPv4 usage, especially:
  - weak support of real-time applications
  - no support of communication security
  - no devices' autoconfiguration support
  - no mobility support
  - etc.
- (many features retroactively implemented into IPv4)

IP Protocol version 6 (IPv6) – Basic Features

- *bigger address space* – 128-bit IPv6 address, theoretically 2^{128} of unique addresses
- *simpler header format* – basic 40B header containing just the most necessary information
- *possibilities of further extensions* – through so-called *extension headers*
- *support of real-time transmissions* – streams' tagging and priorities
- *support of secure communication* – authentication, encryption and integrity verification support
- *mobility support* – using so-called *home agents*
- *devices' autoconfiguration support* – stateful and stateless autoconfiguration

IPv6 Datagram – Basic Header



40B

VER | PRI | Flow label

Payload length | Next header | Hop limit

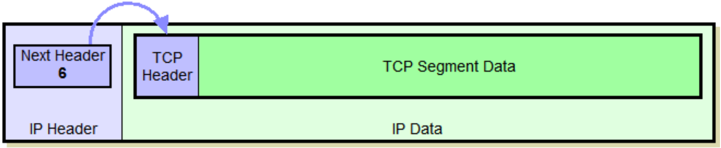
Source address

Destination address

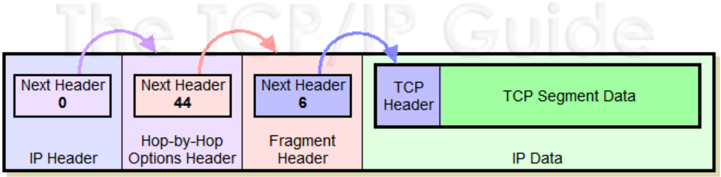
Payload
extension headers
+
Data packet from the upper layer

- fixed basic header size (40 B)
- checksum, options, and fragmentation information not included in basic header
  - options and fragmentation information has to be ensured via extension headers
  - checksum was removed at all (it's ensured on L2 and L4)

IPv6 Datagram – Extension Headers



IPv6 Datagram With No Extension Headers Carrying TCP Segment



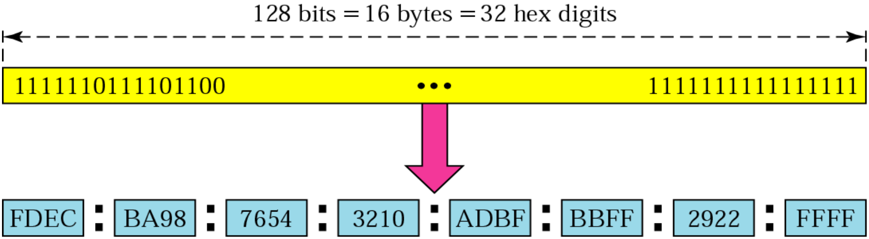
IPv6 Datagram With Two Extension Headers Carrying TCP Segment

Several extension headers have been defined

- e.g., Hop-By-Hop Options, Routing, Fragment, Encapsulating Security Payload, Authentication Header, etc.

IPv6 Addresses

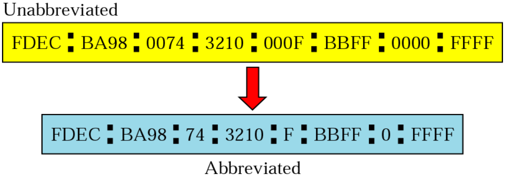
- (currently) final solution to address space shortage
- IPv6 address has 128 bits (= 16 bytes):
  - 2^{128} of unique addresses ($\approx 3 \times 10^{38}$ addresses $\Rightarrow \approx 5 \times 10^{28}$ addresses for every human on the Earth)
  - written in a hexadecimal form instead of decadic (pairs of bytes divided by ":" character)


128 bits = 16 bytes = 32 hex digits
1111110111101100
•••
1111111111111111
FDEC : BA98 : 7654 : 3210 : ADBF : BBFF : 2922 : FFFF

IPv6 addresses – Address Abbreviation

The leading 0s might be omitted in each address group:

- 0074 might be written as 74, 000F as F, . . .
- 3210 **cannot** be abbreviated!



