IP Next Generation (IPng)

A path to the future.

History

- Why?
  - apparently running out of IP addresses
  - routing table bloat
- August 1990
  - projected exhaustion of Class B space by March 1994
- November 1991
  - Routing and Addressing (ROAD) group formed
  - ROAD report (March ‘92)
    - do CIDR
    - issue call for IPng proposals

History, contd.

- July 1992

  IAB issues "IP version 7"

  IETF issues call for IPng proposals

- July 1993

  ipdecide BOF & IESG plenary

  - IESG take on responsibility making IPng recommendation

    (not let the market decide)

- August 1993

  temporary IETF area formed to consolidate IPng activity

History, contd.

- December 1993

  RFC 1550 call for IPng White Papers
- July 1993

  IPng recommendation
- October 1994

  IESG approved recommendation
- February 1995

  base documents ready for Proposed Standard

IPng Proposals

- multiple working groups
- different approaches to solve addressing and routing problems
- different views on problems
- optimize different aspects of problems
- not right or wrong
- learned from all efforts

Available Timeframe

- Address Lifetime Expectations (ALE) working group
  - Frank Solensky, FTP Software <solensky@ftp.com>
  - Tony Li, Cisco Systems <tli@cisco.com>
- made prediction at Seattle & Toronto IETF meetings
  - 2005 - 2011
- mixed view of confidence level
  - questions on base data & assumes no paradigm shifts
  - routing tables are still going to be a problem

Scope of IPng

- development, testing & deployment will take time
- still we seem to have adequate time in IPv4 address space but not excessive (excluding paradigm shifts)
- can do more than 'just' fix addresses
- use requirements process to determine actual scope of IPng effort

IPng Technical Requirements

- IPng requirements process

  Frank Kastenholz, FTP Software <kasten@ftp.com>

  Jon Crowcroft, UCL <J.Crowcroft@cs.ucl.ac.uk>

- RFC1550 request for white papers
- requirements document

  based on Frank Kastenholz/Craig Partridge draft

  criteria, discussion & time frame

- RFC 1726

IPng Criteria

- at least 10^9 networks, 10^{12} end-systems
  - safer goal 10^{12} nets, 10^{15} end-systems
- conservative routing schemes
- topologically flexible
- high performance
- straightforward transition plan from IPv4
- robust service
- media independent

IPng Criteria, contd.

- datagram service
- autoconfiguration
- secure operation
- globally unique names
- access to standards
- support multicasting
- extensible
- support service classes

IPng Criteria, contd.

- support mobility
- include control protocol (ping etc.)
- support for private networks (tunneling)

Result of Proposal Reviews

- significant flaws seen in all proposals
- revised proposal offered by SIPP WG after reviews
- answers most of the perceived problems
  - routing and addressing
  - transition
  - autoconfiguration
  - source routing support
- synthesis of multiple efforts

Address Length

- hotly discussed issue
- four basic views
  - 8 bytes is enough, more is inefficient
  - 16 bytes is about right, 8 is not enough
  - use 20 byte NSAPs, provide global harmonization
  - variable length gives best safety and efficiency
- many detailed arguments
- consensus is that 16 bytes is enough

IPng description

- expanded from IPv4 addressing capability (16 byte addresses)
- simple header
- support for extension headers and options
- support for authentication and privacy
- support for autoconfiguration
- support for source routes
- simple and flexible transition from IPv4
- flow ID

IPng Addresses

- propose mapping algorithms from and to other environments
- where addresses are globally unique and assigned with regard to network topology
- IETF should work with other organizations for development of such mappings
- common addresses facilitate transition to IPng
- goal to provide a 1:1 mapping between address types (e.g. IPX, NSAP, E164)

IPng Documents

IPng Specification:

R. Hinden, Ed. *IPng Protocol Specification*
draft-hinden-ipng-ipv6-spec-00.txt

Addressing Architecture:

R. Hinden, Ed. *Png Addressing Architecture*
draft-hinden-ipng-addr-00.txt

Y. Rekhter et al *An Architecture for IPv6 Unicast Address Allocation*
draft-rekhter-ipng-arch-IPv6-addr-01.txt

Internet Control Message Protocol:

A. Conta et al *ICMP and IGMP for the Internet Protocol Version 6 (IPv6)*
draft-ietf-sipp-icmp-igmp-00.txt

Transition Mechanisms:

R. Gilligan *Simple Internet Transition Overview*
draft-gilligan-ipv6-sit-overview-01.txt

R. Gilligan et al *Transition Mechanisms for IPv6 Hosts and Routers*
draft-gilligan-ipv6-trans-mech-00.txt

IPng Documents, contd.

