The Asia Pacific Regional Internet Conference on Operational Technology (APRICOT '96)

17 January 1996

THE NEW VERSION OF IP

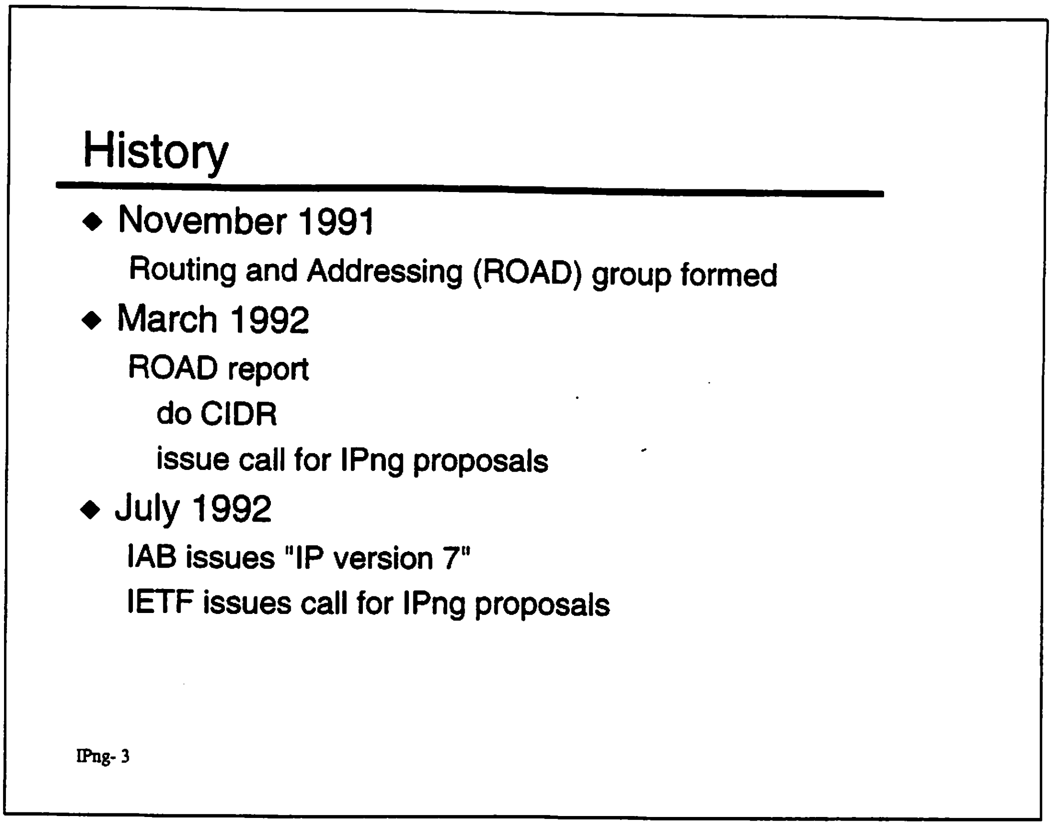
SCOTT BRADNER

IPv6
IP Next Generation (IPng)

A path to the future.

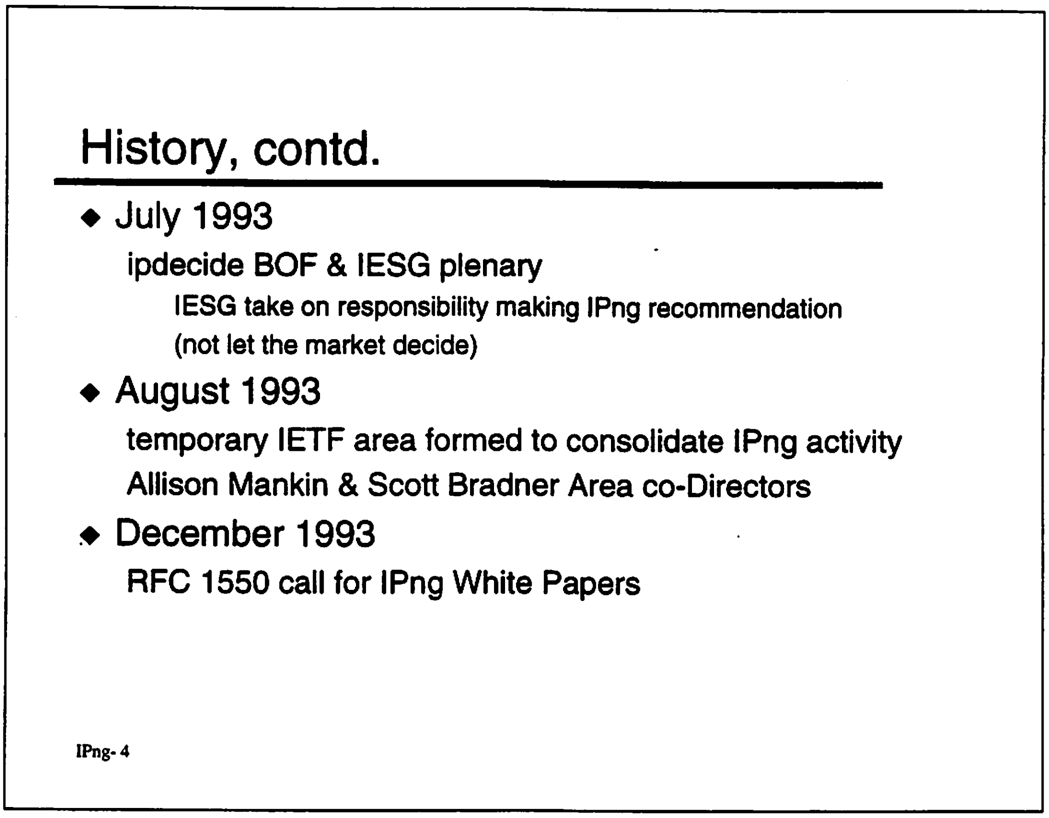
Background - The Need For An IPng

- August 1990
  - projected exhaustion of Class B space by March 1994
  - misrepresented as running out of all IP addresses
- 32 bit address space can identify 4 billion hosts
  - assignment inefficiency reduces utilization (RFC-1715)
  - use of classfull addresses reduces efficiency
- routing table bloat
  - table space increasing faster than memory technology
  - thus could not just use multiple Class C addresses



History

- November 1991
  - Routing and Addressing (ROAD) group formed
- March 1992
  - ROAD report
    - do CIDR
    - issue call for IPng proposals
- July 1992
  - IAB issues "IP version 7"
  - IETF issues call for IPng proposals



History, contd.

- 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
  - Allison Mankin & Scott Bradner Area co-Directors
- December 1993
  - RFC 1550 call for IPng White Papers

History, contd.

