IPv6 Overview

AUUG-SA
Adelaide
2002-10-09

Glen Turner, Network Engineer
glen.turner@aarnet.edu.au

Australian Academic & Research Network
http://www.aarnet.edu.au/

Topics

Introduction

Addressing

Fields

Transition

Vendor support

Aims

Solve address space shortage

- Running out of addresses
- NAT doesn't scale
- Want new address heirarchies

Reduce size of global routing table

- Hardware lookup
- CPU L2 cache

Aims, con'd

Remove IPv4 cruft

- Gets in the way of hardware forwarding

Build in

- Privacy
- Authentication
- Multicast
- Automated configuration

A lot of the IPv6 solutions have been back-ported to IPv4

Topics

Introduction

Addressing

Fields

Transition

Vendor support

Addressing

128 bits

- 64 for network
- 64 for host, this is usually the EUI of the hardware interface

No address classes

Hosts have multiple addresses, usually multiple addresses per interface

- This is the largest conceptual difference

Notation

Each 16 bits delimited with a colon

- *3FFE:3700:0021:0000:0000:11ff:feab:1234*

Abbreviations

- Leading zeros of half-words can be omitted
- sequences of zero half-words can be abbreviated with ::, but only once
- *3FFE:3700:21::11ff:feab:1234*

Notation, cond

Masks are written using number of relevant bits

- a host
  - *3FFE:3700:21::11ff:feab:1234/48*
- a network
  - *3FFE:3700:21::/48*

Notation, cond

Need to make life easy for IPv4 addresses

- *::129.127.40.3*

rather than

- *::817F:2803*

Notation, cond

Bad news for Perl scripters

- There is no regular expression for determining an IPv6 address
- Need to parse
- Whoops

The canonical form for an IPv6 address is complex

- Shortest address, with decimals if IPv4
- With mask

EUI-64 address?

IEEE's response to need for larger MAC addresses

MAC-48

- Three octets for manufacturer (OUI)
- Three octets for device

EUI-64

- Three octets for manufacturer (company_id)
- Five octets for device

EUI-64 addresses

Convering MAC-48 to EUI-64

- Insert FF-FE into centre
  - AB-CD-EF-12-34-56
  - AB-CD-EF-FF-FE-12-34-56

IPv6 uses EUI-64 addresses as the host part of the IPv6 address

- This allows autoconfiguration
- Privacy issues

Top level address allocation

- ::/8 Reserved
  - ::1/128 Loopback
  - ::/96 IPv4
- 1000::/4 Provider independent
- 2000::/3 Aggregatable unicast
- 8000::/3 Geographical unicast
- FE80::/10 Link local
- FEC0::/10 Site local
- FF00::/8 Multicast
  - FF02::1 All hosts
  - FF02:2 All routers

Address allocation

Address assignment, where needed, is done by APNIC, RIPE and ARIN

- Rules for allocations are basically same as IPv4 rules
  - 90% usage before more address space
  - Addressing plans

Why?

- Once bitten, twice shy

Address allocation and routing

AARNet has 2001:388::/32

- It cannot advertise a smaller allocation

Downstream sites must be given at least a /48

- Allowing each site to have 16,000 subnets
- The subnets are of unlimited size

Autoconfiguration

Addressing allows stateless autoconfiguration

- Router provides 64-bit prefix to host
- Host uses interface MAC address to form 64-bit suffix

Multiple routers can provide prefixes

- Hosts can multihome
- Even across multiple ISPs

Manual configuration

Occassionally want to nail down address suffix

- Privacy
- DNS server

Can configure explicitly or use DHCPv6

Autoconfiguration walkthrough

Power on

Assign each interface a link local address

- Prefix *FE80:0000:0000*
- Suffix EUI-64

Send ICMPv6 Solicitation Message to FF02::2, the All Routers multicast group

Autoconfiguration walkthrough, cond

Listen on multicast group *FF02::1*, the All Hosts group

For each incoming ICMPv6 Router Advertisment

- Use prefix from RA plus the EUI-64 to form an address
- Add that address to the receiving interface