Unabbreviated

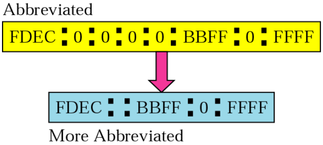
FDEC:BA98:0074:3210:000F:BBFF:0000:FFFF

FDEC:BA98:74:3210:F:BBFF:0:FFFF

Abbreviated

Consecutive groups of zeros might be replaced by "::" character)

- just **a single group** might be replaced!



Abbreviated

FDEC:0:0:0:0:BBFF:0:FFFF

FDEC::BBFF:0:FFFF

More Abbreviated

IPv6 addresses – Hierarchy

- the goal is to simplicity the routing
- the structure of unicast IPv6 addresses is defined by RFC 3587
- basic structure:

n bits	**64-n bits**	**64 bits**
global routing prefix	subnet address	interface address

- global routing prefix ≈ network address
- subnetwork address is usually 16 bits long ⇒ global routing prefix thus has 48 bits
  - first 16 bits contain the value 2001 (hexadecimal form)
  - next 16 bits are assigned by Regional Internet Registry (RIR)
  - next 16 bits are assigned by Local Internet Registry (LIR)

16 bits	**16 bits**	**16 bits**	**16 bits**	**64 bits**
2001	assigned by RIR	assigned by LIR	subnet address	interface address

IPv6 Addresses & CIDR

- IPv6 addresses are just *classless* (classes do not exist)
- IPv6 networks are defined using CIDR notation (similarly as in the IPv4 case)
- e.g., *FDEC:0:0:0:0:BBFF:0:FFFF/60*

IPv6 addresses – address types

- *unicast addresses* – same as in IPv4 (a single network interface identification)
- *multicast addresses* – same as in IPv4, used for addressing a group of devices/hosts
  - data is delivered to all the group members
  - prefix `ff00::/8`
- *anycast addresses* – a newbie in IPv6
  - identify a group of devices/hosts as well
  - but data is delivered just to a single member of the group (the closest one)

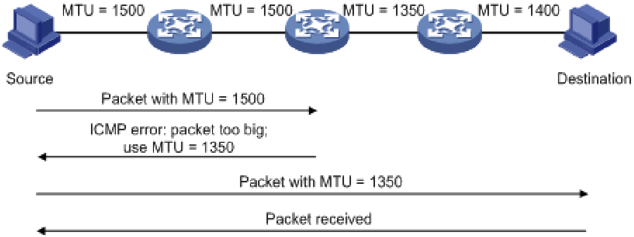
- IPv4 broadcast addresses are not used in IPv6
  - replaced by special multicast addresses (e.g., `FF02::1` – all the nodes on the particular LAN)

IPv6 Path MTU discovery

- just source devices must decide on the correct size of fragments
    - routers can't fragment datagrams, just end nodes can!
    - if a datagram is too large for a router, it must drop the datagram
        - and send back to the source a feedback about this occurrence (in the form of an ICMPv6 *Packet Too Big* message)
- **Path MTU Discovery**
    - a special technique used for determining what size of fragments should be used
    - uses the feedback mechanism performed by ICMPv6 *Packet Too Big* messages
        - the source node sends a datagram that has the MTU of its local physical link (it represents an upper bound on the MTU)
        - if this goes through without any errors, that value for future datagrams to that destination can be used
        - if it gets back any *Packet Too Big* messages, it tries again using a smaller datagram size (indicated in the Packet Too Big message)

IPv6 Path MTU discovery

The Schema



Lecture overview I

1. Lecture Overview
2. Brief IPv6 Introduction
   - IPv6 Addresses
   - Path MTU discovery
3. IPv6 Neighbor Discovery Protocol in Detail
   - L2 address resolution
   - Duplicate Address Detection
   - Neighbor Unreachability Detection
   - Autoconfiguration
   - Summary
4. IPv6 Mobility Support in Detail
   - Return Routability Procedure

Neighbor Discovery Protocol I.