- July 1994
  IPng recommendation
- October 1994
  IESG approved recommendation
- July 1995
  multiple interoperable implementations
- September 1995
  base documents approved as Proposed Standards
- December 1995
  base documents issued as RFCs

Directorate

J. Allard - Microsoft	<jallard@microsoft.com>
Steve Bellovin - AT&T	<smb@research.att.com>
Jim Bound - Digital	<bound@zk3.dec.com>
Ross Callon - Wellfleet	<rcallon@wellfleet.com>
Brian Carpenter - CERN	<brian.carpenter@cern.ch>
Dave Clark - MIT	<ddc@lcs.mit.edu >
John Curran - NEARNET	<curran@nic.near.net>
Steve Deering - Xerox	<deering@parc.xerox.com>
Dino Farinacci - Cisco	<dino@cisco.com>
Paul Francis - NTT	<francis@slab.ntt.jp>
Eric Fleischmann - Boeing	<ericf@atc.boeing.com>
Mark Knopper - Ameritech	<mak@aads.com>
Greg Minshall - Novell	<minshall@wc.novell.com>
Rob Ullmann - Lotus	<ariel@world.std.com>
Lixia Zhang - Xerox	<lixia@parc.xerox.com>

IPng Proposals

- three proposals active when IPng area formed
  - CATNIP
  - SIPP
  - TUBA

CATNIP

- Common Architecture for Next-generation Internet Protocol
  - chair: Vladimir Sukonnik
  - developed from TP/IX working group
- The objective is to provide common ground between the Internet, OSI, and the Novell protocols, as well as to advance the Internet technology to the scale and performance of the next generation of internetwork technology.
- document authors:
  - Michael McGovern, Robert Ullmann

SIPP

- Simple Internet Protocol Plus

  chairs: Robert Hinden, Steve Deering, Paul Francis

  developed from merger of IPIP into IPAE which then merged with SIP then finally with PIP past WG chairs: D. Crocker, C. Huitema
- SIPP is a new version of IP which is designed to be an evolutionary step from IPv4. It is a natural increment to IPv4. SIPP is designed to run well on high performance networks (e.g., ATM) and at the same time is still efficient for low bandwidth networks (e.g., wireless). In addition, it provides a platform for new internetfunctionality that will be required in the near future.
- document authors:

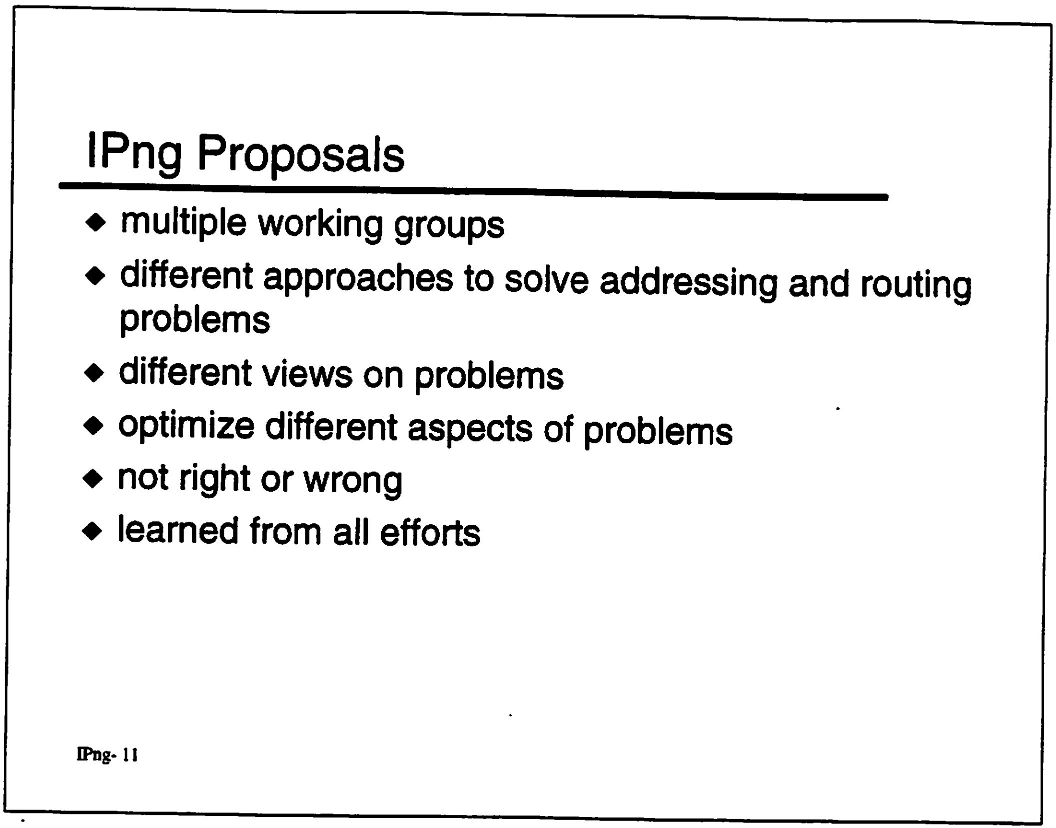
  R. Atkinson, J. Bound, D. Crocker, S. Deering, P. Francis, P. Ford, R. E. Gilligan, R. Govindan, R. Hinden, C. Huitema, T. Li, E. Nordmark, Y. Rekhter, W. A. Simpson, S. Thomson

TUBA

- TCP/UDP Over CLNP-Addressed Networks

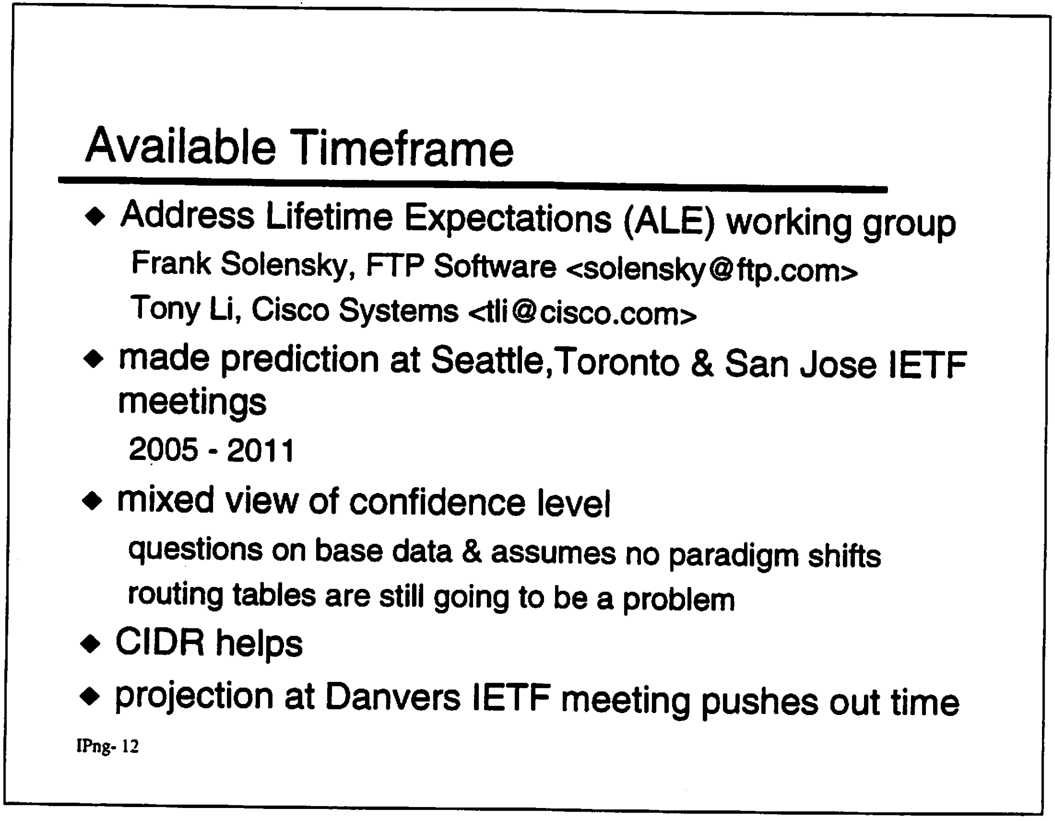
  chairs: Mark Knopper, Peter Ford
- The TUBA effort will expand the ability to route Internet packets by using addresses which support more hierarchy than the current Internet Protocol (IP) address space. TUBA specifies the continued use of Internet transport protocols, in particular TCP and UDP, but specifies their encapsulation in ISO 8473(CLNP) packets. TUBA seeks to upgrade the current system by a transition from the use of IPv4 to ISO/ IEC 8473 (CLNP) and the corresponding large Network Service Access Point address space.
- document authors:

  R. Callon, P. Ford, K. R. Glenn, D. Katz, M. Knopper, D. Marlow, D. Piscitello, Y. Rekhter, J. West (and a fleet of ISO docs)



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



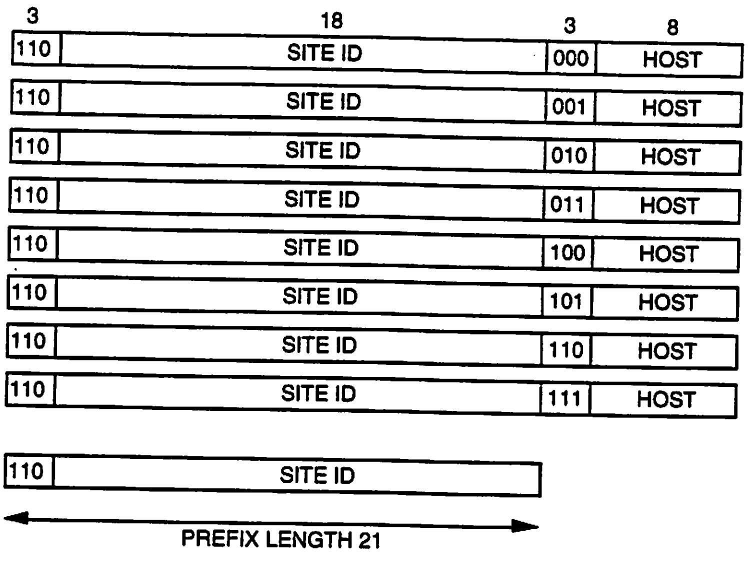
Available Timeframe

- 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 & San Jose 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
- CIDR helps
- projection at Danvers IETF meeting pushes out time

Classless InterDomain Routing (CIDR)

- aggregate routing information
- assign addresses in power-of-two chunks
- advertise power-of-two sized chunk of address space per entry
  - all of a provider's customers can be aggregated into one advertisement
  - reduce size & rate of growth of routing table

CIDR, contd.



3	18	3	8
110	SITE ID	000	HOST
110	SITE ID	001	HOST
110	SITE ID	010	HOST
110	SITE ID	011	HOST
110	SITE ID	100	HOST
110	SITE ID	101	HOST
110	SITE ID	110	HOST
110	SITE ID	111	HOST

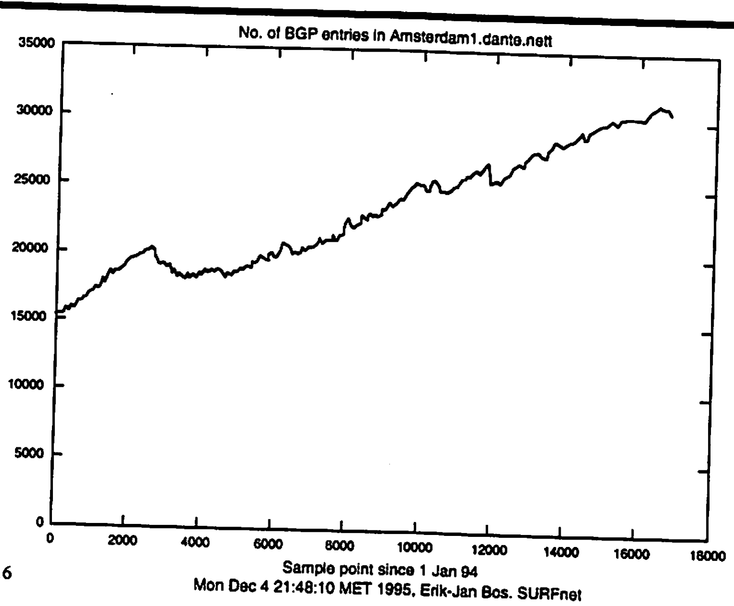
110	SITE ID

PREFIX LENGTH 21

CIDR, contd.

- some issues
  - assumes customers renumber to provider address range
  - tends to bind customer to a provider
  - problem with multi-homed customers
- it works, up to a point

Routing Table Size



Mon Dec 4 21:48:10 MET 1995, Erik-Jan Bos, SURFnet

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

RFC1550 White Papers

Adamson, B.	Tactical Radio Frequency Communication Requirements for IPng, RFC 1677
Bellovin, S.	On Many Addresses per Host, RFC 1681
Bellovin, S.	Security Concerns for IPng, RFC 1675
Bound, J.	IPng BSD Host Implementation Analysis, RFC 1682,
Brazdziunas, C.,	IPng Support for ATM Services, RFC 1680
Britton, E. et al	IPng Requirements of Large Corporate Networks, RFC 1678
Brownlee, J.,	Accounting Requirements for IPng, RFC 1672
Carpenter, B.,	IPng White Paper on Transition and Other Considerations, RFC 1671
Chiappa, J. N.	IPng Tech. Req. Of the Nimrod Routing and Addressing Architecture, RFC 1753
Clark, R. et al	Multiprotocol Interoperability In IPng, RFC 1683,
Curran, J.	Market Viability as a IPng Criteria, RFC 1669
Estrin, D. et al	Unified Routing Requirements for IPng, RFC 1668
Fleischman, E.,	A Large Corporate User's View of IPng, RFC 1687
Green, D et al	HPN Working Group Input to the IPng Requirements Solicitation, RFC 1679
Ghiselli, A. et al	INFN Requirements for an IPng, RFC 1676
Heagerty, D.	Input to IPng Engineering Considerations, RFC 1670
Simpson, W.	IPng Mobility Considerations, RFC 1688
Skelton, R.,	Electric Power Research Institute Comments on IPng, RFC 1673
Symington, S. et al	Modeling and Simulation Requirements for IPng, RFC 1667
Taylor, M.	A Cellular Industry View of IPng, RFC 1674
Vecchi, M.	IPng Requirements: A Cable Television Industry Viewpoint, RFC 1686

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
- datagram service
- autoconfiguration

IPng Criteria, contd.

