Computer Networks :
Protocols and Practice

Part 8
IP version 6

Olivier Bonaventure
http://inl.info.ucl.ac.be/

IP version 6

- Outline
  - Motivations for IP version 6
  - IPv6 addressing architecture
  - IPv6 packets
  - ICMP v6
  - DNS support for IP version 6
  - Mobile IP v6
  - IPv6 Multicast

There are many books and information about IPv6

An interesting book, but written in French, is G. Cizault, IPv6 Théorie et Pratique, O Reilly
The new versions of this book are available online : http://livre.point6.net/index.php/Accueil

A more practically oriented book is
I. van Beijnum, Running IPv6, APress, 2006

IPv6 standardisation is carried out within the IETF, http://www.ietf.org

Other resources include

P. Smith, Introduction to IPv6, NANOG 42, ftp://ftp-eng.cisco.com/pfs/seminars/NANOG42-IPv6-Introduction.pdf

http://www.6journal.org/

http://www.ist-ipv6.org/

Information about IPv6 aware software and hardware is available from

Issues with IPv4

- Late 1980s
  - Exponential growth of Internet
- 1990
  - Other network protocols exist
  - Governments push for CLNP
- 1992
  - Most class B networks have been assigned
  - Class based routing failure
  - Networking experts warn that IPv4 address space could become exhausted

For more information about the exhaustion of IPv4 addresses, see
http://www.potaroo.net/tools/ipv4/index.html

Issues with IPv4 (2)

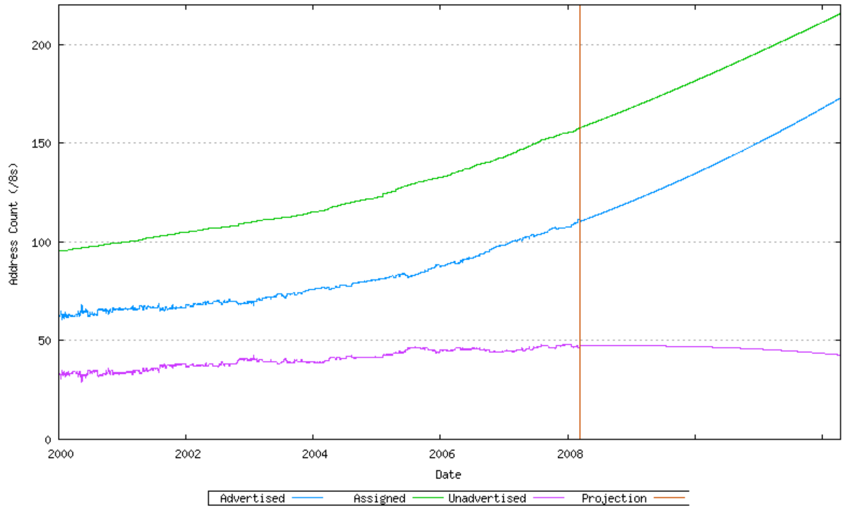
- How to solve the exhaustion of class B addresses ?
- Short term solution
  - Define Classless Interdomain Routing (CIDR) and introduce the necessary changes in routers
  - Deployment started in 1994
- Long term solution
  - Develop Internet Protocol - next generation (IPng)
    - call for proposals RFC1550, Dec 1993
    - Criteria for choix, RFC1719 and RFC1726, Dec. 1994
    - Proposed solutions
      - TUBA - RFC1347, June 1992
      - PIP – RFC1621, RFC1622, May 1994
      - CATNIP – RFC1707, October 1994
      - SIP – RFC1710, October 1994
      - NIMROD – RFC1753, December 1994
      - ENCAPS – RFC1955, June 1996

Issues with IPv4 (3)

- Implementation issues - 1990s
  - IPv4 packet format is complex
  - IP forwarding is difficult in hardware

- Missing functions - 1990s
  - IPv4 requires lots of manual configuration
    - Competing protocols (CLNP, Appletalk, IPX, ...) already supported autoconfiguration in 1990s
  - How to support Quality of Service in IP ?
    - Integrated services and Differentiated services did not exist then
  - How to better support security in IP ?
    - Security problems started to appear but were less important than today
  - How to better support mobility in IP ?
    - GSM started to appear and some were dreaming of mobile devices attached to the Internet

Main motivation today
IPv4 address exhaustion

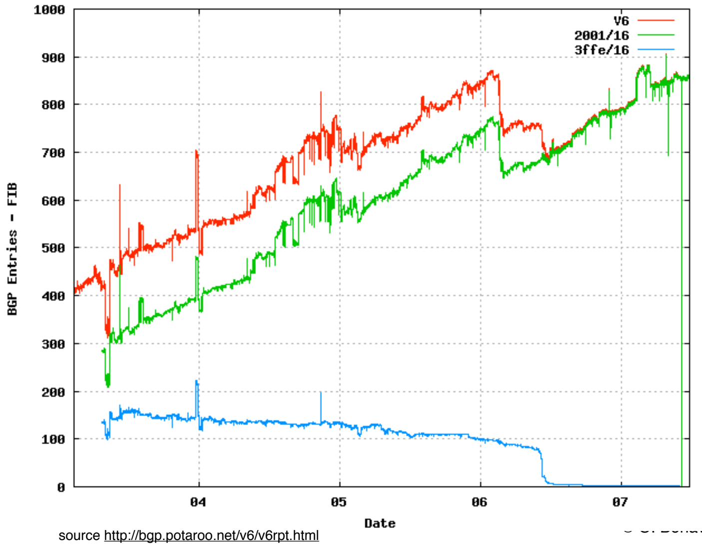


CNPP/2008.8.

Source http://www.potaroo.net/tools/ipv4/index.html

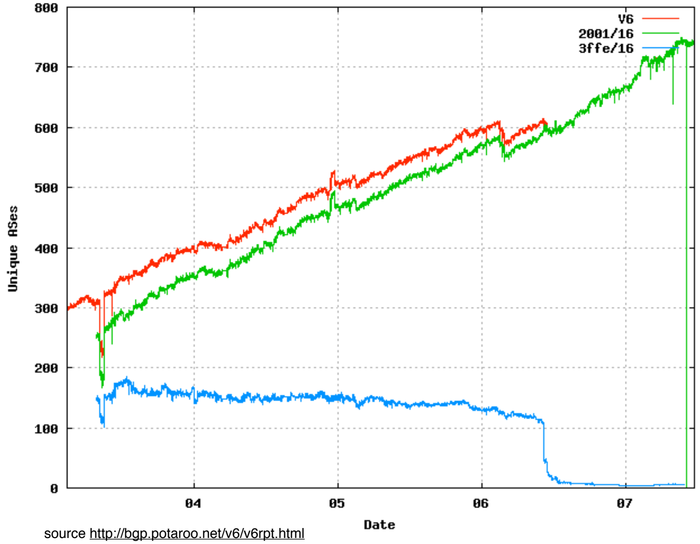
This figure shows the number of IPv4 prefixes used on the global Internet. In addition, some networks, e.g. large cable networks, have had difficulties in using IPv4 due to the limited number of available addresses. For example, comcast is planning to use IPv6 to manage its cable modems mainly because IPv4 does not allow them to have enough addresses to identify all their potential cable modems in a scalable manner, see http://www.nanog.org/mtg-0606/durand.html

IPv6 usage
advertised prefixes



source http://bgp.potaroo.net/v6/v6rpt.html

Current IPv6 usage
ASes using IPv6



CNPP/2008.8.

source http://bgp.potaroo.net/v6/v6rpt.html

ıre 2008

In contrast, the number of ASes using IPv4 is much larger. In March 2008, more than 27000 ASes were advertising IPv4 addresses, see http://bgp.potaroo.net/bgprpts/rva-index.html

Can we avoid deploying IPv6 by using NAT ?

- Network address translation

- Benefits
  - Reduces consumption of public IPv4 addresses
  - “Hides” internal IPv4 addresses inside homes and corporate networks

- Drawbacks
  - Breaks the end-to-end principle
  - Intermediate nodes may modify packet content
    - IP addresses
    - TCP/UDP port information
    - Some protocols encode IP addresses inside payload
      - ftp
      - ...

For a detailed discussion of NAT and its implications, see :

[RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, November 2000.

[RFC3027] Holdrege, M. and P. Srisuresh, "Protocol Complications with the IP Network Address Translator (NAT)", RFC 3027, January 2001.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

IP version 6

- Outline
  - Motivations for IP version 6
  - IPv6 addressing architecture
  - IPv6 packets
  - ICMP v6
  - DNS support for IP version 6
  - Mobile IP v6
  - IPv6 Multicast

IPv6 addresses

IPv4

IP version 6

- Each IPv6 address is encoded in 128 bits
  - 3.4 x 10^38 possible addressable devices
    - 340,282,366,920,938,463,463,374,607,431,768,211,456
  - ~ 5 x 10^28 addresses per person on the earth
  - 6.65 x 10^23 addresses per square meter
  - Looks unlimited.... today
- Why 128 bits ?
  - Some wanted variable size addresses
    - to support IPv4 and 160 bits OSI NSAP
  - Some wanted 64 bits
    - Efficient for software, large enough for most needs
  - Hardware implementers preferred fixed size

IP version 4 supports 4,294,967,296 distinct addresses, but some are reserved for :
private addresses (RFC1918)
loopback (127.0.0.1)
multicast
...

The IPv6 addressing architecture

- Three types of IPv6 addresses
  - Unicast addresses
    - An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address
  - Anycast addresses
    - An identifier for a set of interfaces. A packet sent to an
    - anycast address is delivered to the “nearest” one of the interfaces identified by that address
  - Multicast addresses
    - An identifier for a set of interfaces. A packet sent to a multicast address is delivered to all interfaces identified by that address.

The IPv6 addressing architecture is defined in :

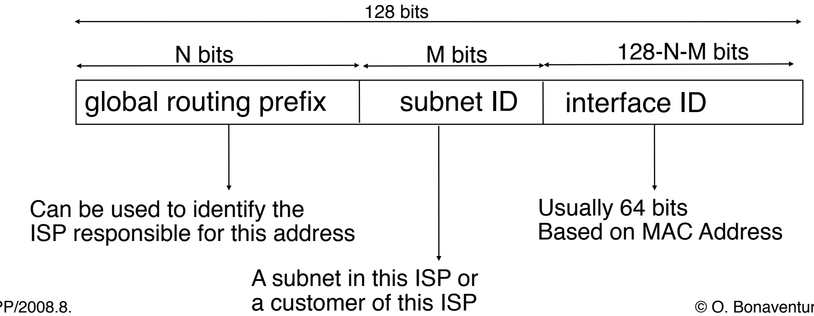
R. Hinden, S. Deering, IP Version 6 Addressing Architecture, RFC4291, February 2006

Representation of IPv6 addresses

- How can we write a 128 bits IPv6 address ?
  - Hexadecimal format
    - FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
    - 1080:0:0:0:8:800:200C:417A
  - Compact hexadecimal format
    - Some IPv6 addresses contain lots of zero
      - utilize "::" to indicate one or more groups of 16 bits of zeros. The "::" can only appear once in an address
      - Examples
        - 1080:0:0:0:8:800:200C:417A = 1080::8:800:200C:417A
        - FF01:0:0:0:0:0:0:101 = FF01::101
        - 0:0:0:0:0:0:0:1 = ::1

The IPv6 unicast addresses

- Special addresses
  - Unspecified address : 0:0:0:0:0:0:0:0
  - Loopback address : 0:0:0:0:0:0:0:1
- Global unicast addresses
  - Addresses will be allocated hierarchically



Today, the default encoding for global unicast addresses is to use :

48 bits for the global routing prefix (first three bits are set to 001)

16 bits for the subnet ID

64 bits for the interface ID

Allocation of IPv6 addresses

- IANA controls all IP addresses and delegates assignments of blocks to Regional IP Address Registries (RIR)
  - RIPE, ARIN, APNIC, AFRINIC, ...
- An organisation can be allocated two different types of IPv6 addresses
  - Provider Independent (PI) addresses
    - Usually allocated to ISPs or very large enterprises directly by RIRs
    - Default size is /32
  - Provider Aggregatable (PA) addresses
    - Smaller prefixes, assigned by ISPs from their PI block
    - Size
      - /48 in the general case, except for very large subscribers
      - /64 when t one and only one subnet is needed by design
      - /128 when it is absolutely known that one and only one device is connecting.