- How can we obtain a link (e.g., Ethernet) address of a node (having its IP address)?
  - IPv4: ARP protocol
  - IPv6: a new mechanism called *Neighbor Discovery Protocol* proposed
- *Neighbor Discovery for IP version 6* (RFC 2461)
  - a part of ICMPv6
  - in comparison with the IPv4's ARP, new functionalities has been added
  - IPv6 nodes use Neighbor Discovery for/to:
    - autoconfiguration of IPv6 address (stateful/stateless autoconfiguration)
    - determine network prefixes, routers and other configuration information
    - duplicate IP address detection (DAD)
    - determine layer two addresses of nodes on the same link
    - find neighboring routers that can forward their packets
    - keep track of which neighbors are reachable and which are not (NUD)
    - detect changed link-layer address

Neighbor Discovery Protocol II.

- consists of five ICMP messages:
  - Router Solicitation (RS)
  - Router Advertisement (RA)
  - Neighbor Solicitation (NS)
  - Neighbor Advertisement (NA)
  - ICMP Redirect
- *Inverse Neighbor Discovery* also possible
  - see the literature for details

Neighbor Discovery – L2 address resolution I.

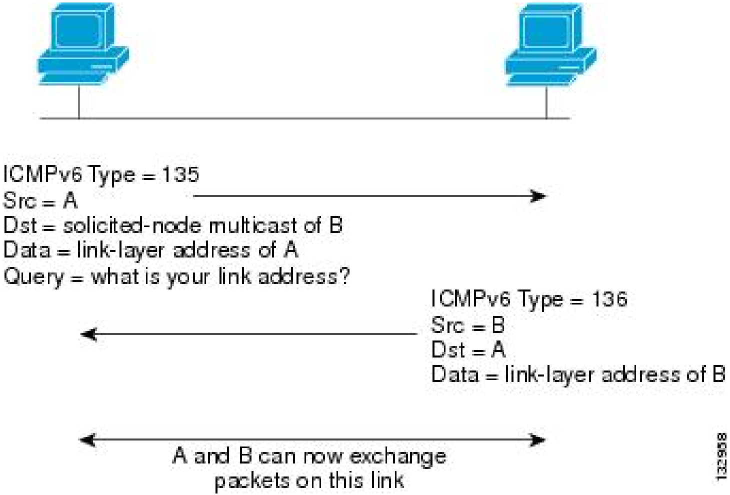
- very similar to ARP in IPv4
- based on *Neighbor Solicitation* and *Neighbor Advertisement* messages
    - a common multicast prefix is defined (`FF02:0:0:0:0:1:FF00::/104`)
    - the node looking for an L2-layer address takes last 24 bits of the IP address, whose L2-address it is looking for, and concatenates it with the prefix
        - e.g., looking for L2-address of `2AC0:56:A319:15:022A:FFF:FE32:5ED1` it receives `FF02:0:0:0:0:1:FF32:5ED1`
        - i.e., the destination address is a multicast address
        - the 24 bits ensure that the multicast group will contain just a few nodes (typically 1 or 0)
    - a **Neighbor Solicitation message** is sent to such a multicast address
        - the message contains the IPv6 address being resolved and the L2 address of the sending node
        - the neighbor has to listen for such messages in his multicast group(s) (based on his IPv6 address(es))

Neighbor Discovery – L2 address resolution II.

- once a node belonging to the particular multicast group receives a NS message, it answers with a **Neighbor Advertisement message**
  - *note:* there might be several nodes in the particular multicast group – just the one having the IPv6 address being resolved answers
- the answer contains:
  - all the IPv6 and L2 addresses the node has
  - an attribute:
    - *R (Router)* – the sender is a router
    - *S (Solicited)* – indicates whether the NA has been solicited or not (unsolicited NAs are possible)
    - *O (Override)* – indicates whether the new information should override the old information previously saved on the node(s)
- *unsolicited Neighbor Advertisement*
  - used in situations, when the node knows that his L2-address has changed
  - these messages are sent to multicast address containing all the nodes (`FF02::1`)

Neighbor Discovery – L2 address resolution II.