Security

R. Atkinson *IPv6 Security Architecture*
draft-atkinson-ipng-sec-00.txt.

R. Atkinson *IPv6 Authentication Header*
draft-atkinson-ipng-auth-00.txt

R. Atkinson *IPv6 Encapsulating Security Payload (ESP)*
draft-atkinson-ipng-esp-00.txt

Domain Name System

S. Thomson et al *DNS Extensions to support IP version 6*
draft-thomson-ipng-dns-00.txt, October 1994.

Auto Configuration

S. Thomson et al *IPv6 Address Autoconfiguration Protocol*

IPng Documents, contd.

Program Interfaces

R. Gilligan et al *IPv6 Program Interfaces for BSD Systems*

draft-gilligan-ipv6-bsd-api-00.txt

OSI NSAP Mapping

B. Carpenter et al *Recommendations for OSI NSAP usage in IP6*

draft-carpenter-ip6-nsap-map-00.txt

Routing

T. Li et al *ERP IPv6 Routing Header*

draft-sdr-ipv6-format-00.txt

G. Malkin *RIP for IPv6*

draft-ietf-ripv2-ripng-00.txt

Y. Rekhter et al *IDRP for IPv6*

draft-ietf-idr-idrp-v6-00.txt

F. Baker et al *OSPF IPv6 Extensions*

draft-ietf-ospf-ipv6-ext-00.txt

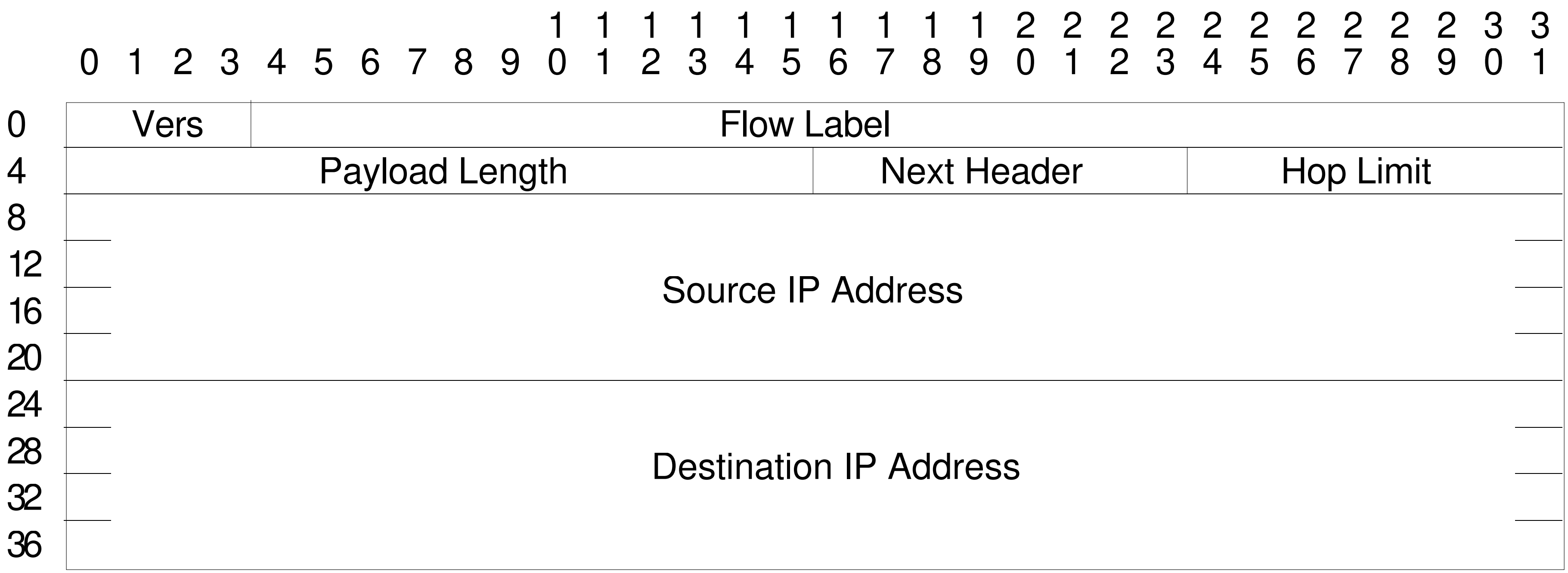
Security

- there is a recognition that the Internet needs strong security
- the goal is to provide strong protection as a matter of course throughout the Internet
- separate use of encryption from use of authentication
- support of authentication header is required in IPng
- support of privacy header is required in IPng

Security, contd.

- a key management infrastructure is required but outside of IPng effort
- inclusion of cryptography for authentication and/or privacy may add to the cost and impact the performance of an implementation, but it is worth the cost
- IETF work underway to embed signed keys in DNS
- need framework for firewalls under IPng

IPv6 Header



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0
4
8
12
16
20
24
28
32
36
Vers
Flow Label
Payload Length
Next Header
Hop Limit
Source IP Address
Destination IP Address

IPv6 Header Fields

Version - Internet Protocol version # = 6 (4-bit field)

Flow Label - used to label packets requesting special handling by routers within a network (28-bit field)

Payload Length - length of the packet following the IPv6 header, in octets (16-bit unsigned integer)

if = 0 the actual packet length in Hop-by-Hop option

Next Header - the type of header immediately following the IPv6 header (8-bit selector field)

Hop Limit - decremented by 1 by each node that forwards the packet.

packet discarded if Hop Limit is decremented to zero