See http://www.ripe.net/ripe/docs/ripe-388.html for the policy used by RIPE to allocate IP prefixes in Europe

The IPv6 link-local addresses

- Used by hosts and routers attached to the same LAN to exchange IPv6 packets when they don't have/need globally routable addresses

128 bits

10 bits	54 bits	64 bits
FE80	0000000000.....00000000000	interface ID

- Each host must generate one link local address for each of its interfaces
  - Each IPv6 host will use several IPv6 addresses
- Each routers must generate one link local address for each of its interfaces

Site-local addresses were defined in the first IPv6 specifications, but they are now deprecated and should not be used.

Recently "private" addresses have been defined as Unique Local IPv6 Addresses as a way to allow entreprise to obtain IPv6 addresses without being forced to request them from providers or RIRs.

The way to choose such a ULA prefix is defined in :

R. Hinden, B. Haberman,   Unique Local IPv6 Unicast Addresses, RFC4193, October 2005

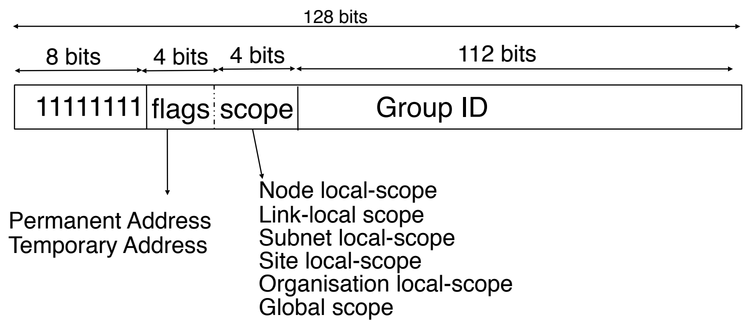
Recently, the case for a registration of such addresses has been proposed, see :
R. Hinden, G. Huston, T. Narten, Centrally Assigned Unique Local IPv6 Unicast Addresses, internet draft, <draft-ietf-ipv6-ula-central-02.txt>, work in progress, June 2007

See also

http://www.ripe.net/ripe/policies/proposals/2007-05.html -

The IPv6 multicast addresses

- An IPv6 multicast address identifies a group a receivers



- Well known groups
  - All endsystem automatically belong to the FF02::1 group
  - All routers automatically belong to the FF02::2 group

The full list of well known IPv6 multicast groups is available from http://www.iana.org/assignments/ipv6-multicast-addresses

Examples include

Node-Local Scope

```
 FF01:0:0:0:0:0:0:1     All Nodes Address                  [RFC4291]
 FF01:0:0:0:0:0:0:2     All Routers Address                [RFC4291]
```

Link-Local Scope

```
 FF02:0:0:0:0:0:0:1     All Nodes Address                  [RFC4291]
 FF02:0:0:0:0:0:0:2     All Routers Address                [RFC4291]
 FF02:0:0:0:0:0:0:5     OSPFIGP                            [RFC2328,Moy]
 FF02:0:0:0:0:0:0:6     OSPFIGP Designated Routers         [RFC2328,Moy]
 FF02:0:0:0:0:0:0:9     RIP Routers                        [RFC2080]
```

The IPv6 anycast addresses

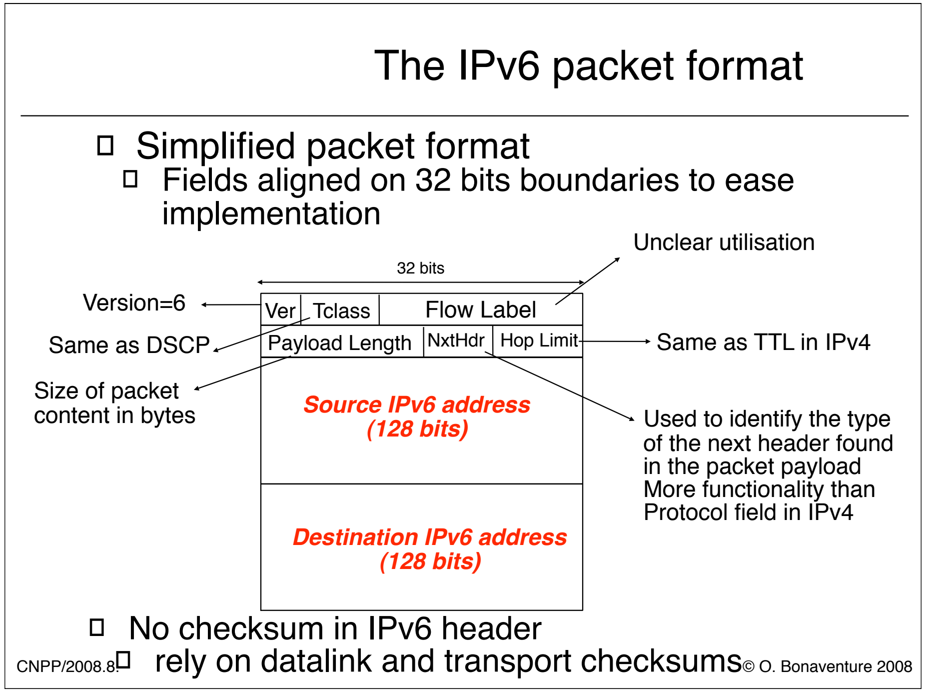
- Definition
  - An IPv6 anycast address is an address that is assigned to more than one interface (typically belonging to different nodes), with the property that a packet sent to an anycast address is routed to the "nearest" interface having that address, according to the routing protocols' measure of distance.
- Usage
  - Multiple redundant servers using same address
  - Example DNS resolvers and DNS servers
- Representation
  - IPv6 anycast addresses are unicast addresses
  - Required subnet anycast address

n bits	128-n bits
IPv6 subnet prefix	00000000000000000000000

The allocated anycast addresses are references in http://www.iana.org/assignments/ipv6-anycast-addresses

IP version 6

- Outline
  - Motivations for IP version 6
  - IPv6 addressing architecture
  - IPv6 packets
  - ICMP v6
  - DNS support for IP version 6
  - Mobile IP v6
  - IPv6 Multicast



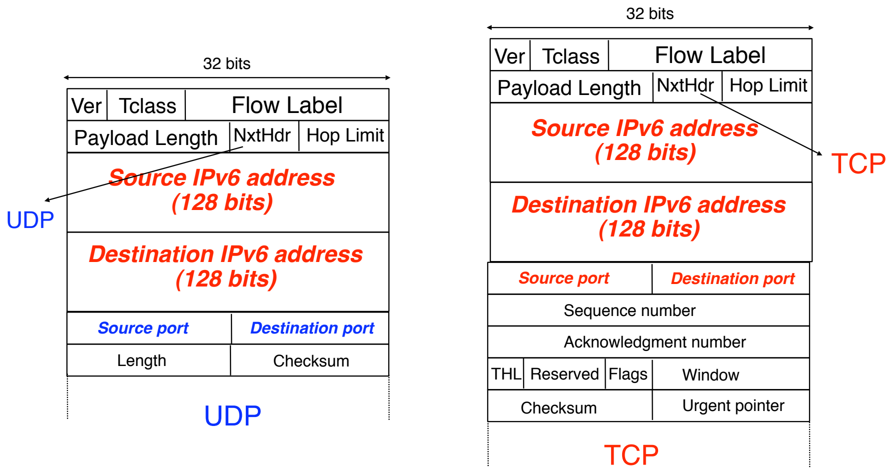
The IPv6 packet format is described in
S. Deering, B. Hinden, Internet Protocol, Version 6 (IPv6) Specification , RFC2460, Dec 1998

Several documents have been written about the usage of the Flow label. The last one is

J. Rajahalme, A. Conta, B. Carpenter, S. Deering, IPv6 Flow Label Specification, RFC3697, 2004

However, this proposal is far from being widely used and deployed.

Sample IPv6 packets



- Identification of a TCP connection
  - IPv6 source, IPv6 destination, Source and Destination ports

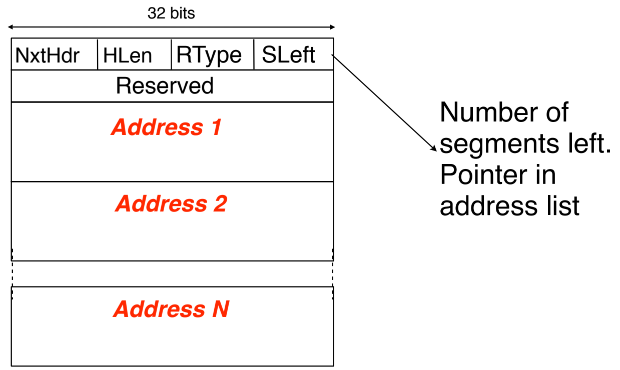
IPv6 does not require changes to TCP and UDP for IPv4. The only modification is the computation of the checksum field of the UDP and TCP headers since this checksum is computed by concerning a pseudo header that contains the source and destination IP addresses.

The IPv6 extension headers

- Several types of extension headers
  - Hop-by-Hop Options
    - contains information to be processed by each hop
  - Routing (Type 0 and Type 2)
    - contains information affecting intermediate routers
  - Fragment
    - used for fragmentation and reassembly
  - Destination Options
    - contains options that are relevant for destination
  - Authentication
    - for IPSec
  - Encapsulating Security Payload
    - for IPSec
- Each header must be encoded as n*64 bits

An example hop-by-hop option is the router alert option defined in
A. Jackson, C. Partridge, IPv6 Router Alert Option RFC2711, 1999

Type 0 Routing header

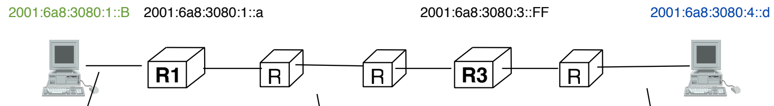


- Defined as "*a mean for a source to list one or more intermediate nodes to be "visited" on the way to a packet's destination*

The Type 0 Routing header is specified in RFC2460

Two other types of routing headers have been defined. Type 1 is experimental and never used. Type 2 is specific for Mobile IPv6 that will be covered later.