The mechanism



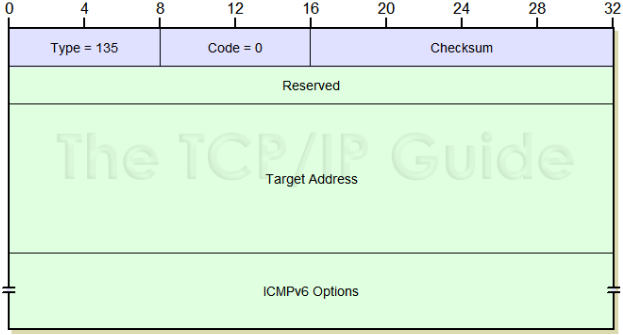
ICMPv6 Type = 135
Src = A
Dst = solicited-node multicast of B
Data = link-layer address of A
Query = what is your link address?

ICMPv6 Type = 136
Src = B
Dst = A
Data = link-layer address of B

A and B can now exchange
packets on this link

Neighbor Discovery – L2 address resolution II.

The mechanism – Neighbor Solicitation message format



Type = 135

Code = 0

Checksum

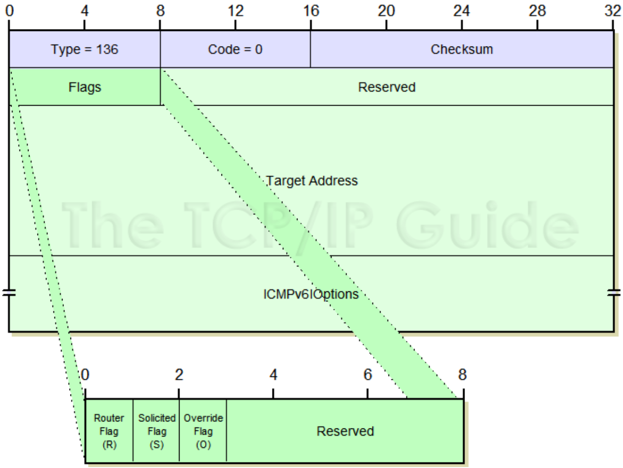
Reserved

Target Address

ICMPv6 Options

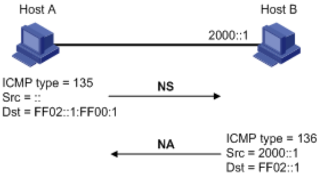
Neighbor Discovery – L2 address resolution II.

The mechanism – Neighbor Advertisement message format



Neighbor Discovery – Duplicate Address Detection (DAD)

- Duplicate Address Detection (DAD)
  - used during autoconfiguration process (see later)
  - the host sends NS message with its own address as the target address
    - destination address in the IPv6 header is set to the solicited-node multicast address
    - the source address is set to the unspecified address (::, i.e. all zeros)
  - if there is another node on the link that is using the same address as the hosts's address, it will reply with a NA message (sent to the all-nodes multicast address), thus exposing the duplicated address to the host

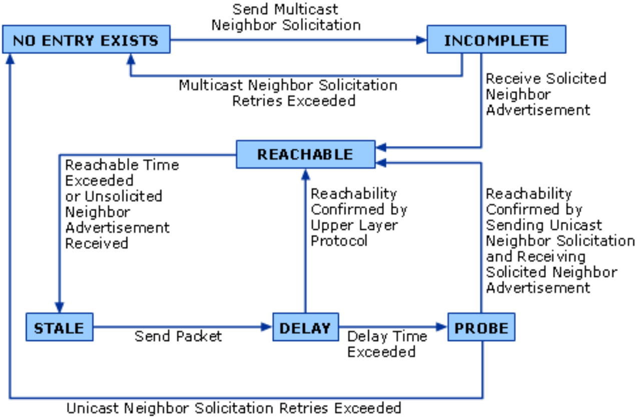


Neighbor Discovery – Neighbor Unreachability Detection

- a node periodically controls the reachability of its neighbors (just the ones it is communicating with)
- can be achieved by two ways:
    - a higher-level protocol (e.g., the TCP) informs IPv6 that the communication proceeds and thus the host is alive
    - otherwise, the IPv6 has to perform such a detection on its own
- a cached address might be in one of the following states:
    - *incomplete* – address resolution is currently being performed and awaiting either a response or timeout (a NS has been sent, but the corresponding NA has not been received yet)
    - *reachable* – this neighbor is currently reachable (positive confirmation within the last *ReachableTime* has been received)
    - *stale* – more than *ReachableTime* milliseconds have elapsed since the last positive confirmation was received
    - *delay* – the neighbor's reachable time has expired; an upper layer protocol might confirm the reachability within a specific time
    - *probe* – a reachability confirmation is being actively attempted