Topics

Introduction

Addressing

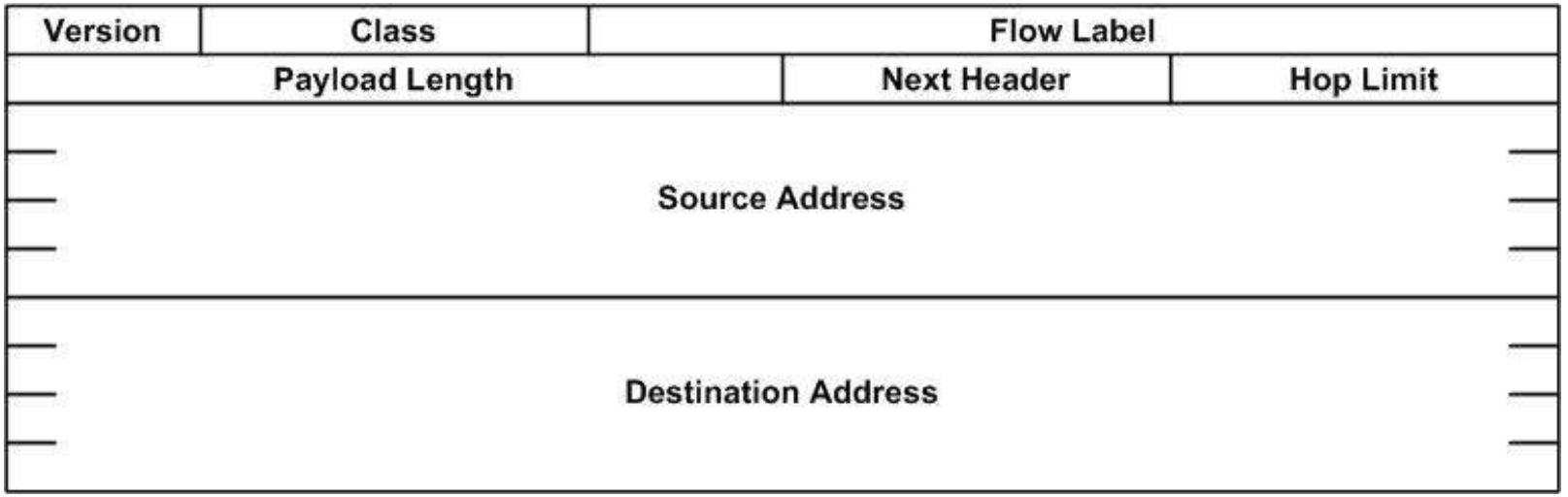
Fields

Transition

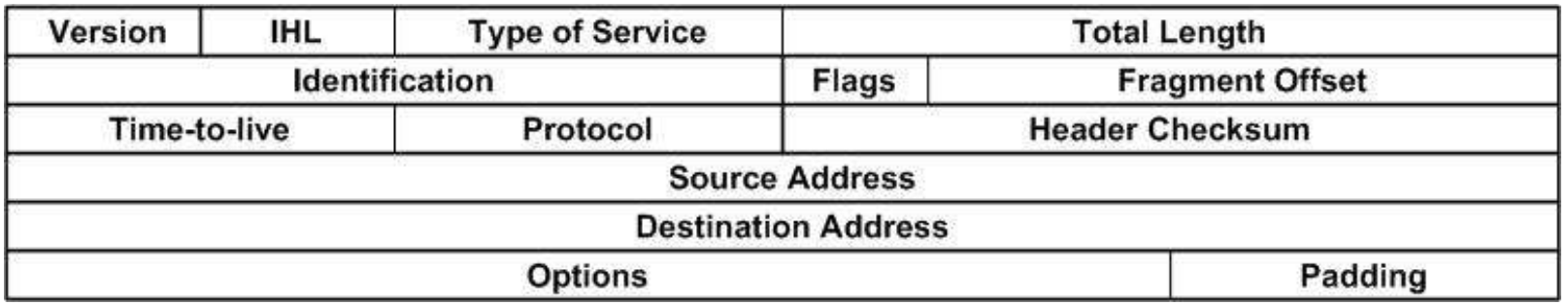
Vendor support

Headers

IPv6



IPv4



What's changed?

All fields fixed lengths

- Options are replaced by extension headers

Header checksum gone

- Rely on Link Layer CRC

No fragmentation

- But explicity MTU discovery support
- Routers return MTU in body of a Packet Too Big ICMP

Field walkthrough

Version (4 bits)

- 6 :-)

Traffic class (8 bits)

- Differentiated services class point

Flow label (20 bits)

- Unique cookie per flow

Payload length (16 bits)

- Rest of packet, in octets

Field walkthough, cond

Next header (8 bits)

- Type of header immediately after IPv6 header
  - Could be a protocol like TCP or UDP
  - Could be a IPv6 extension header

Hop limit (8 bits)

- Renamed TTL

Source address (128 bits)

Destination address (128 bits)

Extension headers

Authentication

Encrypted Security Payload

Destinations Options

Hop-by-hop Options

Routing

Fragmentation

ICMPv6

- Implemented as Extension Header (type 58)
- Must be last Extension Header

ICMPv6 messages

Errors

- Destination unreachable, Packet too big, Time exceeded, Parameter problem

Information

- Echo request, reply

ICMPv6 neighbour discovery

Router advertiment

- Periodic

Router solicitation

- Send a RA now

Neighbour solicitation

- Hello?