Type 0 routing header example



SRC: 2001:6A8:3080:1::B
001:6A8:3080:1::A

NxtHdr	HLen	RType	2
Reserved			

2001:6A8:3080:1::A
2001:6A8:3080:3::FF
2001:6A8:3080:4::D

SRC: 2001:6A8:3080:1::B
2001:6A8:3080:3::FF

NxtHdr	HLen	RType	1
Reserved			

2001:6A8:3080:1::A
2001:6A8:3080:3::FF
2001:6A8:3080:4::D

SRC: 2001:6A8:3080:1::B
2001:6A8:3080:4::D

NxtHdr	HLen	RType	0
Reserved			

2001:6A8:3080:1::A
2001:6A8:3080:3::FF
DST: 2001:6A8:3080:4::D

Issues with Type 0 Routing header

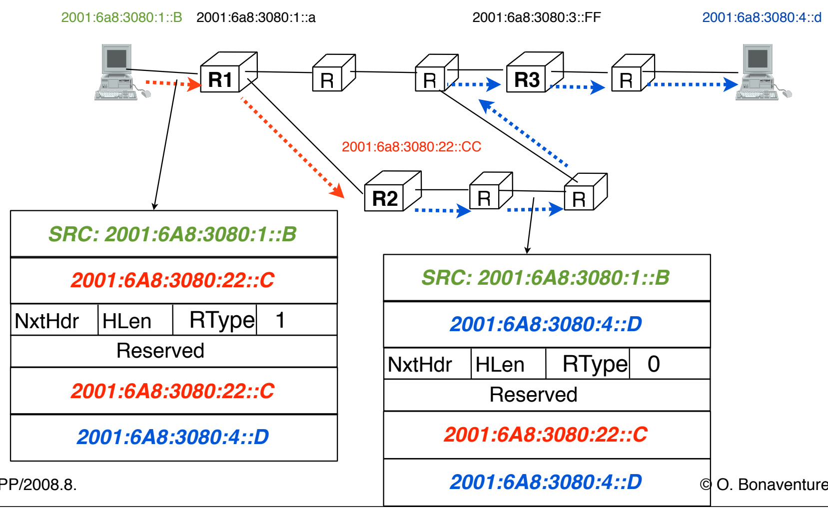
- Type 0 RH is a generalisation of IPv4 source routing
- The IPv6 specification is unclear about the processing of Type 0 RH
  - Node = *a device that implements IPv6*
  - Router = *a node that forwards IPv6 packets not explicitly addressed to itself*
  - Host = *any node that is not a router*
- How to process headers ?
  - *IPv6 nodes must accept and attempt to process extension headers in any order and occurring any number of times in the same packet, . . .*

The type 0 routing header was deprecated in
J. Abley, P. Savola, G. Neville-Neil, Deprecation of Type 0 Routing Headers in IPv6 RFC5095, Dec. 2007

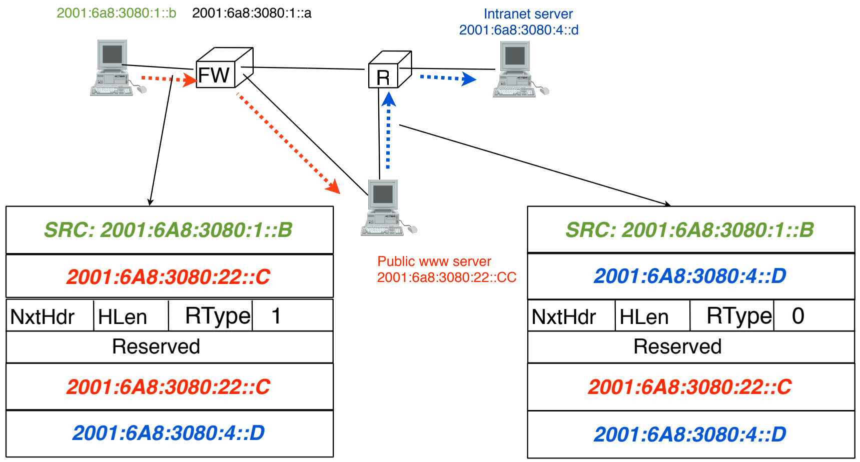
For more information about the security issues with this header, see
Biondi, P. and A. Ebalard, "IPv6 Routing Header Security", CanSecWest Security Conference 2007, April 2007. http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf

Other usage of Type 0 RH

- Improved topology discovery with traceroute

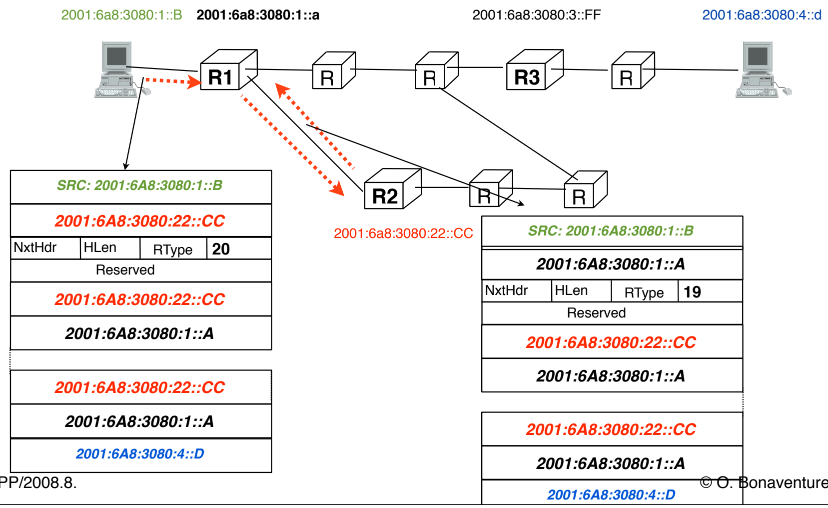


Problems with Type 0 RH



More serious problem with Type 0 RH

- Increases impact of DoS attacks



Hop-by-hop and destination option headers

- TLV format of these options

NxtHdr	HLen	Type	Len
Data (var. length)			

- Two leftmost bits
  - How to deal with unknown option ?
    - 00 ignore and continue processing
    - 01 silently discard packet
    - 10 discard packet and send ICMP parameter problem back to source
    - 11 discard packet and send ICMP parameter problem to source if destination isn’t multicast
  - Third bit
    - Can option content be changed en-route
  - Five rightmost bits
    - Type assigned by IANA

The Len field encodes the size of the data field in bytes. Furthermore, special options have been defined to allow hosts using the options to pad the size of vairable length options to multiples of 64 bits.

Pad1 option (alignment requirement: none)

```
+-+-+-+-+-+-+-+-+
|       0       |
+-+-+-+-+-+-+-+-+
```

NOTE! the format of the Pad1 option is a special case -- it does not have length and value fields.

The Pad1 option is used to insert one octet of padding into the Options area of a header. If more than one octet of padding is required, the PadN option, described next, should be used, rather than multiple Pad1 options.

IPv6 jumbograms

- IPv6 packet format only supports 64 KBytes packets
  - packet size is encoded in 16 bits field
- on most hosts throughput increases with packet size

- Hop-by-hop jumbogram option
  - Increases packet size to 32 bits
    - when used, packet size in IPv6 header should be set to zero

NxtHdr	HLen	C2	Len:4
Packet size			

C2 : 11 0 00020
11 -> ICMP must be sent
if option is unrecognised
0 -> content of option
does not change en-route

As of today, it is unclear whether the jumbogram option has been implemented in practice. Using it requires link layer technologies that are able to support frames larger than 64 KBytes.

The jumbogram option has been defined in

D. Borman, S. Deering, B. Hinden, IPv6 Jumbograms, RFC2675, August 1999

The Kame (http://www.kame.net) implementation on FreeBSD supports this option, but there is no link-layer that supports large frames.

Packet fragmentation

- IPv4 used packet fragmentation on routers
  - All hosts must handle 576+ bytes packets
  - experience showed fragmentation is costly for routers and difficult to implement in hardware
  - PathMTU discovery is now widely implemented
- IPv6
  - IPv6 requires that every link in the internet have an MTU of 1280 octets or more
    - otherwise link-specific fragmentation and reassembly must be provided at a layer below IPv6
  - Routers do not perform fragmentation
    - Only end hosts perform fragmentation and reassembly by using the fragmentation header
    - But PathMTU discovery should avoid fragmentation most of the time

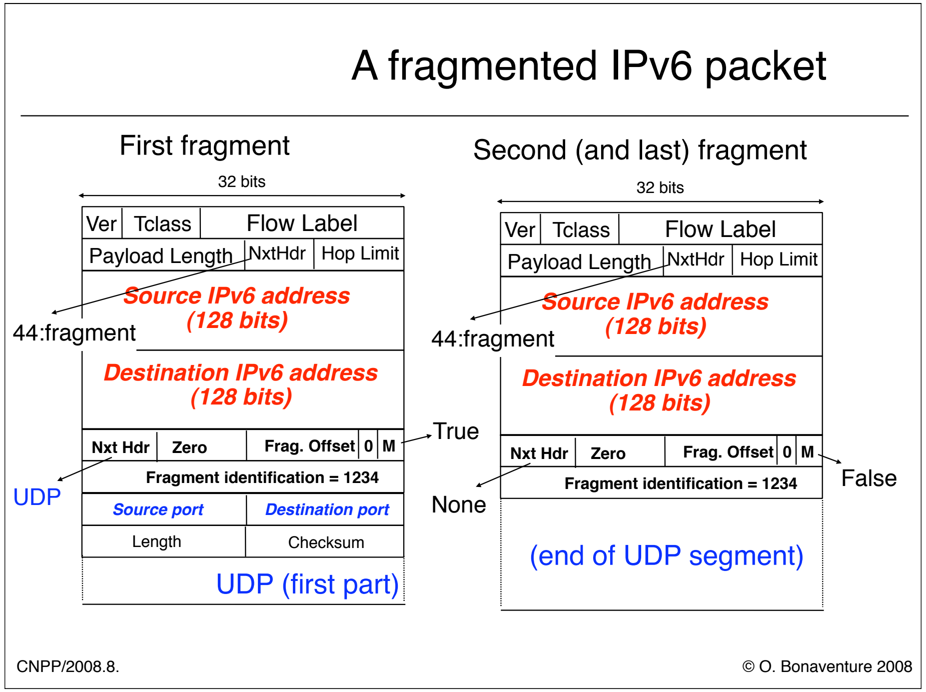
Path MTU discovery is defined in

J. Mogul, S. Deering, Path MTU Discovery, RFC1191, 1996

and in

J. McCann, S. Deering, J. Mogul, Path MTU Discovery for IP version 6, RFC1981, 1996

for IPv6



In IPv6, the fragment identification field is much larger than in IPv4. Furthermore, it is only used in packets that really need fragmentation. IPv6 header does not contain a fragmentation information for each unfragmented packet unlike IPv4.