- secure operation
- globally unique names
- access to standards
- support multicasting
- extensible
- support service classes
- support mobility
- include control protocol (ping etc.)
- support for private networks (tunneling)

Result of Proposal Reviews

- significant flaws seen in all offered 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

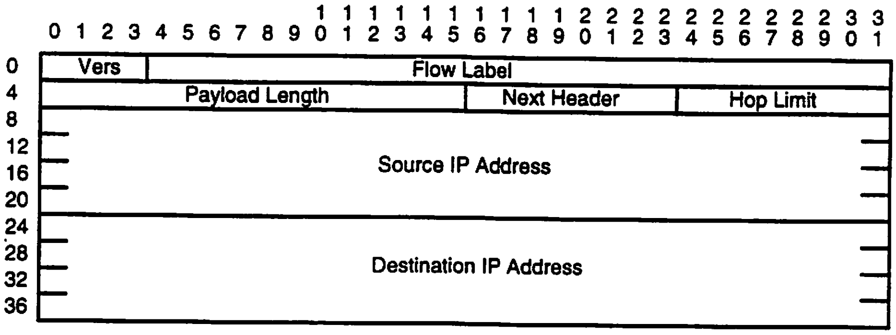
IPv6 Features

- 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

IPv6 Terminology

◆ node	module that implements IPv6
◆ router	node that forwards IPv6 packets not addressed to the node
◆ host	a node that is not a router
◆ neighbor	node on same link
◆ interface	a node's attachment to a link
◆ address	IPv6 identifier for an interface or set of interfaces if they are seen as one logical interface or set of interfaces if anycast address

IPv6 Header



IPv4 Header

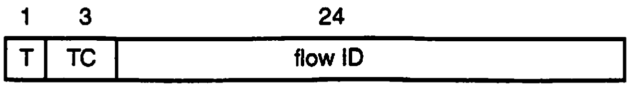
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

0	Vers	Hlen	Pre	D	T	R	C		Total Length
4	Identifier		N	M	Fragment Offset				
8	TTL	Protocol	Header Checksum						
12	Source IP Address								
16	Destination IP Address								
20	Options	Padding							

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
- removed TOS bits

Flow Label

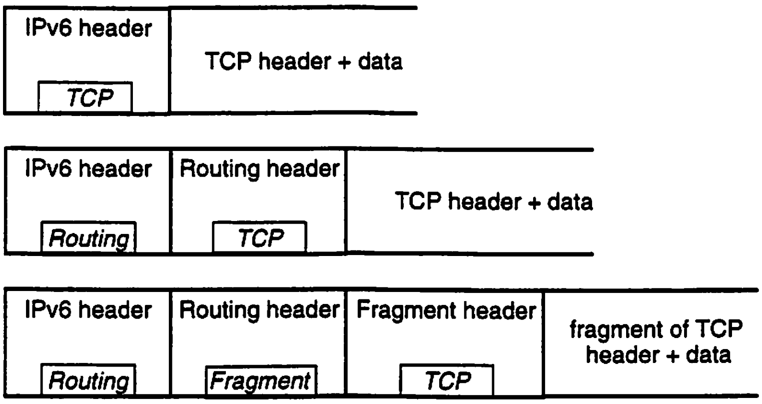


- T - time sensitivity
  0 = yes
  1 = no
- TC - Traffic Class
  type of flow
- Flow ID - random, unique-to-source value
  combined with source address to identify traffic flow

IPv6 Extension Headers

- less used functions moved to Extension Headers
- only present when needed
- only looked at by node with address in Destination Address (except Hop-by-Hop Options)
- extensible
  - Hop-by-Hop Options
  - Routing
  - Fragment
  - Authentication
  - Privacy
  - Destination Options

IPv6 Extension Headers, contd.



IPv6 header | *TCP* | TCP header + data

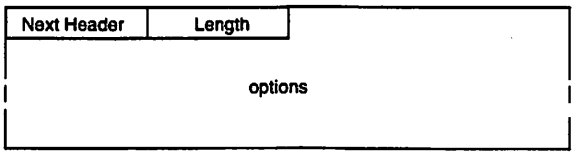
IPv6 header | *Routing* | Routing header | *TCP* | TCP header + data

IPv6 header | *Routing* | Routing header | *Fragment* | Fragment header | *TCP* | fragment of TCP header + data

---

Hop-by-Hop and Destination Options

- contain one or more options



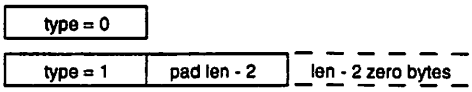
Next Header | Length

options

option type | data length | option data

- pad options

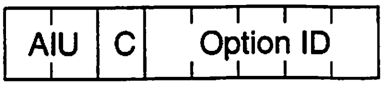
option header must be multiple of 8 bytes



type = 0

type = 1 | pad len - 2 | len - 2 zero bytes

Header Option Handling


AIU | C | Option ID

- AIU - action to be taken if unknown option
  - 00 - skip this option
  - 01 - discard the packet
  - 10 - discard the packet & send ICMP error message
  - 11 - same as 10 except send message only if destination was not a multicast address

  eases introduction of new options
- C - set if option data changes en-route
  (Hop-by-Hop Options only)

  include option in the Authentication integrity assurance computation

Jumbogram Option

- if header length field in IPv6 header = 0
- use jumbogram option in Hop-by-Hop options header to find actual datagram length
- must not also use Fragment Header


type = 194 | opt dat len = 4 | datagram payload length

Fragment Header

- path MTU discovery recommended
- if required, Fragment Header can be used by packet source (routers do not fragment)

Next Header	Length	Fragment Offset	00M
Packet Identifier			

M = more fragments

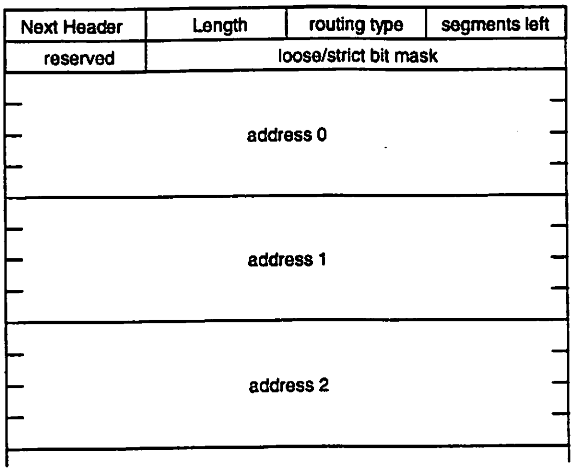
note - if a router is the start of a tunnel, it is the packet source and could fragment

reassembly buffer size 1500 bytes min

Path MTU

- find minimum MTU of any link in a path
- all links must support a MTU of at least 576 Bytes
  - else link-specific fragmentation & reassembly must be used
- send packets to destination using MTU of local link
  - if no MTU info recorded for path
  - reduce size if "packet too big" error returned
  - try again until destination reached
  - "packet too big" ICMP message includes MTU of next hop link - that MTU used for next try

Routing Header



Address Types

- unicast (single destination)
  - global
  - compatible (IPv4, IPX, NSAP, X.121 ...)
  - site-local
  - link-local
- multicast (multiple destinations)
- anycast (nearest destination)
  - prefix with trailing zeroes

Addresses and Interfaces

- an IPv6 interface may have more than one address
  - usually has the same node part with different prefixes
  - link-local address
  - site-local address
  - multiple global addresses
- a single IPv6 unicast address may be assigned to multiple physical interfaces on a node only if that node "sees" the interfaces as a single logical interface

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
- IPv4-compatible
  - 0:0:0:0:0:0:128.103.202.40
  - also - ::128.103.202.40

IPv6 Address Prefixes

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 unicast	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
  0:0:0:0:0:0:0:1 (0::1)
- unspecified address
  0:0:0:0:0:0:0:0 (0::0)

IPv6 Unicast Address Examples, contd.

