IPv6 overview and status

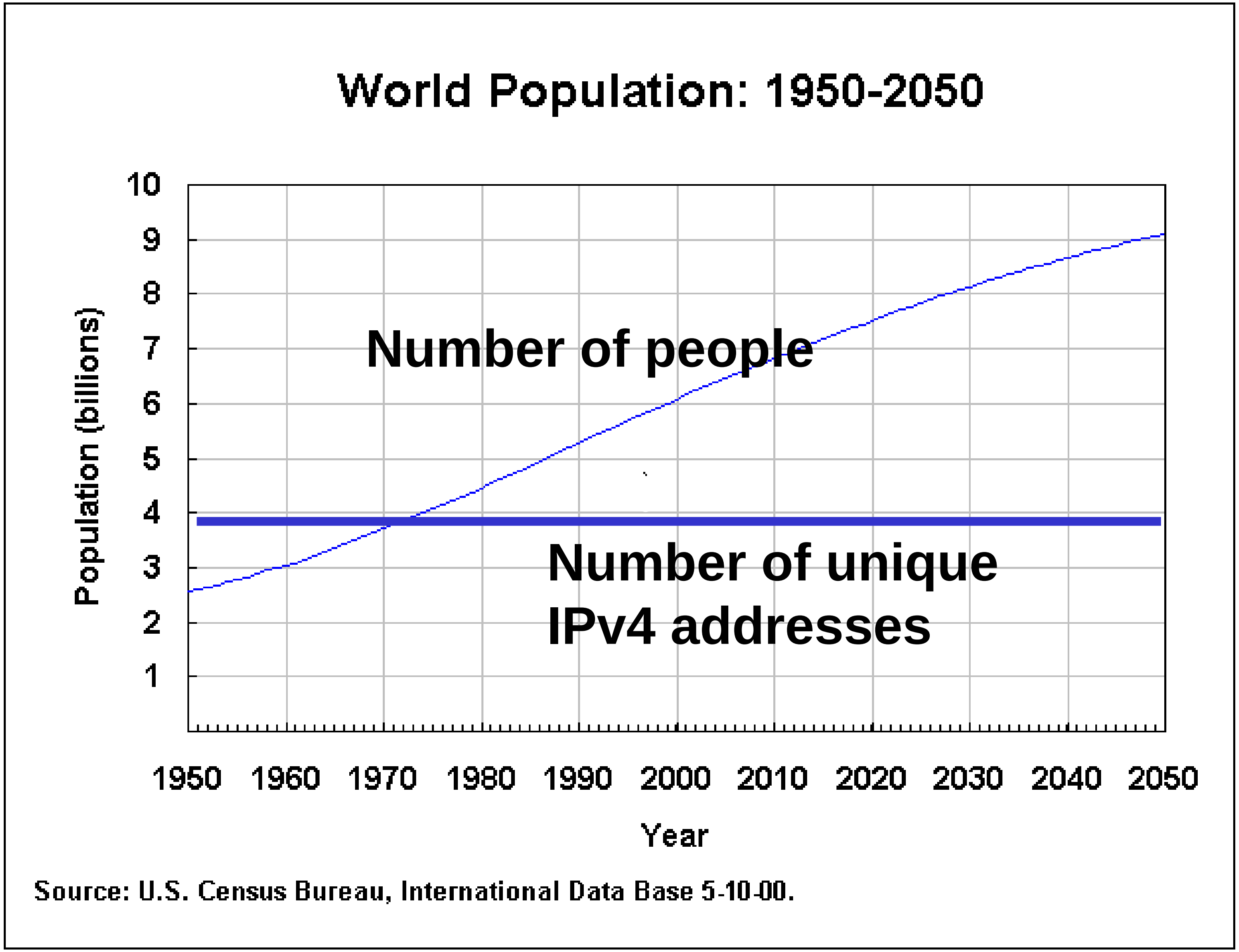
Brian E Carpenter

June 2007

Topics

- IPv6 Basics
- Practical aspects of enabling middleware for IPv6
- Recent developments

Why we need IPv6



Living with too few addresses

- If we don't have many more addresses than we expect to have devices, we will have a fractured network with artificial internal boundaries.
  - The tense is wrong. Today in the US, there is widespread use of ambiguous (net 10) address space with consequent glitches and hacks.
  - Much more acute problem in (e.g.) China.
- This is a major operational cost and an obstacle to innovative applications.
  - In fact, that is exactly why Cerf and Kahn invented IP, but they didn't go far enough. It's time to fix that bug.