IP version 6

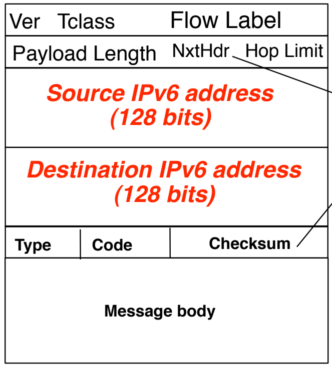
- Outline
  - Motivations for IP version 6
  - IPv6 addressing architecture
  - IPv6 packets
  - ICMP v6
  - DNS support for IP version 6
  - Mobile IP v6
  - IPv6 Multicast

ICMPv6

- Provides the same functions as ICMPv4, IGMP and Address Resolution Protocol (ARP)
- Types of ICMPv6 messages
  - Destination unreachable
  - Packet too big
    - Used for PathMTU discovery
  - Time expired (Hop limit exhausted)
    - Traceroute v6
  - Echo request and echo reply
    - Pingv6
  - Multicast group membership
  - Router advertisments
  - Neighbor discovery
  - Autoconfiguration

ICMPv6 is defined in :
A. Conta, S. Deering, M. Gupta, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC4443, March 2006

ICMPv6 packet format



58 for ICMPv6

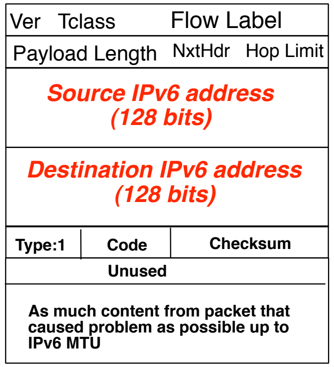
Covers ICMPv6 message and part of IPv6 header

- Type
- ICMPv6 error messages (0<type<127)
  - 1 Destination Unreachable
  - 3 Time Exceeded
  - 2 Packet Too Big
  - 4 Parameter Problem
  - 100 Private experimentation
  - 101 Private experimentation
  - 127 Reserved for expansion
- ICMPv6 informational messages:
  - 128 Echo Request
  - 129 Echo Reply
  - 200 Private experimentation
  - 201 Private experimentation
  - 255 Reserved for expansion

  of ICMPv6 informational

ICMPv6 uses a next header value of 58 inside IPv6 packets

ICMPv6 destination unreachable



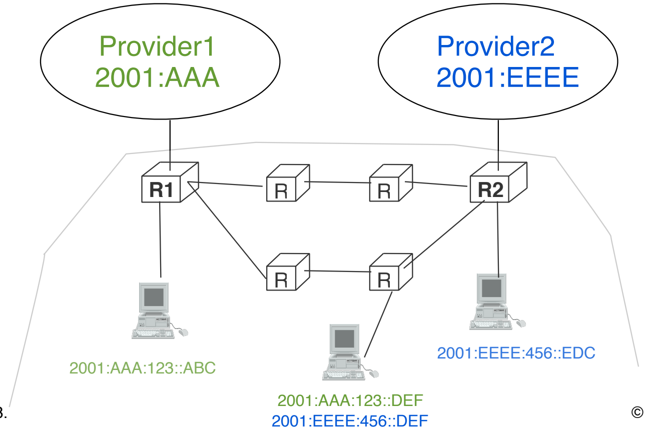
- Code
  - 0 - No route to destination
  - 1 - Communication with destination administratively prohibited
  - 2 - Beyond scope of source address
  - 3 - Address unreachable
  - 4 - Port unreachable
  - 5 - Source address failed ingress/egress policy
  - 6 - Reject route to destination

The Unused field is used to align the content of the ICMPv6 message to a 64 bits boundary.

Note that for security reasons, it is recommended that implementations should allow sending of ICMP destination unreachable messages to be disabled, preferably on a per-interface basis.

Ingress and egress policies

- For security reasons, a provider should only accept packets from sources belonging to allocated prefixes

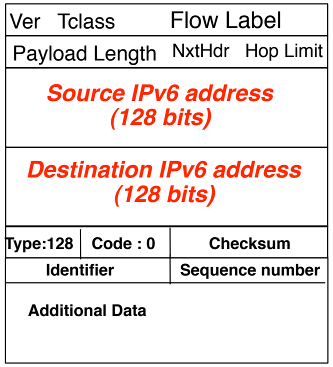


These policies are described in

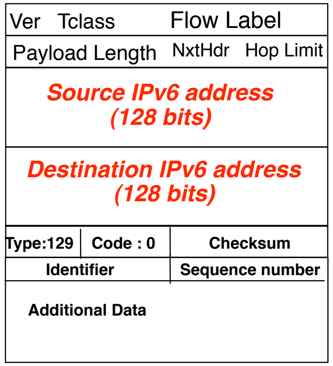
F. Baker, P. Savola, Ingress Filtering for Multihomed Networks, RFC3704, March 2004

ICMPv6
echo request and reply

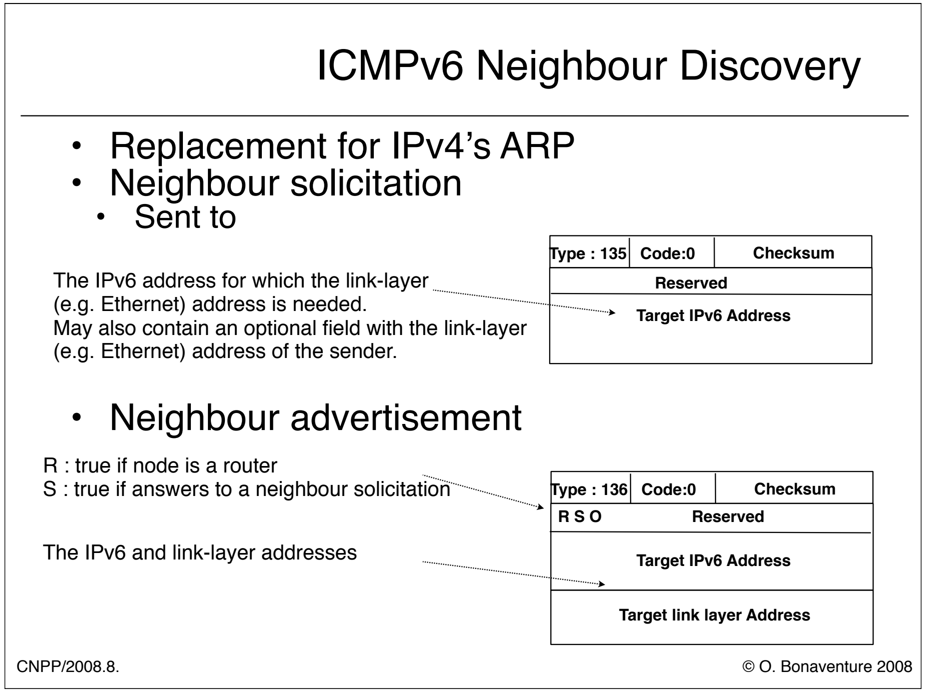
Echo request



Echo reply



- Identifier and sequence number
  - chosen by source to aid in correlating reply with request
  - copied by destination when generating echo reply



The ICMPv6 neighbour discovery messages are sent with HopLimit=255

The role of the R, S and O flags is described as follows in RFC4861

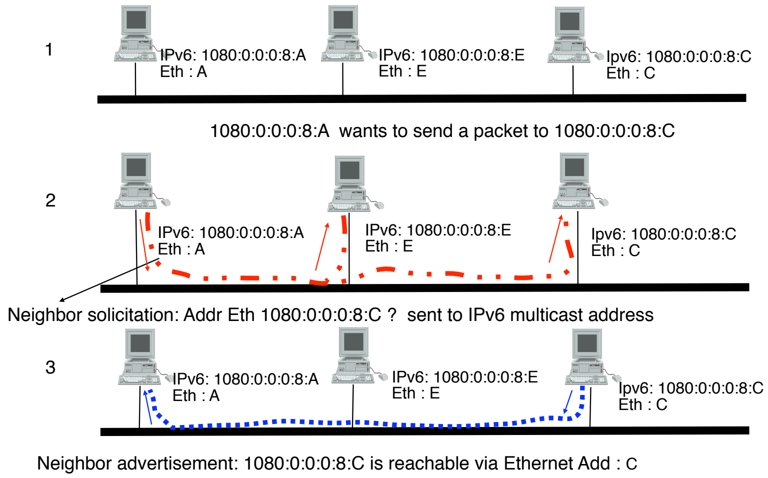
R Router flag. When set, the R-bit indicates that the sender is a router. The R-bit is used by Neighbor Unreachability Detection to detect a router that changes to a host.

S Solicited flag. When set, the S-bit indicates that the advertisement was sent in response to a Neighbor Solicitation from the Destination address. The S-bit is used as a reachability confirmation for Neighbor Unreachability Detection. It MUST NOT be set in multicast advertisements or in unsolicited unicast advertisements.

O Override flag. When set, the O-bit indicates that the advertisement should override an existing cache

IPv6 over Ethernet

- Neighbor discovery / address resolution



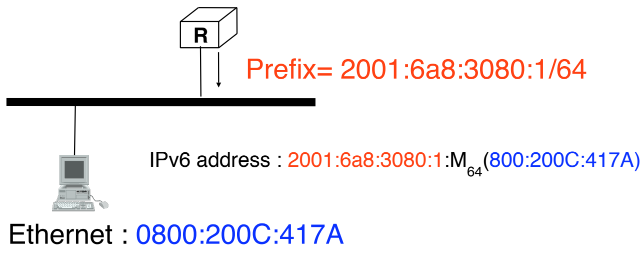
CNPP/2008.8.

The transmission of IPv6 packets over Ethernet is defined in :
M. Crawford, Transmission of IPv6 Packets over Ethernet Networks, RFC2464, December 1998

Note that in contrast with ARP used by IPv4, ICMPv6 neighbour solicitation messages are sent to a multicast ethernet address and not to the broadcast ethernet address. This implies that only the IPv6 enabled hosts on the LAN will receive the ICMPv6 message.

IPv6 autoconfiguration

- How can a node obtain its IPv6 address ?
  - Manual configuration
  - From a server by using DHCPv6 as in IPv4
- Automatically
  - Router advertises prefix on LAN by sending ICMPv6 messages to “all IPv6 hosts” multicast address
  - Hosts build their address by concatenating the prefix with their MAC Address converted in 64 bits format



CNPP/2008.8.