- IPv4 compatible IPv6 address example

80	16	32
0000....0000	0000	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
  - carry NSEL in destination option
- can also use truncated NSAP address
  - carry full NSAP in destination option
  - mismatch between NSAP areas & IP subnets can cause a problem

IPng & Other 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)

Specific IPv6 Multicast Addresses

- all-nodes multicast address
  - link-local scope address to reach all IPv6 nodes
    - FF02::1
- all-routers multicast address
  - link-local scope address to reach all IPv6 routers
    - FF02::2
- solicited-node multicast address
  - link-local scope address based on target address
  - FF02:0:0:0:0:1 + low-order 32 bits of target IP address
  - limits number of nodes that receive neighbor solicitations

Neighbor Discovery

- router send out advertisements
  - lists prefix(s) for link
  - say if host can use prefix to create global address
  - say if host can use prefix to determine 'on-link'
  - say if host must use DHCPng to get address
- if host can use prefix to create global address
  - host appends 'MAC' address to prefix
  - checks for duplicate addresses
- host MAC addresses resolved with ARP-like request/response procedure
  - sent to multicast address formed from dest IP address

Router Advertisement

<table>
<tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>0</td><td colspan="8">Type = 134</td><td colspan="8">Code = 0</td><td colspan="16">Checksum</td></tr>
<tr><td>4</td><td colspan="8">Max Hop Limit</td><td>M</td><td>O</td><td colspan="6">reserved</td><td colspan="16">Router Lifetime</td></tr>
<tr><td>8</td><td colspan="32">Reachable Time</td></tr>
<tr><td>12</td><td colspan="32">Retrans Timer</td></tr>
<tr><td>16</td><td colspan="11">Options</td></tr>
</table>

IP Destination Address - source address of router solicitation or all-nodes multicast address

Max Hop Limit - hop count for nodes to use (if non-zero)
M - if =1 then node must use DHCPng in addition to autoconf
O - if =1 then node must use DHCPng to get other information
Router Lifetime - if non 0, then router can be used as a default router for N sec
Reachable Time - time in msec for dead node detection timeout
Retrans Timer - time in msec between Neighbor Solicitation messages

Legit Options Source link-layer address, MTU, prefix information

Router Solicitation

```
                       1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
0  |    Type = 133     |     Code = 0      |          Checksum             |
4  |                              reserved                                 |
8  |      Options    ...
```

IP Destination Address - all-routers multicast address

Legit Options - Source link-layer address

Neighbor Solicitation

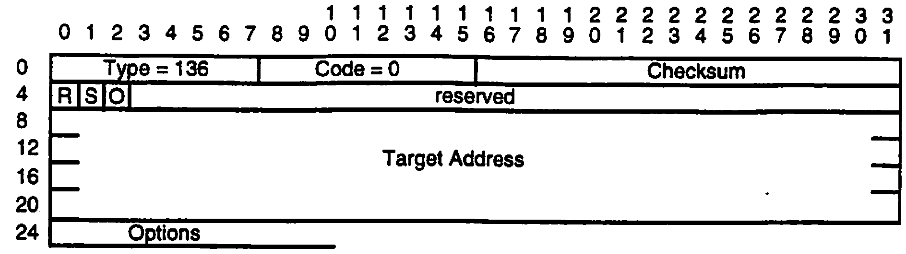
```
                       1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
0  |    Type = 135     |     Code = 0      |          Checksum             |
4  |                              reserved                                 |
8  |                                                                       |
12 +-                                                                     -+
   |                           Target Address                              |
16 +-                                                                     -+
   |                                                                       |
20 +-                                                                     -+
   |                                                                       |
24 |      Options    ...
```

IP Destination Address - solicited-node multicast address for target or target address

reserved	- must be 0
Target Address	- IPv6 Address of the target of the solicitation must not be a multicast address

Legit Options Source link-layer address

Neighbor Advertisement

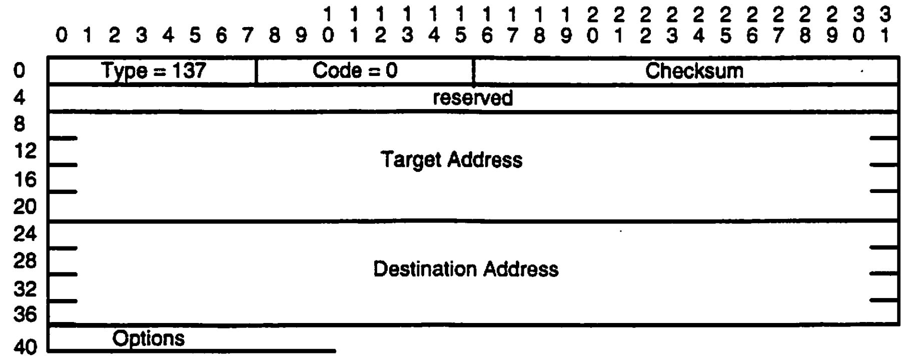


IP Destination Address source address from solicitation
all-nodes multicast if unspecified source address
all-nodes multicast if unsolicited

R - if =1 sender is a router
S - if =1 message is in response to a solicitation
O - if =1 override any existing cache entries
reserved - must be 0
Target Address - IPv6 Address of the target of the solicitation
or node whos address changed

Legit Options Target link-layer address

Redirect



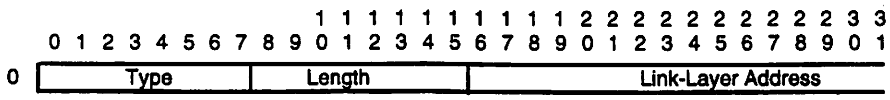
IP Destination Address source address from solicitation
all-nodes multicast if unspecified source address
all-nodes multicast if unsolicited

reserved - must be 0
Target Address - link-layer address for node
Destination Address- better 1st hop address to use