Neighbor Discovery – Neighbor Unreachability Detection

The schema


NO ENTRY EXISTS
Send Multicast Neighbor Solicitation
INCOMPLETE
Multicast Neighbor Solicitation Retries Exceeded
Receive Solicited Neighbor Advertisement
REACHABLE
Reachable Time Exceeded or Unsolicited Neighbor Advertisement Received
Reachability Confirmed by Upper Layer Protocol
Reachability Confirmed by Sending Unicast Neighbor Solicitation and Receiving Solicited Neighbor Advertisement
STALE
Send Packet
DELAY
Delay Time Exceeded
PROBE
Unicast Neighbor Solicitation Retries Exceeded

Neighbor Discovery – Autoconfiguration

- designed to ensure that manually configuring hosts before connecting them to the network is not required
  - even larger sites should not need a DHCP server to configure hosts
  - a key feature when all sorts of devices (TVs, refrigerators, DVD players, etc.) will use IP addresses
- IPv6 supports two types of autoconfiguration:
  - *Stateful autoconfiguration* – like DHCP in IPv4 world (here called DHCPv6)
  - *Stateless autoconfiguration* – new type of autoconfiguration
  - they might be combined
    - stateless configuration can be used to generate IPv6 address and stateful autoconfiguration for additional parameters (e.g., DNS servers)

Neighbor Discovery – Stateless autoconfiguration

- RFC 2462
- assumes that there are clever wisemen (routers) in the network, who know everything necessary
  - from time to time, they inform all the nodes about current configuration (**Router Advertisements**)
  - a new node just waits for an RA or asks for it (**Router Solicitation**)
- router advertisements:
  - periodically sent by every router
    - in random intervals to all the connected networks (via multicast to all connected hosts), or
    - as an answer to router solicitation message (via unicast to the host that has sent the RS)
  - contains specific information about the router
    - MTU
    - prefixes
    - L2-address of the router's interface through which the RA has been sent
    - etc.

Neighbor Discovery – Stateless autoconfiguration

The mechanism I.

- to generate its IP address, a host uses a combination of local information (such as its MAC address or a randomly chosen ID), and information received from routers
- steps, which a device takes when using stateless autoconfiguration:
  - **Link-Local Address Generation** – the device generates a link-local address (so-called *tentative address*)
    - link-local addresses have `1111 1110 10` as first 10 bits (prefix `FE80`)
    - the generated address uses **those ten bits** followed by **54 zeroes** and then the **64 bit interface identifier** (the MAC address or a randomly chosen ID)
  - **Link-Local Address Uniqueness Test** – the node tests to ensure that the address it generated isn't for some reason already in use on the local network
    - this is very unlikely an issue if the link-local address came from a MAC address but more likely if it was based on a generated ID
    - it sends NS message and listens for NA response (see *Duplicate Address Detection* mentioned before)

Neighbor Discovery – Stateless autoconfiguration

The mechanism II.

- cont'd:
    - **Link-Local Address Assignment** – assuming the uniqueness test passes, the device assigns the link-local address to its IP interface
        - this address can be used for communication on the local network, but not on the wider Internet (since link-local addresses are not routed)
    - **Router Contact** – the node next attempts to contact a local router for more information on continuing the configuration
        - this is done either by listening for RA messages sent periodically by routers, or by sending a specific RS message to ask a router for information on what to do next (to the all-routers multicast group, i.e. `FF02::2`)
    - **Router Direction** – the router provides direction to the node on how to proceed with the autoconfiguration
        - it may tell the node that on this network the "stateful" autoconfiguration is in use, and tell it the address of a DHCP server to use. Alternatively, it may tell the host how to determine its global Internet address.