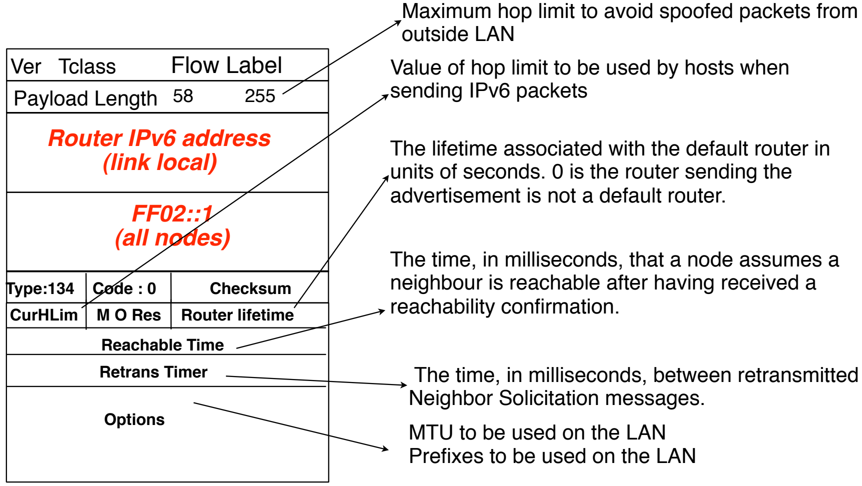
M_{64}(800:200C:417A) is a function that converts a 48 bits MAC address into a 64 bits Interface Identifier. This function is defined in :

R. Hinden, S. Deering, IP Version 6 Addressing Architecture, RFC4291, February 2006

The IPv6 autoconfiguration is defined in :

S. Thomson, T. Narten, T. Jinmei, IPv6 Stateless Address Autoconfiguration, RFC4862, Sept. 2007

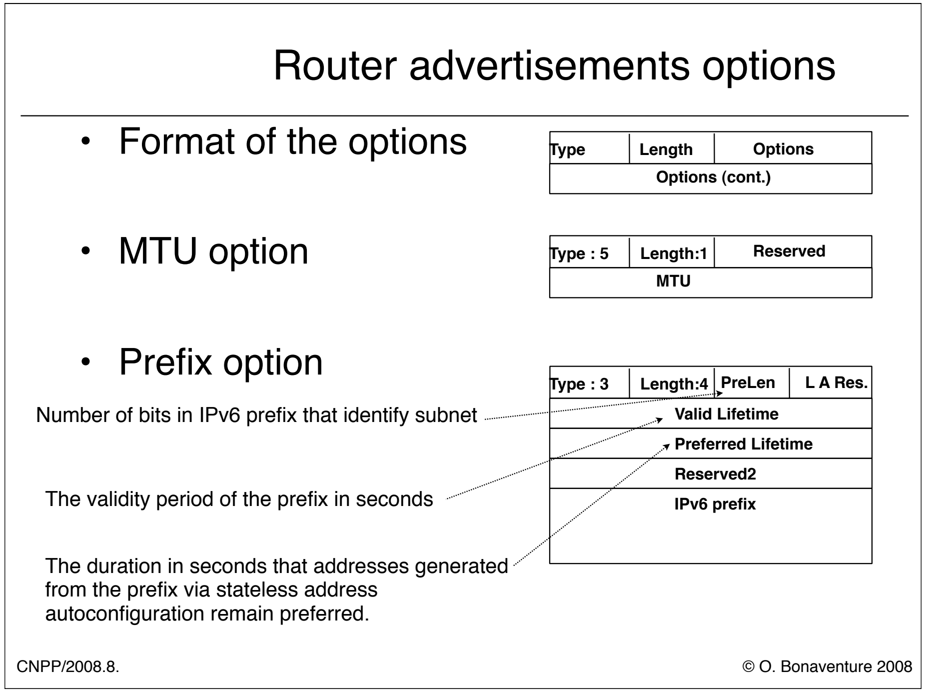
Router advertisements



When the M bit is set to true, this indicates that IPv6 addresses should be obtained from DHCPv6

When the O bit is set to true, this indicates that the hosts can obtain additional information (e.g. address of DNS resolver) from DHCPv6

The router advertisements messages can also be sent in unicast in response to solicitations from hosts. A host can obtain a router advertisement by sending a router solicitation which is an ICMPv6 message containing only the router solicitation message (type 133).



The two L and A bits are defined as follows :

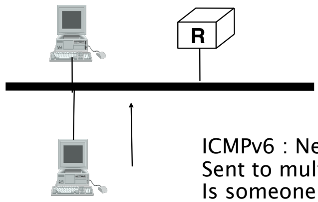
L 1-bit on-link flag. When set, indicates that this prefix can be used for on-link determination. When not set the advertisement makes no statement about on-link or off-link properties of the prefix. In other words, if the L flag is not set a host MUST NOT conclude that an address derived from the prefix is off-link. That is, it MUST NOT update a previous indication that the address is on-link.

A 1-bit autonomous address-configuration flag. When set indicates that this prefix can be used for stateless address configuration.

Other options have been defined for the router advertisements. For example, the RDNSS option defined in
J. Jeong, S. Park, L. Beloeil, S. Madanapalli, IPv6 Router Advertisement Option for DNS Configuration, RFC 5006, Sept. 2007

IPv6 autoconfiguration (2)

- What happens when an endsystem boots ?
  - It knows nothing about its current network
    - but needs an IPv6 address to send ICMPv6 messages

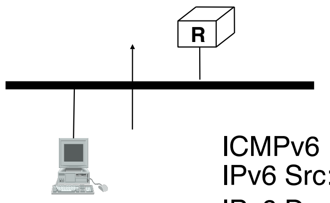


- Use Link-local IPv6 address (FE80::/64)
  - Each host, when it boots, has a link-local IPv6 address
  - But another node might have chosen the same address !

This utilisation of ICMPv6 Neighbour solicitation is called Duplicate Address Detection. It is used everytime a host obtains a new IPv6 address and is required to ensure that a host
is not using the same IPv6 address as another host on the same LAN.

IPv6 autoconfiguration (2)

- How to obtain the IPv6 prefix of the subnet ?
  - Wait for router advertisements (e.g. 30 seconds)
  - Solicit router advertisement

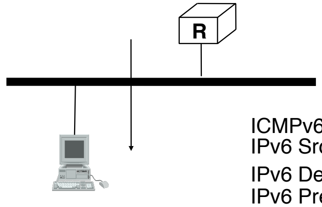


ICMPv6 : Router Solicitation
IPv6 Src: FE80::M_{64}(800:200C:417A)
IPv6 Dest: FF02::2

Ethernet : 0800:200C:417A
FE80::M_{64}(800:200C:417A)

IPv6 autoconfiguration (3)

- Router will re-advertise prefix



Ethernet : 0800:200C:417A
FE80::M_{64}(800:200C:417A)

ICMPv6 : Router Advertisement
IPv6 Src: FE80::M_{64}(EthernetR)
IPv6 Dest: FF02::1
IPv6 Prefix = 2001:6a8:1100::/48
Prefix lifetime

- IPv6 addresses can be allocated for limited lifetime
  - This allows IPv6 to easily support renumbering

IPv6 is supposed to easily support renumbering and IPv6 router advertisements are one of the ways to perform this renumbering by allowing hosts to update their IPv6 addresses upon reception of new router advertisement messages. However, in practice renumbering an IPv6 network is not easily because IPv6 addresses are manually encoded in too many configuration files, see e.g. :

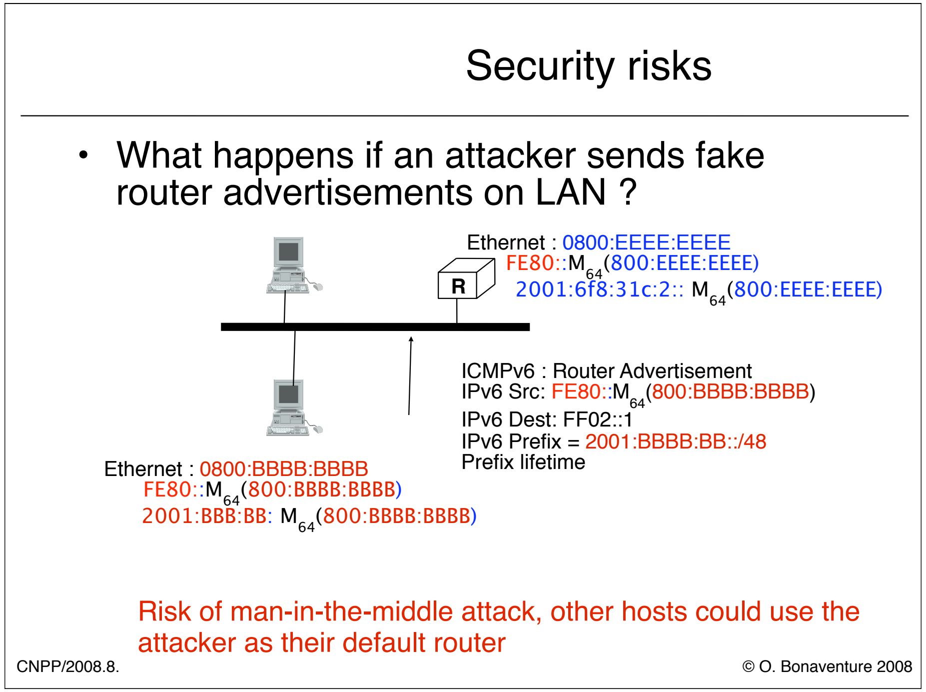
F. Baker, E. Lear, R. Droms, Procedures for Renumbering an IPv6 Network without a Flag Day, RFC4192, 2005

Privacy issues with IPv6 address autoconfiguration

- Issue
  - Autoconfigured IPv6 addresses contain the MAC address of the hosts
    - MAC addresses are fixed and unique
    - A laptop/user could be identified by tracking the lower 64 bits of its IPv6 addresses
- How to maintain privacy with IPv6 ?
  - Use DHCPv6 and configure server to never reallocate the same IPv6 address
  - Allow hosts to use random host ids in lower 64 bits of their IPv6 address
    - algorithms have been implemented to generate such random host ids on nodes with and without stable storage

This extension to support privacy-aware IPv6 addresses is defined in

T. Narten, R. Draves, S. Krishnan, Privacy Extensions for Stateless Address Autoconfiguration in IPv6, RFC4941, Sept. 2007



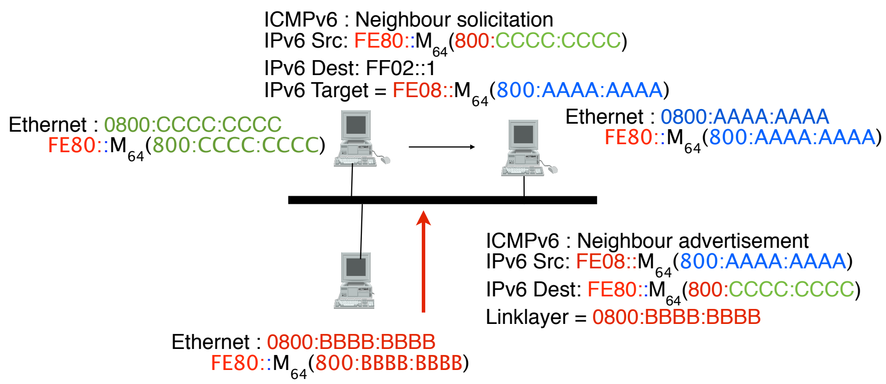
48

A discussion of the security issues with Neighbour discovery may be found in

P. Nikander, J. Kempf, E. Nordmark, IPv6 Neighbor Discovery (ND) Trust Models and Threats, RFC3756, May 2004

Security risks (2)

- What happens if an attacker sends fake ICMPv6 neighbour advertisements ?

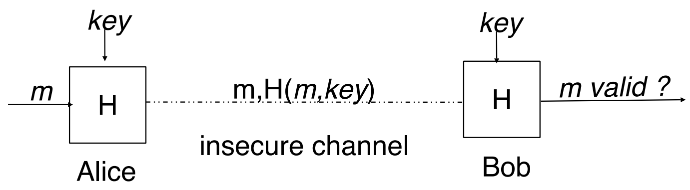


Securing ICMPv6

- Principle of the solution
  - A host that replies to an ICMPv6 neighbour solicitation should be able to prove that it owns the corresponding IPv6 address
  - A router that sends router advertisements should be able to prove that it is authorised to serve as a router using the advertised prefixes
- Issues
  - How to exchange theses proofs and authorisations ?
  - Is IPSec a solution ?