Legit Options Target link-layer address, Redirected Header

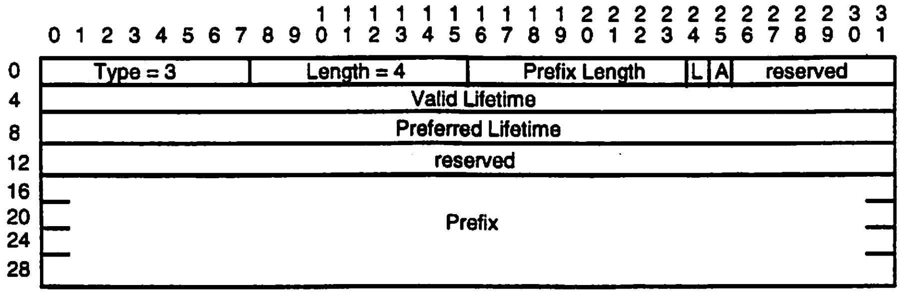
Link-layer Options

- ◆ Source Link-layer Address
- ◆ Target Link-layer Address



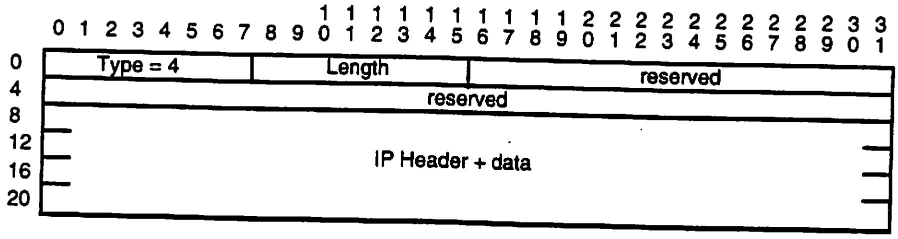
Type = 1	- Source Link-Layer Address
Type = 2	- Target Link-Layer Address
Length	- length of option in 8 byte units
Link-Layer Address	- media specific link layer address

Prefix Information Option



Prefix Length	- number of leading bits in the prefix that are valid
L	- if =1 prefix may be used to determine 'on-link' destinations
A	- if =1 prefix may be used to auto-configure node address
reserved	- must be 0
Valid Lifetime	- # seconds that prefix may be used of 'on-link' determination
Preferred Lifetime	- # seconds autoconfigured addresses are valid
Prefix	- address prefix, trailing bits must be zero

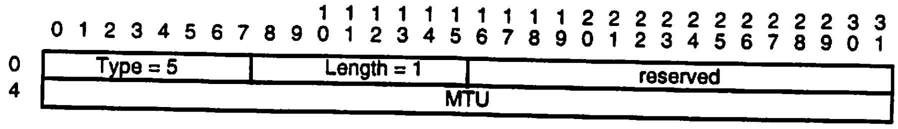
Redirected Header Option



0	Type = 4	Length	reserved
4	reserved		
8	IP Header + data		
12			
16			
20			

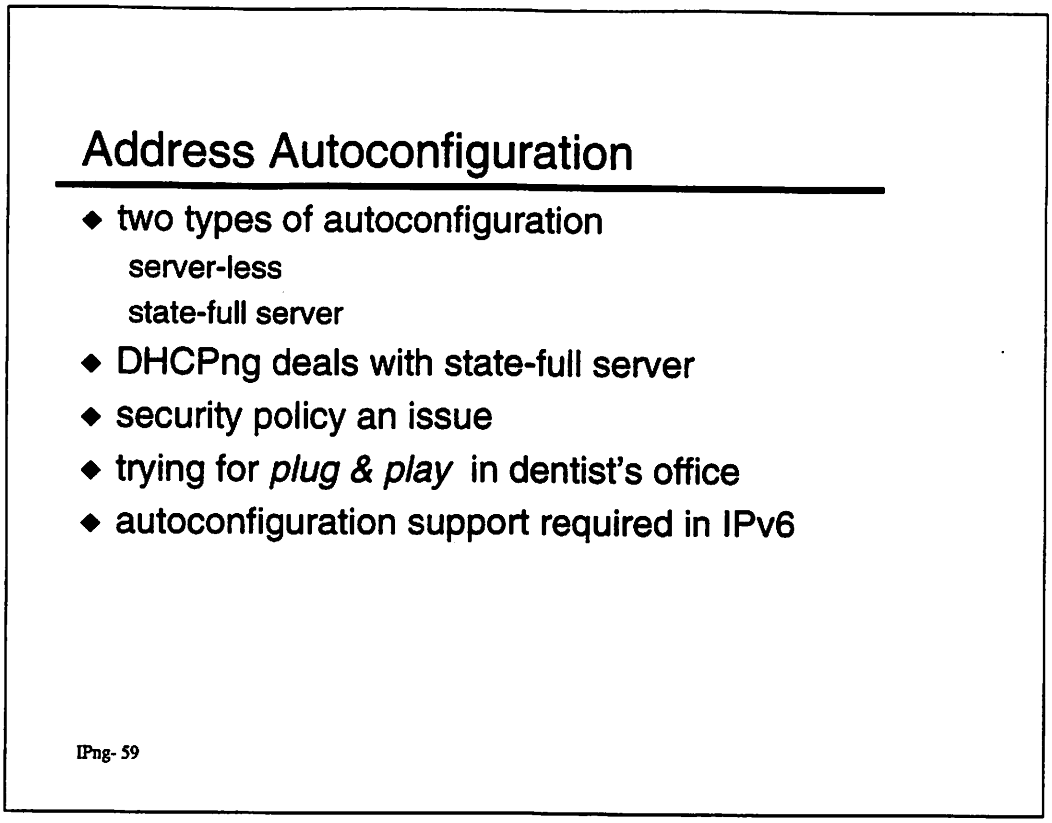
Length	- length of option in 8 byte units
reserved	- must be zero
IP Header + data	- header & data from packet being redirected (up to a total packet size of 576 bytes)