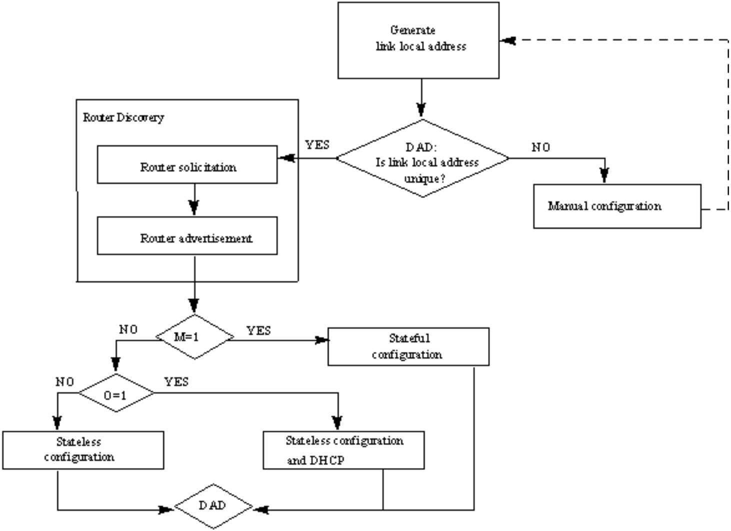
Neighbor Discovery – Stateless autoconfiguration

The mechanism III.

- cont'd:
  - **Global Address Configuration** – assuming that stateless autoconfiguration is in use on the network, the host configures itself with its globally-unique Internet address
    - this address is generally formed from a network prefix provided to the host by the router, combined with the device's identifier as generated in the first step

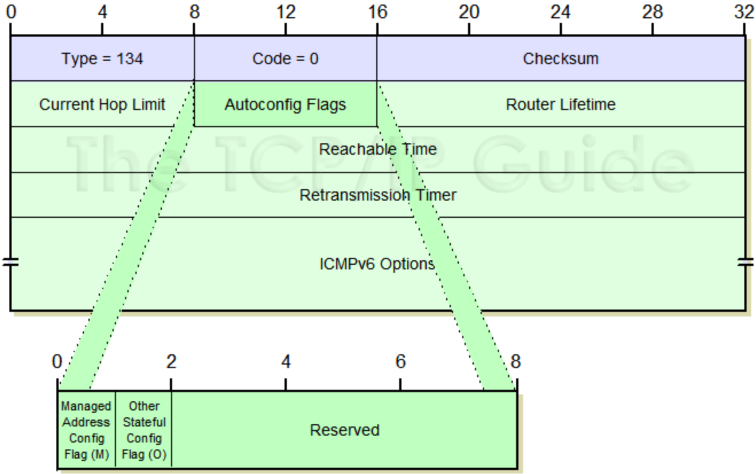
Neighbor Discovery – Stateless autoconfiguration

The schema



Neighbor Discovery – Stateless autoconfiguration

Router Advertisement I.



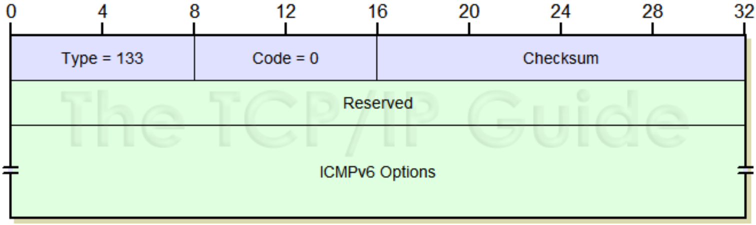
Neighbor Discovery – Stateless autoconfiguration

Router Advertisement II.