Cryptographical building blocks
Hash functions

- Hash functions

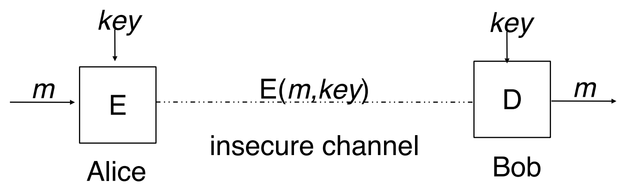


  - Properties
    - Easy to compute H(Msg,key)
    - Very difficult to find Msg2 : H(Msg,k)=H(Msg2,k)

- Example hash functions
  - MD5, MD4, SHA-1

Cryptographical building blocks
Secret-key cryptography

- Secret-key cryptography



- Advantages
  - Efficient algorithms exist
  - Security is f(implementation and key size)
- Drawbacks
  - Key must be distributed securely
  - Does not provide any authentication scheme
- Examples : DES, AES, RC-4, IDEA,...

CNPP/2008.8.

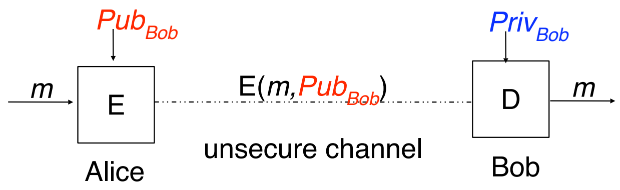
A detailed description of (too) many cryptographical algorithms may be found in :
B. Scheneier, Applied Cryptography, second edition, Wiley, 1995

A more concise description appears in :

C. Kaufman, R. Perlman and M. Speciner, Network Security : Private Communications in a public world, Prentice Hall, second edition, 2002

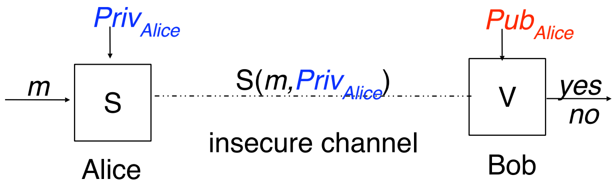
Cryptographical building blocks
Public-key cryptography

- Public-key cryptography
  - Each user maintains two keys
    - A public key ($Public_{Key}$) which can be made public and can be used by any user to send him/her encrypted messages
    - A private key ($Private_{Key}$) which is kept secret and can be used to decrypt information encrypted with the public



Cryptographical building blocks
Public-key cryptography (2)

- Advantages
  - Users do not need to share a secret key to be able to encrypt messages
  - Public-key cryptography allows signatures



  - Security is f(implementation and key size)
- Drawbacks
  - Public-key cryptography is 10 or 100 times slower than secret-key cryptography
- Examples : RSA, DSS

First solution : certificates

- Principle
  - Each router has a public/private keypair
  - A certificate is generated for each router to confirm :
    - that the keypair belongs to the router
    - that the owner of the keypair is a valid router
  - Certificate must be anchored on an authority that is trusted by both routers and hosts
  - ICMPv6 router advertisement messages are signed by the router
- Protocol issues
  - Need to extend ICMPv6 to support signatures and certificates

Additional information about the utilisation of X.509 certificates to represent IP prefixes and AS resources, see :

Lynn, C., Kent, S. and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.

The development of these certificates is being performed within the SIDR working group of the IETF.

Cryptographically Generated Addresses

- Placing certificates on all hosts is too difficult
- We usually don't need to prove that a host is a host

- Can we verify the validity of signed messages without relying on a PKI ?

- Principle of the solution
  - Assume that IPv6 addresses are variable-length
  - Generate IPv6 addresses as follows

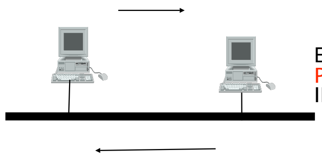
Global prefix + subnet id (64bits)	Host's public key

  - Use private key to sign ICMPv6 neighbour advertisement messages

Principles of Secure Neighbour Discovery

ICMPv6 : Neighbour solicitation
IPv6 Src: FE80:: KeyC
IPv6 Dest: FF02::1
IPv6 Target = FE80::KeyA
Nonce=1234
Timestamp : April14,2008, 10.00:01

Ethernet : 0800:CCCC:CCCC
Public key : KeyC
IPv6 : FE80::KeyC



Ethernet : 0800:AAAA:AAAA
Public key : KeyA
IPv6 : FE80::KeyA

ICMPv6 : Neighbour Advertisement
IPv6 Src: FE80::KeyA
IPv6 Dest: FE80::KeyC
IPv6 Target = FE80::KeyA
Nonce=1234
CGA Parameter : KeyA...
Timestamp : April14,2008, 10.00:07
Signature : Message signed with KeyA

Cryptographically Generated Addresses

- IPv6 addresses have a fixed size
  - Unfortunately, only 62 bits are available in host id
    - A 62 bits RSA public-key is not secure
- Solution
  - To secure a binding between a MAC address and an IPv6 address, each host
    - generates its (public $_{key}$,private $_{key}$) key pair
    - uses a special HostId = Hash$_{62}$(public $_{key}$)
    - Signs the Neighbour advertisement by using its private$_{key}$

CNPP/2008.8.

The utilisation of a 62 bits hash instead of a 64 bits hash is necessary because some bits of the host id part of the IPv6 address are reserved. When using CGAs, the two high order bits of the hostid must be set to 0 to indicate that this host id is not globally unique

Cryptographically Generated Addresses (2)

- Issue with CGA
  - A 62 bits hash is not very secure
    - an attacker could use brute-force to find a public-key whose hash is equal to a given value
- Improving CGA security beyond 62 bits
  - Increases the difficulty of computing $Hash_{62}(public_{key})$
  - Define security parameter, Sec=0,1,2,3
    - Encode Sec in 2 high order bits of HostId
    - If Sec=0, then HostId = $Hash_{62}(Random \mid public_{key})$
    - If Sec=1, then
      Find Random : $High_{20}(Hash_{80}(Random \mid public_{key}))=0$
      HostId = $Low_{60}(Hash_{80}(Random \mid public_{key}))$
    - ...

CNPP/2008.8.

This is a simplified description of the computation of a cryptographically generated address. For more details, see :

J. Arkko et al. Securing IPv6 Neighbor and Router Discovery, WiSe 02, September 2002

Lynn, C., Kent, S. and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.

Aura, T., "Cryptographically Generated Addresses (CGA)", RFC3972, March 2005.

Cryptographically Generated Addresses (3)

- Issues with CGA
  - The HostId should not only depend on public key
- Solution
  - CGA depends on several parameters
    - Modifier
      - 16 octets random value
    - Subnet prefix
      - 8 octets
    - Collision Count
      - Incremented each time a duplicate address is found
    - Public key

The structure described above will be send by the endsystem in the neighbor advertisement and will be used by the recipient of the message to check the validity of the signature.

The utilization of CGA by the Neighbor Discovery protocol for IPv6 is defined in :

J. Arkko, J. Kempf, B. Sommerfeld, B. Zill, P. Nikander, Secure Neighbor Discovery (SEND), Internet draft, draft-ietf-send-ndopt-06.txt, July 2004, work in progress

Extensions to ICMPv6

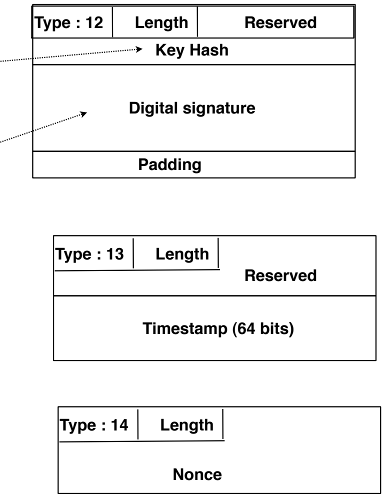
- Signature option

SHA-1 hash (most significant 128 bits)
of the public key used to compute signature.

The signature is computed over the following information :
- random message tag
- 128 bits source address of IPv6 header
- 128 bits destination address of IPv6 header
- Type, Code and Checksum of ICMPv6 header
- NDP message header and options

- Timestamp option
  - used to avoid replay attacks

- Nonce option



See Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

The random message tag is (0x086F CA5E 10B2 00C9 9C8C E001 6427 7C08.) This value was chosen at random by the editor of the above document.

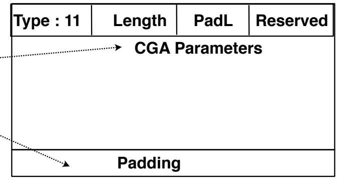
A nonce option is also defined. This option is used to secure the replies sent by routers to neighbour solicitations.

Extensions to ICMPv6 (2)

- CGA option

Parameters used to compute the CGA address

Padding to ensure that CGA option is n*8 bytes



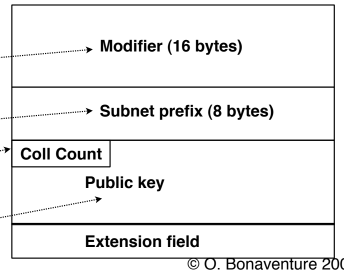
- CGA Parameters

Random value, used to add randomness in the generation of the CGA to improve privacy

The subnet prefix where the address resides

Number of collision in CGA generation

RSA public key, at least 384 bits

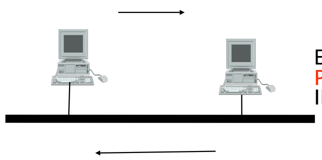


T. Aura, Cryptographically Generated Addresses (CGA) , RFC3972, March 2005

Secure Neighbour Discovery

ICMPv6 : Neighbour solicitation
IPv6 Src: FE80:: Hash(KeyC)
IPv6 Dest: FF02::1
IPv6 Target = FE80::Hash(KeyA)
Nonce=1234
Timestamp : April14,2008, 10.12:01

Ethernet : 0800:CCCC:CCCC
Public key : KeyC
IPv6 : FE80::KeyC



Ethernet : 0800:AAAA:AAAA
Public key : KeyA
IPv6 : FE80::KeyA

ICMPv6 : Neighbour Advertisement
IPv6 Src: FE80::Hash(KeyA)
IPv6 Dest: FE80::Hash(KeyC)
IPv6 Target = FE80::KeyA
Nonce=1234
CGA Parameter : KeyA...
Timestamp : April14,2008, 10.12:07
Signature : Message signed with KeyA

IP version 6

- Outline
  - Motivations for IP version 6
  - IPv6 addressing architecture
  - IPv6 packets
  - ICMP v6
  - DNS support for IP version 6
  - Mobile IP v6
  - IPv6 Multicast

DNSv6

- Three problems to solve
  - How to encode IPv6 addresses in the DNS ?
  - How to support reverse DNS ?
  - How to perform all DNS requests by using only IPv6

DNSv6

- Each DNS messages is composed of resource records (RR) encoded as TLV

- < Name, Value, Type, TTL>
- Types de RR
  - A (IPv4 Address)
    - Name is a hostname and Value an IPv4 address
  - AAAA (IPv6 Address)
    - Name is a hostname and Value an IPv6 address
  - NS (NameServer)
    - Name is a domain name and Value is the hostname of the DNS server responsible for this domain
  - MX (Mail Exchange)
    - Name is a domain name and Value is the name of the SMTP server that must be contacted to send emails to this domain
  - Type CNAME
    - Alias

S. Thomson,C. Huitema, V. Ksinant, M. Souissi, DNS Extensions to Support IP Version 6 , RFC3596 , October 2003

Supporting reverse DNS

- First solution : IP6.INT
  - Encode IPv6 address in reverse order, one character per group of four bits
  - Example
    - 4321:0:1:2:3:4:567:89ab
    - b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.
      1.0.0.0.0.0.0.0.1.2.3.4.IP6.INT.
- Standard solution : IP6.ARPA
  - ARPA=Address and Routing Parameters Area
  - Example
    - 4321:0:1:2:3:4:567:89ab
    - b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.
      1.0.0.0.0.0.0.0.1.2.3.4.IP6.ARPA.