The Challenge from IPv4: Network Address Translation

- Ambiguous addresses: a “quick and nasty” solution
- Falsely marketed as “private” addresses
- NAT breaks many non-client-server applications as well as hindering network level security
  - Especially annoying for multi-party communications such as Grid virtual organisations
  - Causes great operational complexity and cost
  - Requires troublesome interworking units (NAT + ALG + proxy)
- NAT is the major barrier to innovative applications on the Internet today
  - In the best case, we end up with messy work-arounds

The Internet as a platform for innovation must scale up

- A reasonable goal is 10 billion Internet nodes
  - One node per human in 2050
  - 10 billion nodes squeezed into 4 billion IPv4 addresses –why would we do that?
- Immediate benefit for applications actively hurt by NAT today
  - release the known potential
- Strategic benefit for the next 50 years at least
  - avoid the opportunity cost of staying with IPv4

Scaling up the Internet

- IPv6 represents a major step in the Internet’s ability to scale, like the introduction of IPv4 25 years ago.
  - The only way out is bigger addresses.
  - The IETF picked 128 bits.

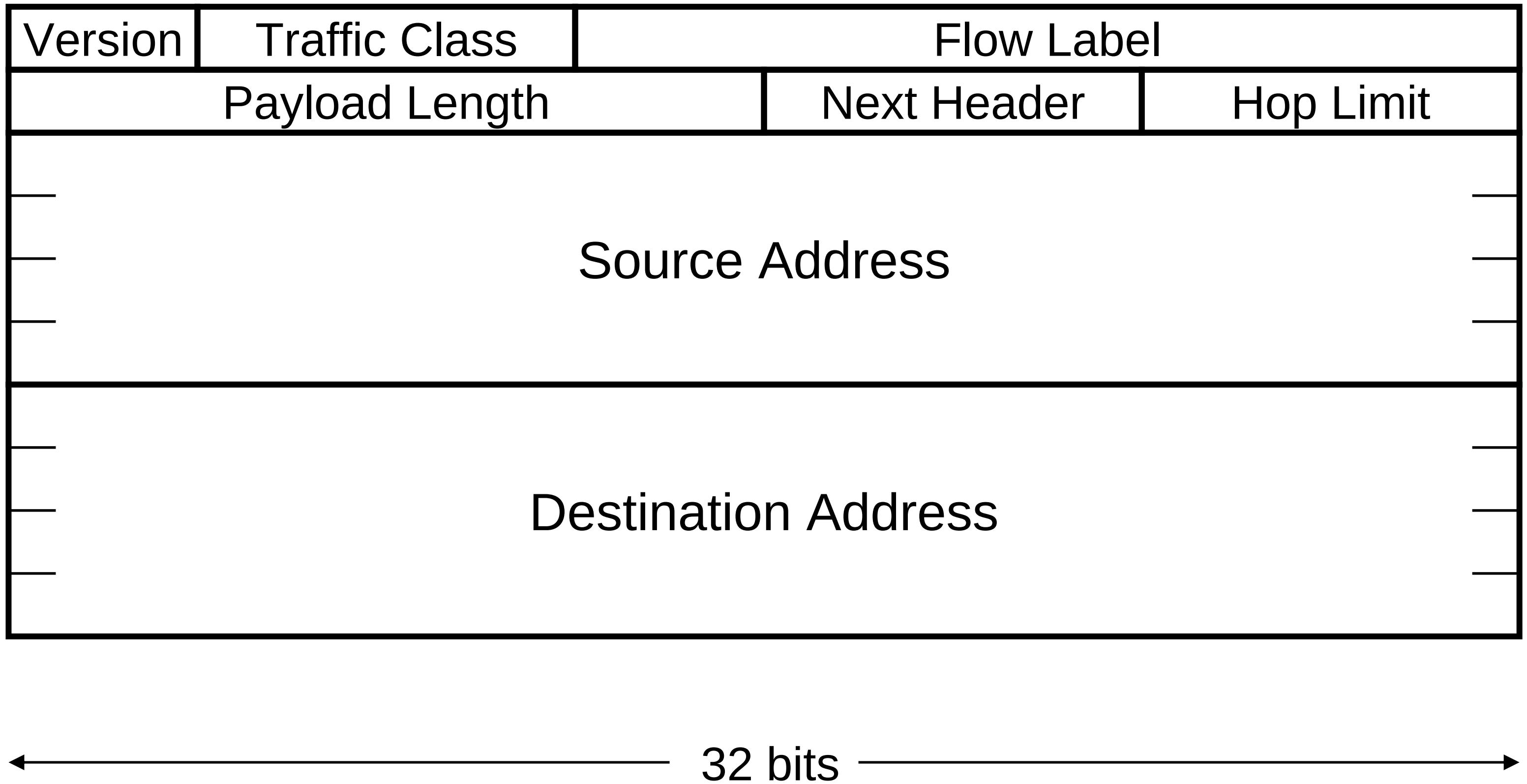
Other major benefits of IPv6

- Automatic configuration
  - stateless, for manager-free networks
  - stateful (DHCPv6), for managed networks
  - help for site renumbering
- Potential for better aggregated routing than IPv4
- Complete Mobile IP solution
- Global addressability allows IPSEC end to end.
  - mechanisms for secure firewall traversal will come
- Simplified header format with clean extensibility.
  - allows effective header compression

Untapped potential of IPv6

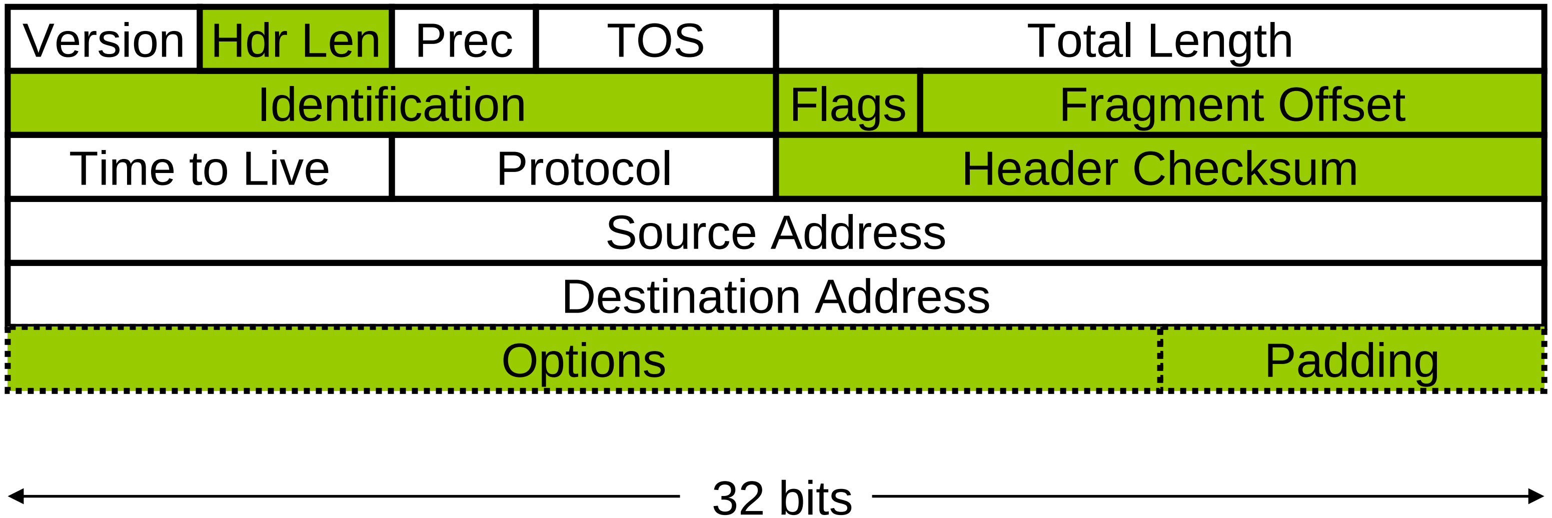
- Provision for a QOS flow label.
- Ability to manufacture addresses by the thousands creates scope for innovative virtualization techniques.