(8-bit unsigned integer)

IPv6 Header Fields, contd.

Source Address - an address of the initial sender of the packet (128 bit field)

Destination Address - an address of the intended recipient of the packet (128 bit field)

- may not be the ultimate recipient, if Routing Header is present

IPv6 Header Changes from IPv4

- longer address - 32 bits -> 128 bits
- fragmentation fields moved to separate header
- header checksum field eliminated
- header length field eliminated (fixed length header)
- length field excludes IPv6 header
- “Time to Live” -> “Hop Limit”
- “Protocol” -> “Next Header”
- 64 bit field alignment
- added Flow Label
- excluded TOS bits

IPv6 Extension Headers

- less used functions moved to Extension Headers
- only present when needed
- extensible

  Hob-by-Hop

  Routing

  Fragment

  Authentication

  Privacy

  End-to-End

Hop-by-Hop Options Header

- used to carry optional information
- examined by every node along packet's delivery path
- e.g., extended packet length option

IPv6 Header Option Handling

- highest-order two bits of each option specify the action to be taken if unknown option
  - 00 - skip this option
  - 01 - discard the packet
  - 10 - discard the packet & send ICMP message
  - 11 - undefined

  eases introduction of new options
- third-highest-order bit in Hop-by-Hop options

  include option in the Authentication integrity assurance computation (option data that changes en route must be excluded)

Routing Header

- lists one or more intermediate nodes to be "visited" on the way to a packet's destination
- nodes or clusters
- designed to support SDRP

Fragment Header

- used to send payloads larger than path MTU to destination
- fragmentation performed only by source nodes

Authentication Header

- provides authentication and integrity assurance
- uses Security Assoc. ID (SAID)

  identifies to the receiver(s) the pre-established security association to which this packet belongs. (32 bit field)

Privacy Header

- provides confidentiality by encrypting data
- data can be
  - transport-layer (e.g. UDP or TCP) frame
  - entire IPv6 datagram
  - part of IPv6 datagram
- must be the last non-encrypted field in a packet (if present)
- uses Security Association Identifier (SAID)

End-to-End Option Header

- used to carry optional information
- examined only by destination node
- same format as Hop-by-Hop Option header