MTU Option



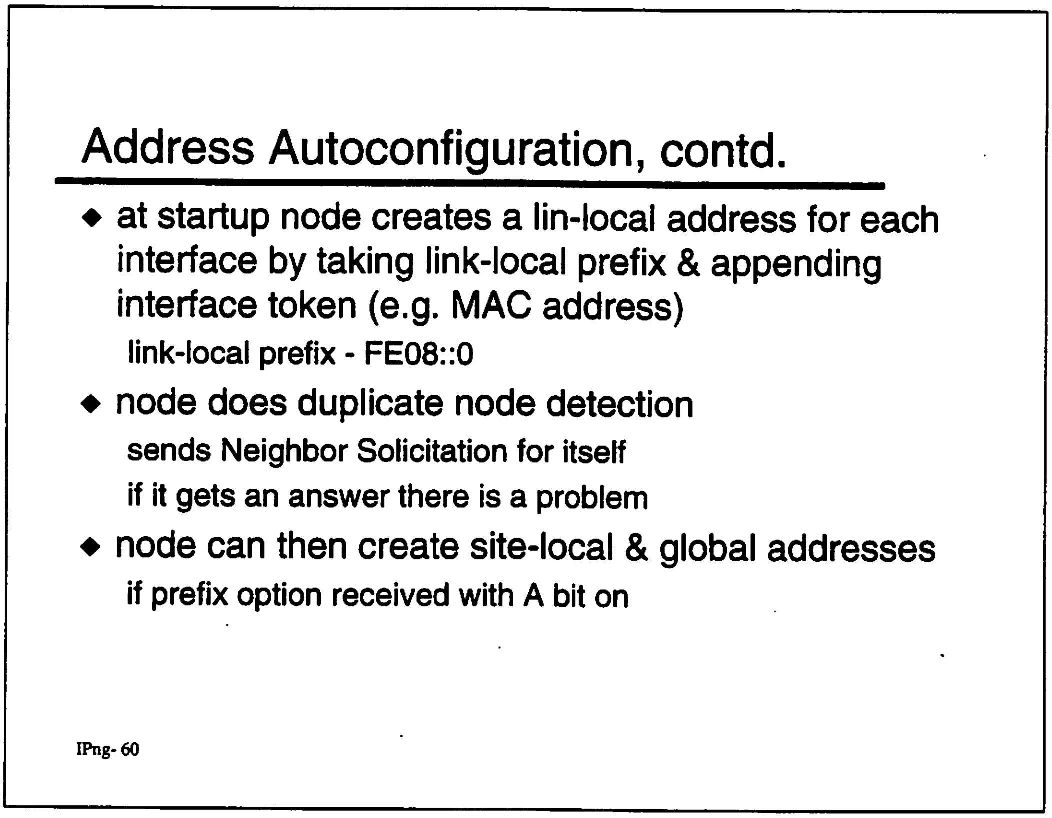
0	Type = 5	Length = 1	reserved
4	MTU		

MTU	- MTU to be used by nodes on this link



Address Autoconfiguration

- two types of autoconfiguration
  - server-less
  - state-full server
- DHCPng deals with state-full server
- security policy an issue
- trying for *plug & play* in dentist's office
- autoconfiguration support required in IPv6



Address Autoconfiguration, contd.

- at startup node creates a lin-local address for each interface by taking link-local prefix & appending interface token (e.g. MAC address)
  - link-local prefix - FE08::0
- node does duplicate node detection
  - sends Neighbor Solicitation for itself
  - if it gets an answer there is a problem
- node can then create site-local & global addresses
  - if prefix option received with A bit on

Address Lifetime

- address becomes “deprecated” when preferred lifetime expires
  - deprecated addresses should not be used to start new communications
  - deprecated addresses should be accepted on packets addressed to node
- address becomes “invalid” when invalid lifetime expires
  - invalid addresses must not be used as source addresses
  - packets destined to invalid addresses must be discared

Renumbering

- renumbering hosts
  - advertise prefix for provider A
  - connect to provider B
  - advertise prefix for provider A and provider B
    - use prefix B for new connections
  - advertise prefix for provider B only - do not use A

IPv6 ICMP

- error messages

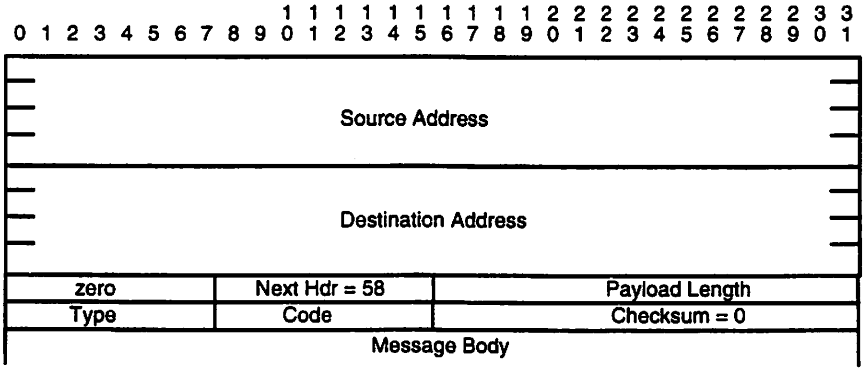
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem

- information messages

128	echo request
129	echo reply
130	group membership query
131	group membership report
132	group message termination