Neighbour advertisment

- Response to a NS or an interface MAC address change

Topics

Introduction

Addressing

Fields

Transition

Vendor support

Transition

IPv6 has a huge number of transition strategies

Tunnelling – 6over4

Standard tunnel idea

Put IPv6 packet into IPv4 packet

Doesn't use topology well

Very manual set up

- And pull down, once a real network is in place
- Multicast experience

AARNet uses 6over4 for long-haul links

Tunnelling – 6to4

Your IPv4 address is an IPv6 address

Easy to set up

But

- no DNS
- still a tunnel, so performance sucks
- security nightmare - “accept all incoming tunnels”

Tunnel broker

Automatic establishment of tunnel and DNS

http://www.freenet6.net/

Experiences with native

- Works fine over all media, including wireless
- Got to watch detail of vendor claims of support

DNS

No IPv6 DNS root

- So not practical to have an IPv6-only stack

Reverse DNS is a typing challenge

- So use dynamic DNS

```
6.a.6.3.8.b.e.f.f.f.b.5.6.0.2.0.0.1.0.0.0.0.0.1.8.8.3.0.1.0.0.2.ip6.
   arpa IN PTR gingernut.aarnet.edu.au.

gingernut.aarnet.edu.au. IN AAAA 2001:388:1000:10:206:5bff:feb8:36a6
```

E-mail

Save the curious and run a

- IPv6+IPv4 primary
- IPv4-only secondary

```
$ORIGIN aarnet.edu.au.
aarnet.edu.au. IN MX 10 gingernut
aarnet.edu.au. IN MX 20 arrowroot
gingernut IN A ...
gingernut IN AAAA ...
arrowroot IN A ...
```

E-mail, cond

The curious have enabed IPv6 but have no IPv6 connectivity

- So their connections to mail exchangers which support IPv6 time out
- This is easy to debug with interactive protocols but difficult with store-and-forward protocols like e-mail

Software

Most software used in the network infrastructure now supports IPv6

- BIND
- Apache
- Sendmail

Software, cond

A lot of client software supports IPv6

- Most Unix command line utilities
  - Some have special IPv6 versions. These ensure that *only* IPv6 is used.
    - ping6, traceroute6
- SSH
- Mozilla and IE

Topics

Introduction

Addressing

Fields

Transition

Vendor support

Linux

"IPv6" support in kernel was poor

- no autconfiguration
- no security

USAGI project is fixing this, with integration into 2.4 and 2.5 progressing

- http://www.linux-ipv6.org/
  - kernel patches
  - binaries for popular distributions

BSDs

Demonstration platform for WIDE initiative, so good support

- http://www.wide.ad.jp/

KAME project is continuing development

- http://www.kame.net/

Sun

Solaris

- In production with version 8

The new HP

Tru64 UNIX

- In production with version 5.1

HP-UX

- Download for HP-UX11i

Roadmap fuzzy

IBM

AIX for pSeries (was RS/6000)

- In production with version 5L

zOS for zSeries (mainframe)

- In production with version 1.4
- No OS/390 support, go R10 1.4

OS/400 for iSeries (was AS/400)

- None planned

Linux

Apple

MacOS

- In production with 10.2 (Jaguar)

Microsoft

Windows desktop

- In production with XP
  - SP1 a good idea
- Additionally, developers require
  - SDK from Jan 2000 or later
  - Visual C++ 6.0 or later

Windows server

- Production support anticipated in Windows .NET Server 2003 (upgrade path from Windows 2000 Server)

Juniper

JUNOS

- In production with 5.1

Cisco

IOS

- In production with 12.2(2)T IP Plus
- Backbone routers run 12.0S, so that wasn't much use
- 12.0(22)S runs IPv6, only on dCEF-capable hardware

CatOS

- Never

Further material

AARNet web site

- http://www.aarnet.edu.au/network/design/ipv6/

AARNet Advanced Internet Workshop on IPv6

- Adelaide 29-30 October
- $440 (AARNet members) or $660
  - Covers costs
- Very hands on

That's all folks