The IPv6 Packet Header



credit: Steve Deering

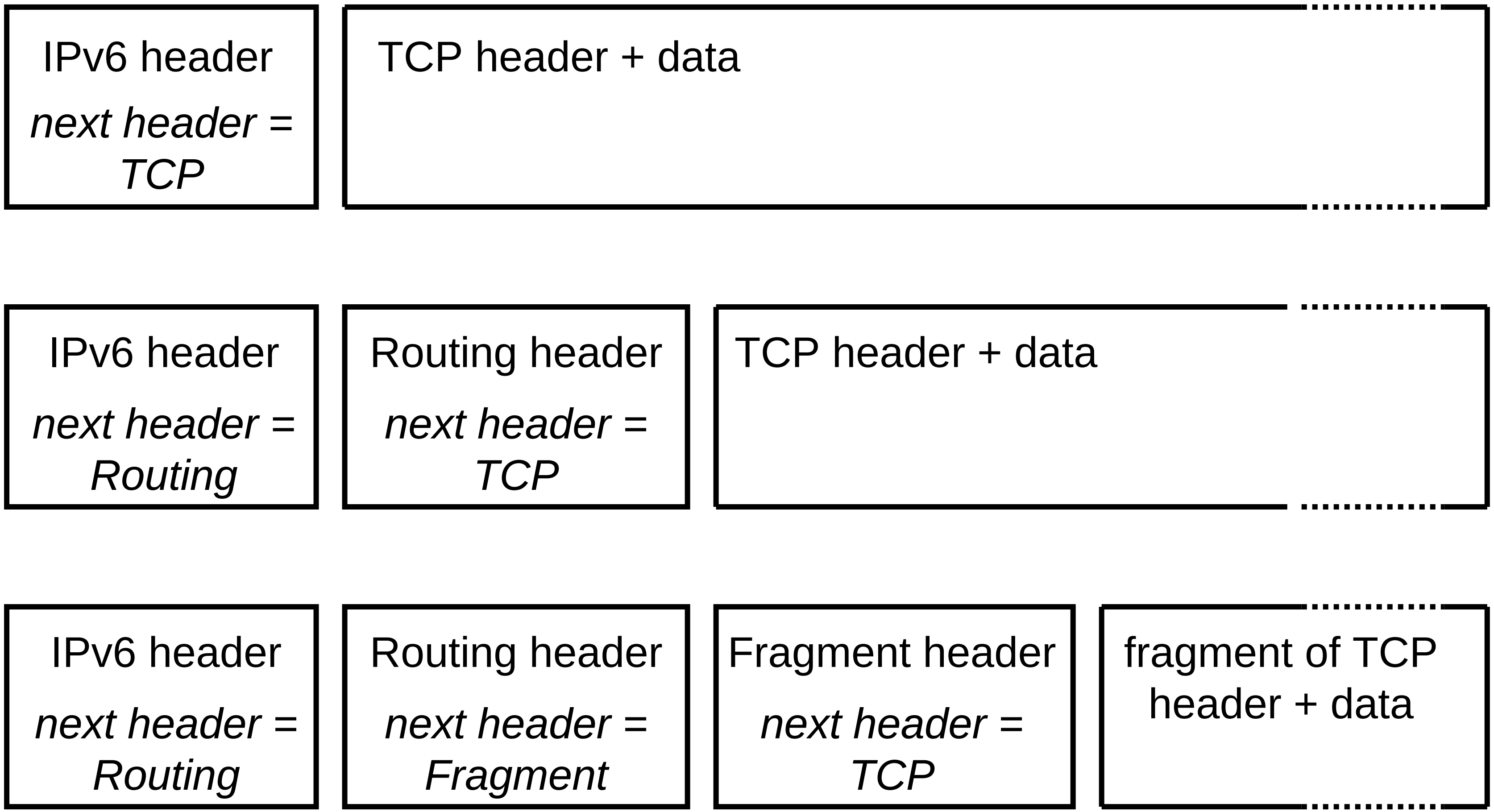
The IPv4 Header



Shaded fields are absent from IPv6 header

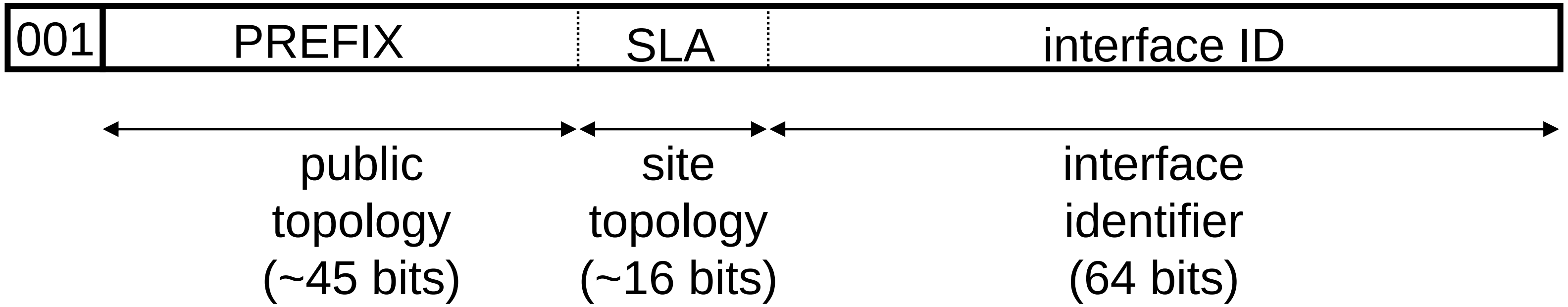
credit: Steve Deering

Extension Headers



credit: Steve Deering

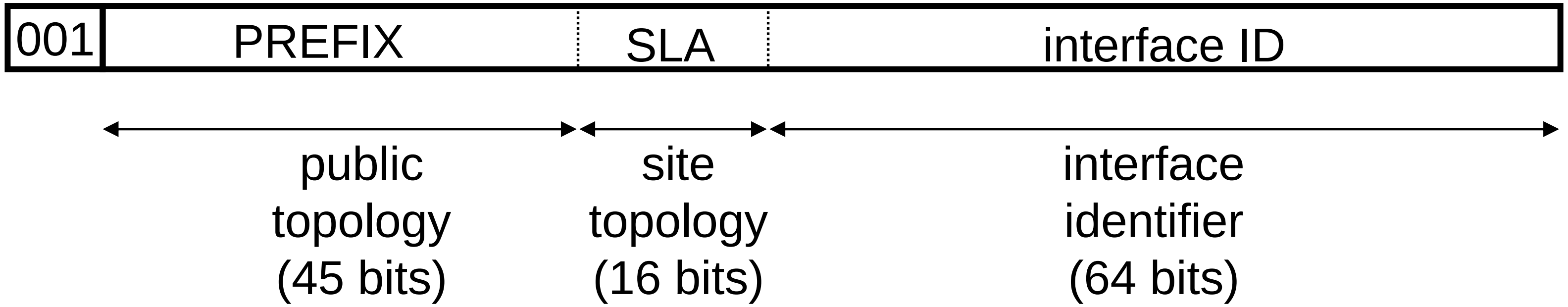
Global Unicast Addresses



- Prefix ranges may be assigned to providers or exchanges
- Currently recommended that all sites including homes get 48 bit prefixes (35,184,372,088,832 are available)
  - Or longer prefixes to be very conservative
- SLA = Site-Level Aggregator (subnet prefix)
- Subfields variable-length, non-self-encoding (cf CIDR) → much better route aggregation than legacy IPv4