Adding IPv6 addresses to the DNS root

- Took a much longer time than expected
  - Initially DNS root was only reachable via IPv4
    - List of DNS root servers is encoded in one DNS reply
    - All DNS implementations must support DNS replies of 512 bytes, but encoding the 13 IPv4 DNS root servers already consumes 400 bytes. Adding IPv6 for all DNS root servers requires 811 bytes in the reply
  - Several TLD moves quickly to IPv6
    - One IPv6 authoritative server for .be since Sept. 2004
  - DNS was extended to support larger replies
- February 2008
  - 6 root DNS servers now support IPv6
  - IPv6-only hosts can at last use the DNS

The recent introduction of several IPv6 enabled root servers is described in

http://arstechnica.com/news.ars/post/20080205-icann-flips-switch-on-ipv6-dns-root-servers.html

Additional information about the DNS root servers may be found in

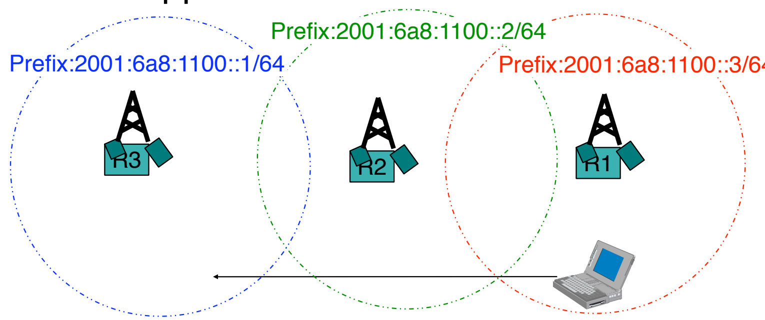
http://www.root-servers.org/

IP version 6

- Outline
  - Motivations for IP version 6
  - IPv6 addressing architecture
  - IPv6 packets
  - ICMP v6
  - DNS support for IP version 6
  - Mobile IP v6
  - IPv6 Multicast

Mobile IPv6

- What happens when a node moves ?



- Mobile receives a new IPv6 address in each subnet
- How to maintain connectivity and preserve TCP/UDP flows while moving ?

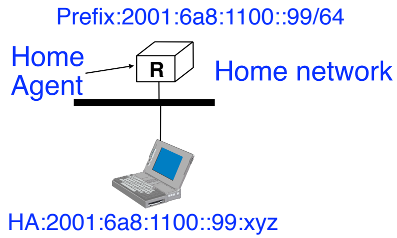
Mobile IPv6 is defined in :
David B. Johnson, Charles E. Perkins, Jari Arkko, Mobility Support in IPv6, Internet draft, draft-ietf-mobileip-ipv6-19.txt, work in progress, Oct 2002

A detailed presentation of Mobile IPv6 may be found in

H. Soliman, Mobile IPv6 : Mobility in a Wireless Internet, Addison Wesley, 2004

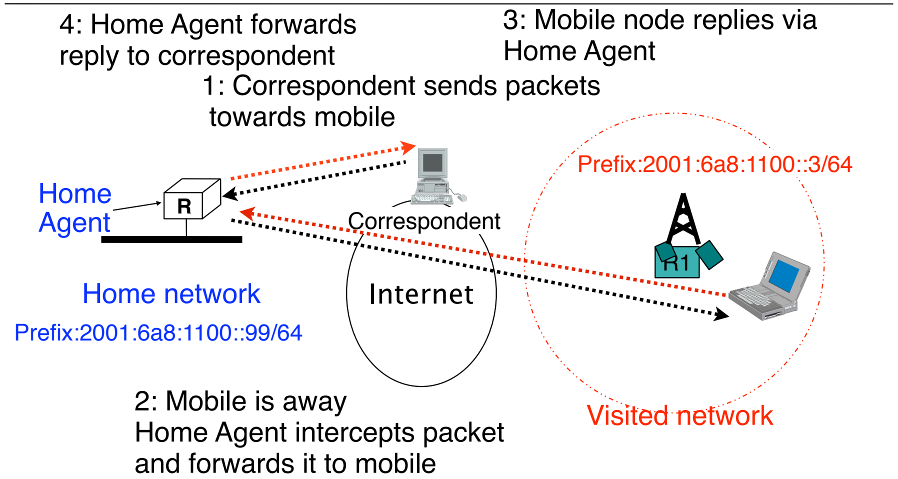
Principle of the solution

- Each mobile node has a home network
  - where it receives packets when not moving
  - where packets will arrive by default when mobile host is moving



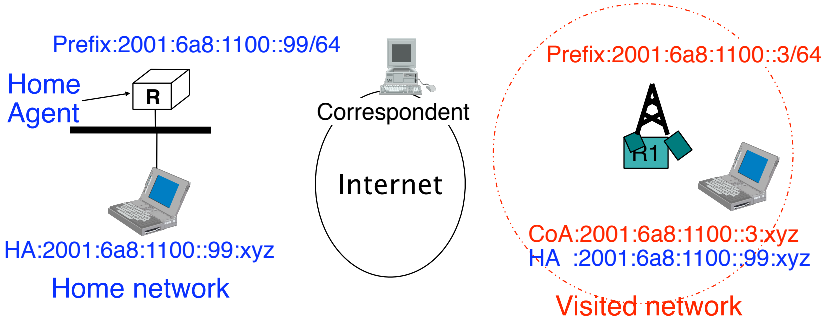
- On home network locate Home Agent which is responsible for
  - receiving packets addressed to mobile node when the mobile node is away
  - forward received packets to mobile node
  - forward to Internet packets from mobile node

A simple example



The Mobile IPv6 addresses

- Two types of addresses
  - A Mobile node has two IPv6 addresses
    - Home Address
      - Its official address, to be used while inside its home network
    - Care Of Address
      - A temporary address, used in combination with the Home address while inside a visited network



Mobile IPv6 packets

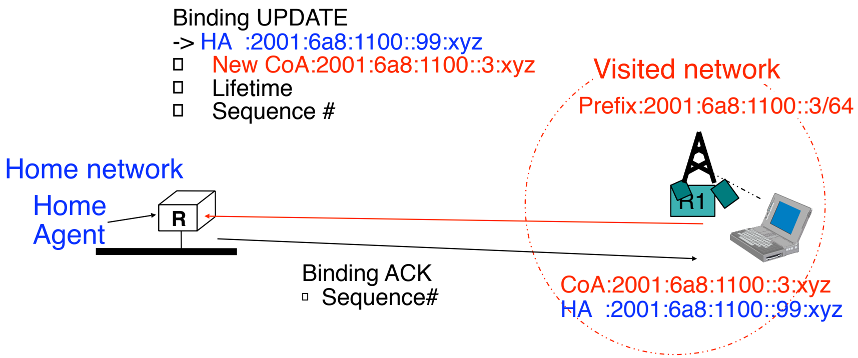
- Define a new IPv6 header extension
  - The mobility header is used to provide mobility information in IPv6 packets

- Two important mobility messages

  - Binding update
    - Sent by a mobile node to inform its Home Agent of its current CareOfAddress

  - Binding acknowledgement
    - Sent to confirm the reception of a Binding Update message

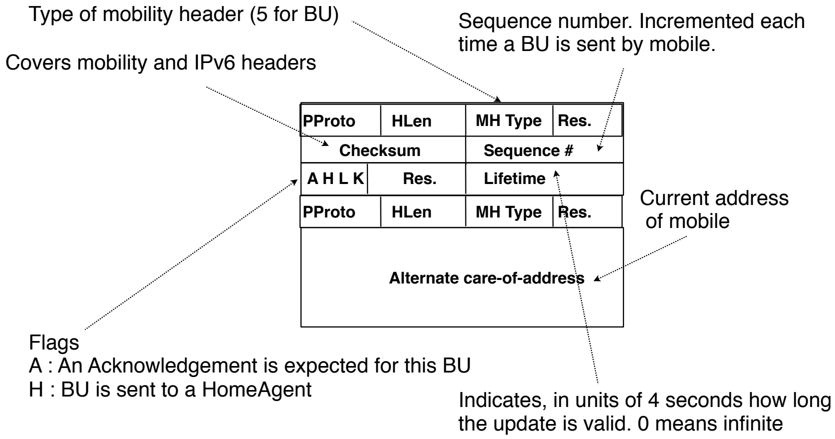
Binding example



- HA's Binding cache
- 2001:6a8:1100::99:xyz <->2001:6a8:1100::3:xyz, lifetime sequence#

The Binding messages

- Binding update



The L flag is used to deal with link local addresses. The K flag is used to deal with security issues. These flags are outside the scope of this presentation.

The Binding messages (2)

- Binding Acknowledgement

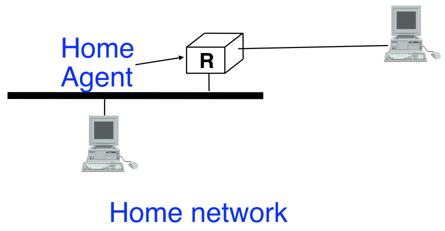
PProto	HLen	MH Type	Res.		
Checksum		Status		K	Res.
Sequence #		Lifetime			

Status
128 : The Binding Update with the provided sequence number was correctly installed
Other values indicate a failure

The K bit flag is used for security reasons but is outside the scope of this presentation.

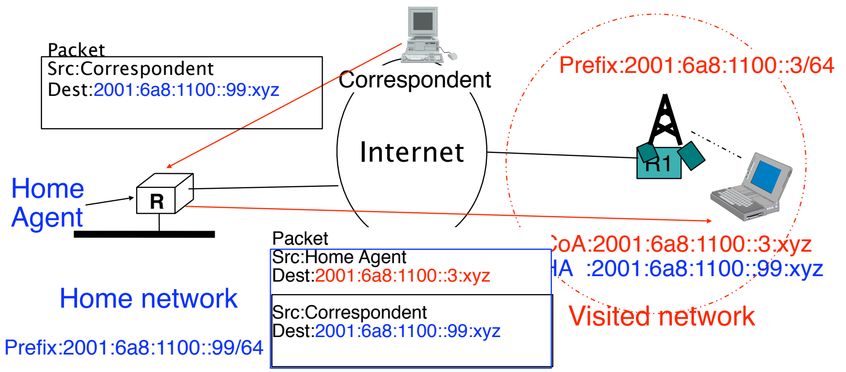
How can the Home Agent capture packets ?