- *autoconfiguration flags*:
  - **M** *(Managed Address Configuration Flag)* – tells hosts to use stateful method for address configuration (e.g., the DHCPv6)
  - **O** *(Other Stateful Configuration Flag)* – tells hosts to use stateful method for information other than addresses
- *router lifetime* – tells the host receiving this message how long this router should be used as a default router; if set to 0, tells the host this router should not be used as a default router
- *reachable time* – tells hosts how long they should consider a neighbor to be reachable after they have received reachability confirmation
- *retransmission timer* – the amount of time, in milliseconds, that a host should wait before retransmitting
- *ICMPv6 options* – RA messages may contain three possible options:
  - **source L2 Address** – included when the router sending the RA knows its L2 address
  - **MTU** – used to tell local hosts the MTU of the local network
  - **prefix information** – informs what prefix(es) to use for the local network

Neighbor Discovery – Stateless autoconfiguration

Router Solicitation I.



Type = 133	Code = 0	Checksum
Reserved		
ICMPv6 Options		

ICMPv6 options: if the device sending the RS knows its L2 address, it should be included

Neighbor Discovery Protocol

Summary

- *Neighbor solicitation (NS) message*
  - used to acquire the link-layer address of a neighbor
  - used to verify whether the neighbor is reachable
  - used to perform a duplicate address detection
- *Neighbor advertisement (NA) message*
  - used to respond to a neighbor solicitation message
  - when the link layer address changes, the local node initiates a neighbor advertisement message to notify neighbor nodes of the change
- *Router solicitation (RS) message*
  - once started, a host sends a router solicitation message to request the router for an address prefix and other configuration information (autoconfiguration)
- *Router advertisement (RA) message*
  - used to respond to a router solicitation message
  - a router regularly sends a router advertisement message containing information such as address prefix and flag bits
- *Redirect message*
  - the default gateway might send a redirect message to the source host so that the host can reselect a better/correct next hop router to forward its packets

Lecture overview I

1. Lecture Overview

2. Brief IPv6 Introduction
   - IPv6 Addresses
   - Path MTU discovery

3. IPv6 Neighbor Discovery Protocol in Detail
   - L2 address resolution
   - Duplicate Address Detection
   - Neighbor Unreachability Detection
   - Autoconfiguration
   - Summary

4. IPv6 Mobility Support in Detail
   - Return Routability Procedure

IPv6 – Mobility Support I.

- **main idea:** even mobile devices are somewhere "at home"
    - i.e., their *home network* exists
- used addresses:
    - *Home Address* – a global unicast persistent address, through which a mobile node is always accessible (even though not being in its home network)
    - *Care-of Address* – a global unicast address for the mobile node while it is in a foreign network (the address is based on the network where the host is currently located)
- *Correspondent Node (CN)* – a peer node with which a mobile node is communicating
- *Home Agent (HA)* – a router in the home network, through which the mobile node is always accessible
    - receives datagram destined to the mobile node and forwards them (via a tunnel) to it
- *route optimization* – direct communication of the mobile and corresponding nodes
    - in order to optimize the communication
    - not necessary (the communication might proceed through the home agent all the time)

IPv6 – Mobility Support II.

How it works

- as long as the mobile node is at home, it receives packets through regular IP routing mechanism and behaves like any other host
- when the mobile node is away from the home network, it has an additional care-of address (received via a mechanism available in the foreign network)
    - the association of home address and care-of address is called *binding*
- the mobile node registers its care-of address with a router on its home link (its *Home Agent (HA)*)
- there are two ways to communicate for a correspondent node and a mobile node:
    - *bidirectional tunneling* – packets from the correspondent node are sent to the HA, which encapsulates them and sends them to the mobile node's care-of address (and vice versa)
    - *route optimization* – the communication between the mobile node and correspondent node can be direct without the usage of the HA
        - the mobile node has to register its care-of address with the correspondent node, and
        - the binding has to be authorized through the *Return Routability Procedure*

IPv6 – Mobility Support II.

The schema

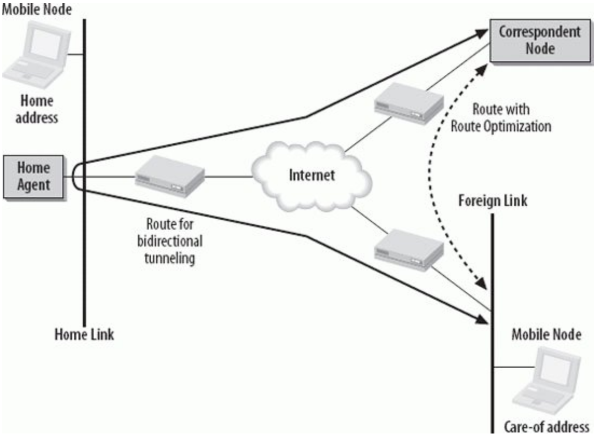


Figure: An illustration of home agent's functionality in IPv6.