Packet Size Issues

- minimum link MTU = 576 bytes
  - IPv4 = 68 bytes
- links with MTU < 576 must do link-layer fragmentation & reassembly
- IPv6 nodes may avoid Path MTU Discovery by limiting packets to 576 bytes
- minimum reassembly buffer = 1500

Flow Label

1	3	24
T	TC	flow ID

- T - time sensitivity

  0 = yes

  1 = no

- TC - Traffic Class

  type of flow

- Flow ID - random, unique-to-source value

  combined with source address to identify traffic flow

IPv6 Address Representation

- HEX in blocks of 16 bits

  ABFE:76B3:0000:0000:0000:34DE:3421:0012
- leading zero suppression

  ABFE:76B3:0:0:0:34DE:3421:12
- compressed format removes strings of 0s

  ABFE:76B3::34DE:3421:12

IPv6 Address Types

Allocation	Prefix (binary)	Fraction
reserved	0000 0000	1/256
reserved	0000 0001	1/256
NSAP Allocation	0000 001	1/128
IPX Allocation	0000 010	1/128
reserved	0000 011	1/128
reserved	0000 1	1/32
reserved	0001	1/16
reserved	001	1/8
provider-based unicast	010	1/8
reserved	011	1/8
reserved for geographic	100	1/8
reserved	101	1/8
reserved	110	1/8
reserved	1110	1/16
reserved	1111 0	1/32
reserved	1111 10	1/64
reserved	1111 110	1/128
local use address	1111 1110	1/256
multicast address	1111 1111	1/256

IPv6 Unicast Address Examples

- Address Autoconfiguration example

n	m	48
subscriber prefix	subnet ID	node ID

- global provider address example

3	n	m	p	125-n-m-p
010	provider prefix	subscriber ID	subnet ID	node ID

IPv6 Unicast Address Examples, contd.

- local use address example

8	n	m	120-n-m
1111 1110	0000....0000	subnet ID	node ID

- loopback address

  FE00:0:0:0:0:0:0:1 (FE00::1)

- unspecified address

  0:0:0:0:0:0:0:0 (0::0)

IPv6 Unicast Address Examples, contd.

- IPv4-only address example

80	16	32
0000....0000	0000	IPv4 address

- IPv4 compatible IPv6 address example

80	16	32
0000....0000	FFFF	IPv4 address

Other IPv6 Address Examples, contd.

- cluster address example

n	128-n
cluster prefix	00000.....00000

- IPv6 multicast address example

8	4	4	112
1111 1111	flags	scop	GID

low order flag bit

0 - permanent

1 - non-permanent

OSI NSAPs & IPv6

- basic recommendation
- use NSAP with imbedded IPv6 address
- use normal IPv6 address architecture
- if mapping needed
  - can map some NSAPs into 16 byte IPv6 address
  - not encouraged
  - requires routing exchange between environments

IPv6 Transition Goals

- allow incremental upgrade from IPv4 hosts to IPv6
- few sequence dependencies
- support what vendors will do
- allow IPv4-only hosts to talk to IPv6-only hosts
- finish before IPv4 addresses run out

IPv6 Transition Techniques

- IPv4 compatible addresses
- IPv4 address embedded in IPv6 address
- IPv6 in IPv4 encapsulation
- tunnel IPv6 across IPv4 topology
- IPv4 <-> IPv6 header translation
  optional

IPv6-in-IPv4 Encapsulation

- allows IPv6 hosts to exchange traffic over IPv4 networks
- 'sending rules' tell hosts & routers when to encapsulate
- use of embedded IPv4 addresses allow tunnel auto-configuration
- mostly used host-to-host & router-to-host
- encapsulated by IPv4 source node
- IPv4 ICMP errors return to the right place

IPv6/IPv4 Header Translation

- allows IPv6-only hosts to exchange traffic with IPv4-only hosts
- requires translating router within network
- algorithmic mapping of addresses
- translation discouraged by many

Address Autoconfiguration

- three types of autoconfiguration
  - server-less
  - state-less server
  - state-full server
- IPng Address Autoconfiguration effort deals with 1st two
- DHCP deals with state-full server
- security policy an issue
- aim to minimize host knowledge of routing