IPv6 ICMP pseudo-header

- ICMP checksum required
- pseudo-header fields



```
                    1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Source Address		
Destination Address		
zero	Next Hdr = 58	Payload Length
Type	Code	Checksum = 0
Message Body		

Also Payload Length from Hop-by-Hop Jumbo Payload option if present

ICMP Source Address

- if ICMP is in response to a packet sent to a node's unicast address
  - ICMP source address must be that unicast address
- if ICMP is in response to a packet sent to multicast address
  - ICMP source address must be unicast address of receiving interface
- if ICMP is in response to a packet not to the node
  - ICMP source address must be unicast address of interface on which forwarding failed
- else - look in routing table to get sending interface
  - ICMP source address must be unicast address of that interface

IPv6 ICMP must not be sent when getting

- an IPv6 error message
- a packet addressed to a IPv6 multicast address
  - except Packet to Big - to allow Path MTU to work with multicast
- a packet sent to link-level multicast address
- a packet sent to link-level broadcast address
- a packet whos source is not an IPv6 unicast address

- also nodes must limit rate of ICMP transmission

Destination Unreachable message

- codes
  - 0 no route to destination
  - 1 communication with destination administratively prohibited
    - e.g. blocked by a firewall
  - 2 not a neighbor (source route & strict bit set for hop)
  - 3 address unreachable
  - 4 port unreachable

Packet too Big message

- sent if packet larger than 576 bytes & too large for MTU of next hop link
- return MTU of next hop link

Time Exceeded message

- sent if
  - hop count on packet received by a router = 0 or is decrimented to 0
  - reassembly of fragmented packet timed out

UDP

- pseudo-header checksum now required
  - to provide check of proper deliverly
  - note - IPv4 header checksum was removed in IPv6

DNS

- new record type (AAAA) for IPv6 addresses
- reverse lookup via ip6.int

  IPv6 address encoded in reverse order by 4 bit nibbles

  e.g.

  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

API

- informational RFC of API for BSD-based systems

  IETF does not "do" APIs
- API provides for control of

  source & destination IPv6 addesses

  source route

  flow label

  hop count

  multicast group functions
- compatable with both IPv4 and IPv6

  provide support for IPv4 binaries

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

- dual stack

  new machines will support both IPv4 and IPv6
- 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

IPv6 Routing

- hierarchical addresses used in IPv6
- 1st version 'provider based' hierarchy
- working on geographic based
- address assignment a concern from the start
- easy renumbering may be important in maintaining efficient use of routing table space

Routing Paradigm

- longest-match routing will be used

- existing routing protocols will be modified for IPv6
  - RIPv2
  - OSPF
  - IS-IS
  - IDRP
- also source routing
  - provider selection
  - reduce per packet processing

From the IPng Recommendation

> "We feel that an improvement in the basic level of security in the Internet is vital to its continued success. Users must be able to assume that their exchanges are safe from tampering, diversion and exposure. Organizations that wish to use the Internet to conduct business must be able to have a high level of confidence in the identity of their correspondents and in the security of their communications. **The goal is to provide strong protection as a matter of course throughout the Internet.**"

IPng Security Recommendations

- support for the Authentication Header be required in all compliant IPv6 implementations
- support for a specific authentication algorithm be required
- support for the Privacy Header be required in all compliant IPv6 implementations
- support for a privacy authentication algorithm be required

IPv6 Security Design Goals

- provide authentication and integrity separate from encryption to minimize export control issues
- provide standard encryption mechanism
- ensure that encryption can deal with whole datagrams or portions
- keep mechanisms algorithm-independent
- decouple key management
- retain the ability to use firewalls
- same system for both IPv4 and IPv6

IPv6 Security Recommendation Issues

- performance impact
- key management
- what if kernel lies about having security
- "wrong layer" - should be in application
- export and use restrictions on encryption

Security Export Issue

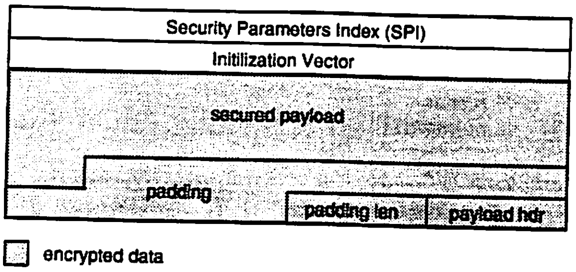
- strong encryption faces export controls in some countries
- strong encryption faces usage restrictions in some countries
- if IPv6 requires the implementation of strong encryption in order to be compliant to the standard
  - U.S. vendors have to ship non-compliant versions for export
  - all vendors must ship non-compliant versions for some countries
- some vendors do not like to be non-compliant
- big discussion during Danvers IETF meeting

Authentication Header

Next Header	Auth Data Len	reserved
Security Parameters Index (SPI)		
Authentication Data		

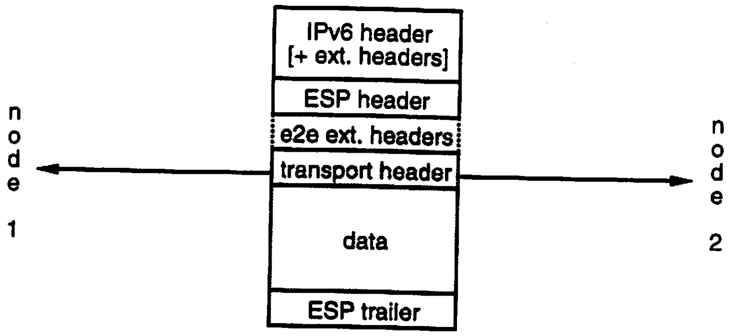
- Destination Address + SPI = security association
  - identifies algorithm, key etc
- used to authenticate all fields in packet that do not change en-route
- Keyed MD5 is the required default algorithm

Encapsulating Security Payload

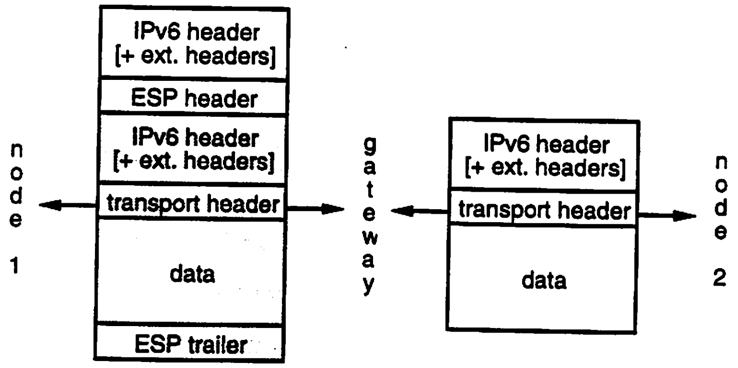


- DES-CBC is required default algorithm
- must be last non-encrypted header
- can encapsulate part or full packet

Transport Mode ESP (end-to-end)



Tunnel Mode ESP (end-to-gateway)


node 1

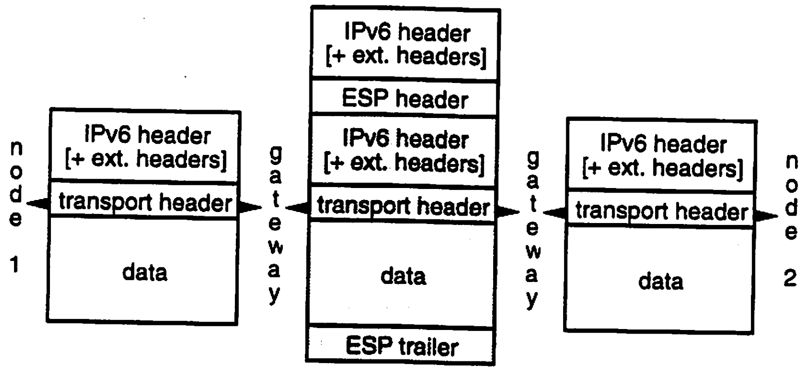
IPv6 header [+ ext. headers]
ESP header
IPv6 header [+ ext. headers]
transport header
data
ESP trailer

gateway

IPv6 header [+ ext. headers]
transport header
data

node 2

Tunnel Mode ESP (gateway-to-gateway


node 1

IPv6 header [+ ext. headers]
transport header
data

gateway

IPv6 header [+ ext. headers]
ESP header
IPv6 header [+ ext. headers]
transport header
data
ESP trailer

gateway

IPv6 header [+ ext. headers]
transport header
data

node 2

Security RFCs

- RFC1825 - Security Architecture for the Internet Protocol
- RFC1826 - IP Authentication Header
- RFC1827 - IP Encapsulating Security Payload (ESP)
- RFC1828 - IP Authentication using Keyed MD5
- RFC1829 - The ESP DES-CBC Transform

- work with IPv4 & IPv6

Issues - Addressing Architecture

- routing table efficiency requires that addressing hierarchy follow network hierarchy
- otherwise tend toward flat routing
  - require entry for all non hierarchical routes
- addressing that follows network hierarchy is provider based addressing
  - tends to bind customers to vendors
- geographic addressing requires richly connected network
  - must get vendor cooperation at many places

Addressing Issues, contd.

- in future may have charges for addresses that can not be aggregated
- IPv6 auto configuration may help
  - make renumbering easier
  - support for multi-homed domains
    - multiple addresses per interface
  - can migrate from one provider to another

Status

- base documents now at Proposed Standard level
- multiple interoperable implementations
  - hosts & routers
- interoperability test at UNH in Feb 1996

Effects

- confusion
  - when should IPv6 be used?
  - conformance issues
  - new technology
- feature support
  - some features must wait native network support
    - flow ID, ERP routing
  - some features can be supported in tunnels
    - authentication, encryption, NSAP & IPX addresses

Effects, contd.

- migration from OSI CLNP
  - larger addresses supported
  - can map U.S. gosip NSAPs
  - support for OSI apps over IPv6
- set top box autoconfiguration
  - will help to get router autoconfiguration

Prospects

- carrots may be hard to see
  - plug & play, security, flow ID, advanced routing
- NAT box option
  - network address translation box
  - use private IPv4 address and map to subset for external
- IPv6 too little, too early?
- a few key decisions will determine IPv6 fate
- high level of assumption that IPv6 is the right thing
- disagreement over size of step taken

IPng Summary

- evolutionary step from IPv4
- simple transition and coexistence
- extensible
- secure
- QoS support hooks
- future use depends on many factors
  - security requirements
  - vendor support
  - Internet/cable TV/NII/GII confusion