IPv6 – Mobility Support II.

Return Routability Procedure

- mobile node must prove to correspondent node that it owns both home address and care-of address
    - but mobile node does not share any secret with the correspondent node
    - initially performed using *IPsec*
        - however, there is no world-wide Public Key Infrastructure (PKI) available for the nodes
- *Return Routability (RR) Procedure*
    - RFC 3775
    - enables the correspondent node to obtain some reasonable assurance that the mobile node is in fact addressable at its claimed care-of address as well as at its home address
        - only when successfully proven, the route optimization might take place
        - reduces the risk of a security attack (a harmful node working off the mobile node)

IPv6 – Mobility Support II.

Return Routability Procedure – the steps

1. MN sends a **Home Test Init (HoTI)** message via HA to the CN (this message carries a *Home Init Cookie*)
   - this way the CN learns the home address of the MN
2. MN sends a **Care-of Test Init (CoTI)** message to the CN (this message carries a *Care-of Init Cookie*) – this is sent to the CN directly (not through the HA)
   - this way the CN learns the care-of address of the MN
3. CN replies to the Home Test Init message with a **Home Test (HoT)** message sent via HA (this message carries the *Home Init Cookie* and the *Home Nonce Index*)
   - the MN can now generate a *Home Keygen Token*
4. CN replies to the Care-of Test Init message with a **Care-of Test (CoT)** message sent to the MN's care-of address (this message carries the *Care-of Init Cookie* and the *Care-of Nonce Index*)
   - the MN can now generate a *Care-of Keygen Token*
5. both the MN and the CN compute a 20-byte *Management Key*, which is used to secure the Binding Update messages
   - having the correct *Management Key* the MN has proven that it is reachable both via its home and care-of addresses

IPv6 – Mobility Support III.

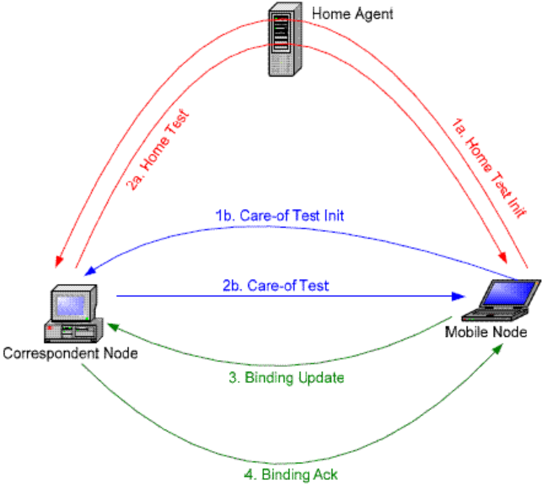
Home Agent Functionality

Home Agent:

- maintains binding cache and a list of home agents
  - every router, that sits on the same link and provides home agent services, must be listed
- processes bindings
  - indicates primary care-of address
  - processes care-of addresses' changes/removals
- tunnels received packets to care-of address
  - performs Neighbor Advertisements by the name of mobile node
- supports *Home Agent Address Discovery*
  - normally, mobile nodes are configured statically with a home agent's address
  - once a home agent is renumbered (or goes down being replaced by another HA with a different IP), dynamic discovery of the HA's address takes place
    - *Home Agent Address Discovery Request* (sent using home agents' anycast address) and *Home Agent Address Discovery Reply* messages
    - see details in the literature

IPv6 – Mobility Support II.

Return Routability Procedure – the schema


Home Agent
2a. Home Test
1a. Home Test Init
1b. Care-of Test Init
2b. Care-of Test
Correspondent Node
Mobile Node
3. Binding Update
4. Binding Ack