- Principle
  - When the Mobile Node is not in its home network, the Home Agent should behave as if it were the Mobile Node
    - HomeAgent will receive Neighbor solicitations requesting the MAC address corresponding to the Home Address of the Mobile Node
    - HomeAgent will send Neighbor Advertisements instead of the Mobile Node



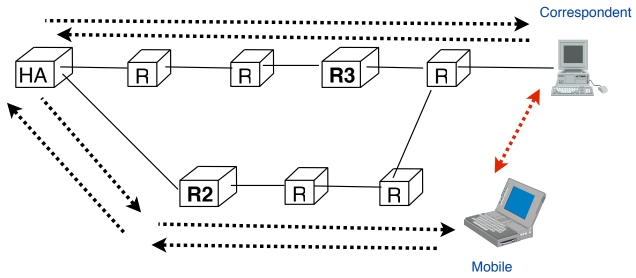
How to send packets to mobile node ?

- Default solution
  - Correspondent node sends to Home Address
  - Home Agent captures packet and sends it to CoA



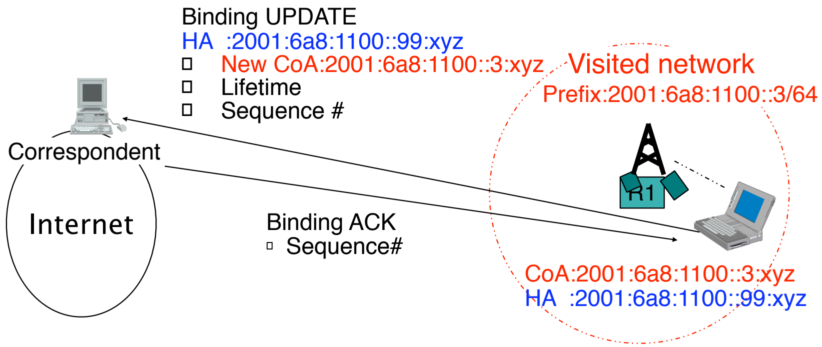
Issue with this tunnel

- Mobile node may be far from Home Agent but topologically close to correspondent



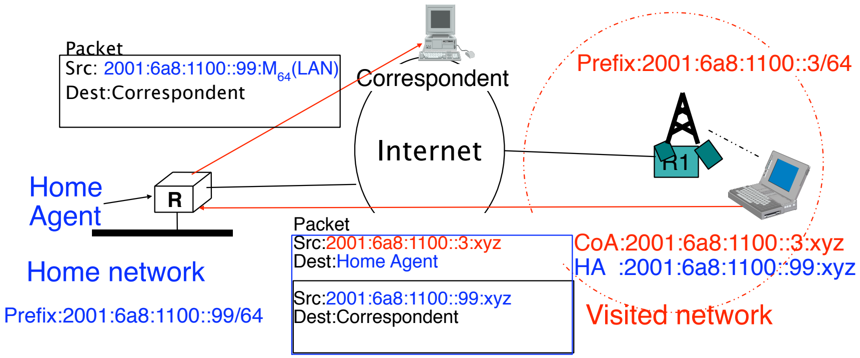
How to send packets to mobile node (2)?

- Route Optimisation
  - Mobile Node sends BU directly to Correspondent
    - MN will need to informs the Correspondent when it moves
  - Correspondent can send packets directly to MN
    - But correspondent needs to maintain a binding cache



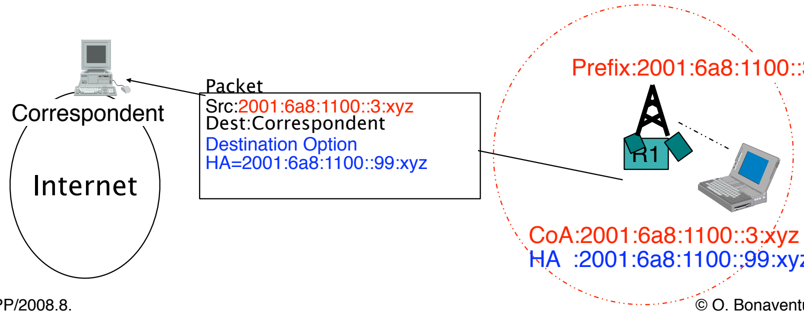
How can the mobile node send packets ?

- By sending tunnelled packets via Home Agent
  - Conforms to ingress/egress policies
  - Drawback
    - Not always efficient



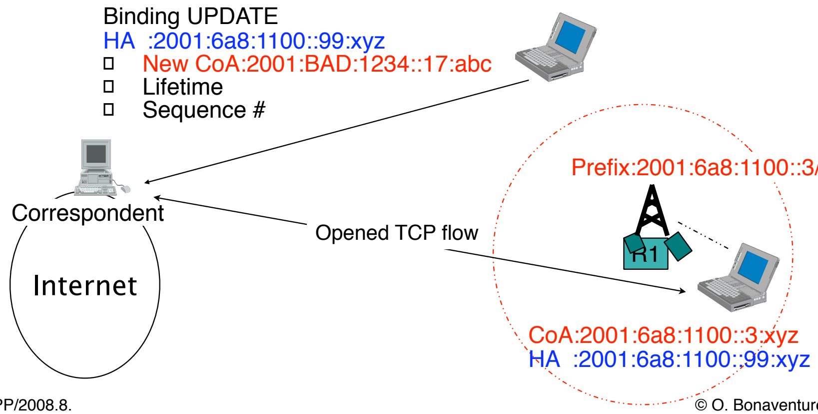
How can the mobile node send packets (2)?

- By using its Care of Address as source
  - Conforms with ingress/egress policies
  - Issue
    - How to avoid breaking TCP connections ?
      - Solution : Home Address in Destination option of all IPv6 packets



Security issues with Mobile IPv6

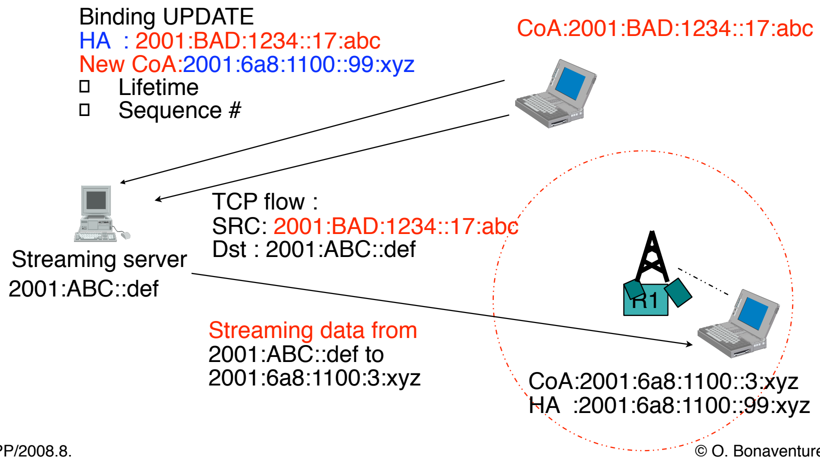
- Mobile IPv6 allows a mobile host to send binding updates to inform correspondent
  - But attackers may also send binding updates...



By sending Binding Updates, it is also possible for the attacker to create a Man in the Middle Attack by putting itself in the middle between the mobile and the correspondent. It is also possible to send Binding Updates pointing to a non-existent address to cause a Denial of Service Attack to make the mobile host unreachable.

Reflection/flooding attack

- How to send lots of packets to a victim ?



IP version 6

- Outline
  - Motivations for IP version 6
  - IPv6 addressing architecture
  - IPv6 packets
  - ICMP v6
  - DNS support for IP version 6
  - Mobile IP v6
  - IPv6 Multicast

IPv6 multicast

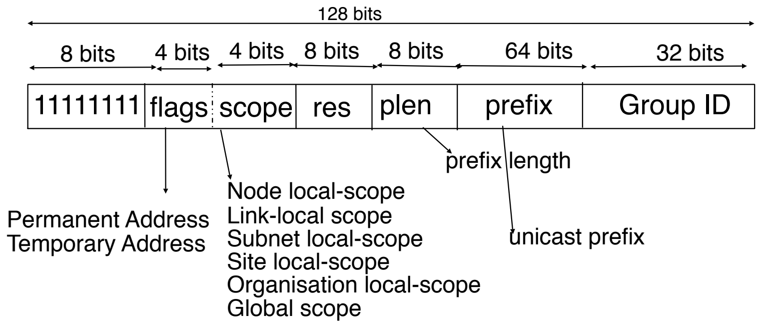
- Differences between IPv4 multicast and IPv6 multicast
  - IPv6 multicast addressing architecture
  - IGMP replaced by Multicast Listener Discovery
  - IPv6 multicast routing protocols are essentially equivalent to IPv4 multicast routing protocols
  - Transmission of IPv6 multicast packets over Ethernet

When transmitting IPv6 multicast packets over Ethernet, the low order 32 bits of the IPv6 multicast address are used to build the Ethernet multicast destination address composed of 33-33 (hexa) as the first two bytes, the low order 4 bytes of the IPv6 multicast destination address

See M. Crawford, Transmission of IPv6 Packets over Ethernet Networks, RFC2464, 1998

IPv6 multicast addressing architecture

- Unicast-Prefix-based IPv6 Multicast Addresses



- Motivation
  - Pragmatic method to allow each prefix to have unique IPv6 multicast addresses

See B. Haberman, D. Thaler, Unicast-Prefix-based IPv6 Multicast Addresses, RFC3306, August 2002

Multicast Listener Discovery

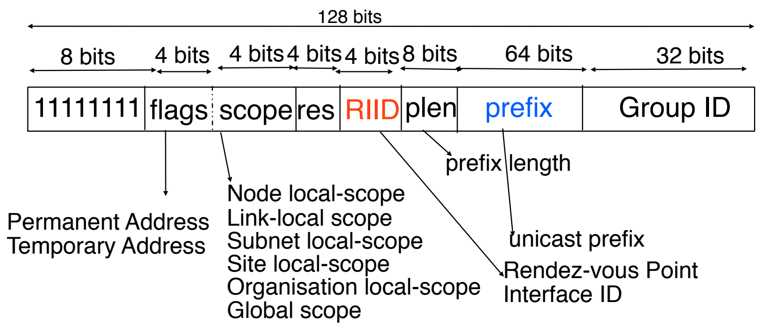
- Multicast Listener Discovery v1
  - Based on IGMPv2
  - Main difference : is part of ICMPv6 instead of being transported directly over IP as IGMP
- Multicast Listener Discovery
  - Based on IGMPv3
  - runs inside ICMPv6

See

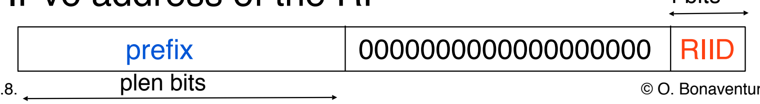
S. Deering, W. Fenner, B. Haberman, Multicast Listener Discovery (MLD) for IPv6, RFC2710, October 1999
R. Vida, Ed., L. Costa, Ed., Multicast Listener Discovery Version 2 (MLDv2) for IPv6, RFC3810, June 2004

IPv6 multicast addressing architecture

- Embedding RP addresses inside IPv6 multicast addresses



- IPv6 address of the RP



See P. Savola, B. Haberman, Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address, RFC3956, November 2004