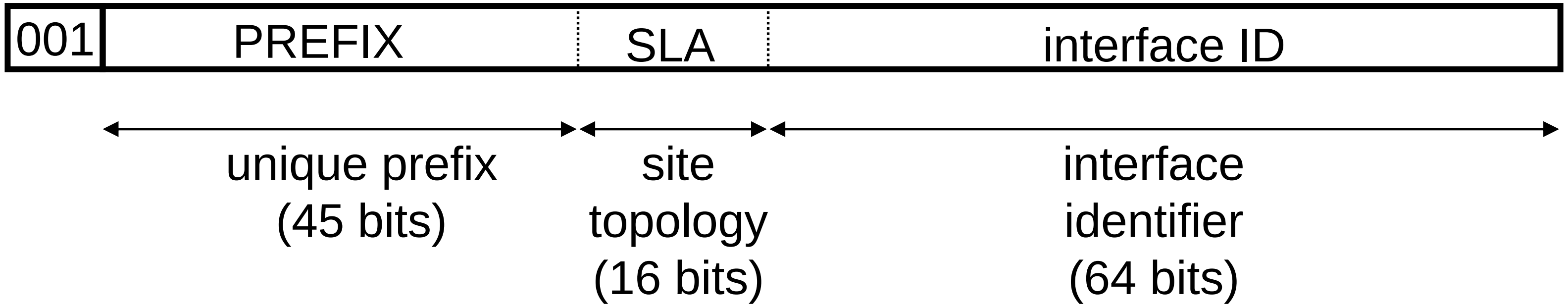
`credit: Steve Deering`

Privacy Addresses



- Identical, except that the interface ID is a pseudo-random number assigned for a fixed time period
- Prevents long-term tracking of devices and users

Unique Local Unicast Addresses



- Identical, except that the prefix includes a 40 bit unique pseudo-random value
- Allows unambiguous private addressing within a site, and cannot be accessed from outside the site

Stateless auto-configuration

- Intended for "dentist's office" scenario (i.e. no manual configuration needed)
  - in this respect, a dentist's office is not so different from a tactical military deployment
- Nodes start by acquiring a unique* Link Local address
- Routers issue Router Advertisements to proffer connectivity and prefix information to new nodes
- Nodes then use Neighbor Discovery and Duplicate Address Detection procedures to acquire a unique* routeable address & find neighbors

* Unique ID can be generated from hardware MAC address

DHCPv6

- Similar to DHCP for IPv4, but not identical.
- Intended for managed networks

Mobile IPv6 (1)

- Complex topic, deserves its own talk
- Routing protocol for mobile IPv6 hosts
  - Transparent to upper layer protocols and applications
  - Tries to avoid actively involving routers
  - Protocol state held in end stations (*mobile nodes & correspondent nodes*)
  - One exception: the *Home Agent* (i.e. the mobile node must have a home site to support it)

Mobile IPv6 (2)

- When away from home, mobiles
  - Acquire care-of address from host network
  - Register care-of address with home agent and any relevant correspondent nodes
- IPSec protects signalling between mobile node and its home agent
  - This doesn't protect conversation, which needs separate security (e.g. IPsec or TLS)

The Flow Label

- A 20 bit field in every packet
  - zero by default (no special treatment)
  - non-zero values allow routers to efficiently recognize flows of related traffic for QOS, load balancing, etc.
  - flows can be fine-grained (one phone call) or coarse-grained (all email traffic) according to usage model
  - new technology, not yet widely supported, but great potential

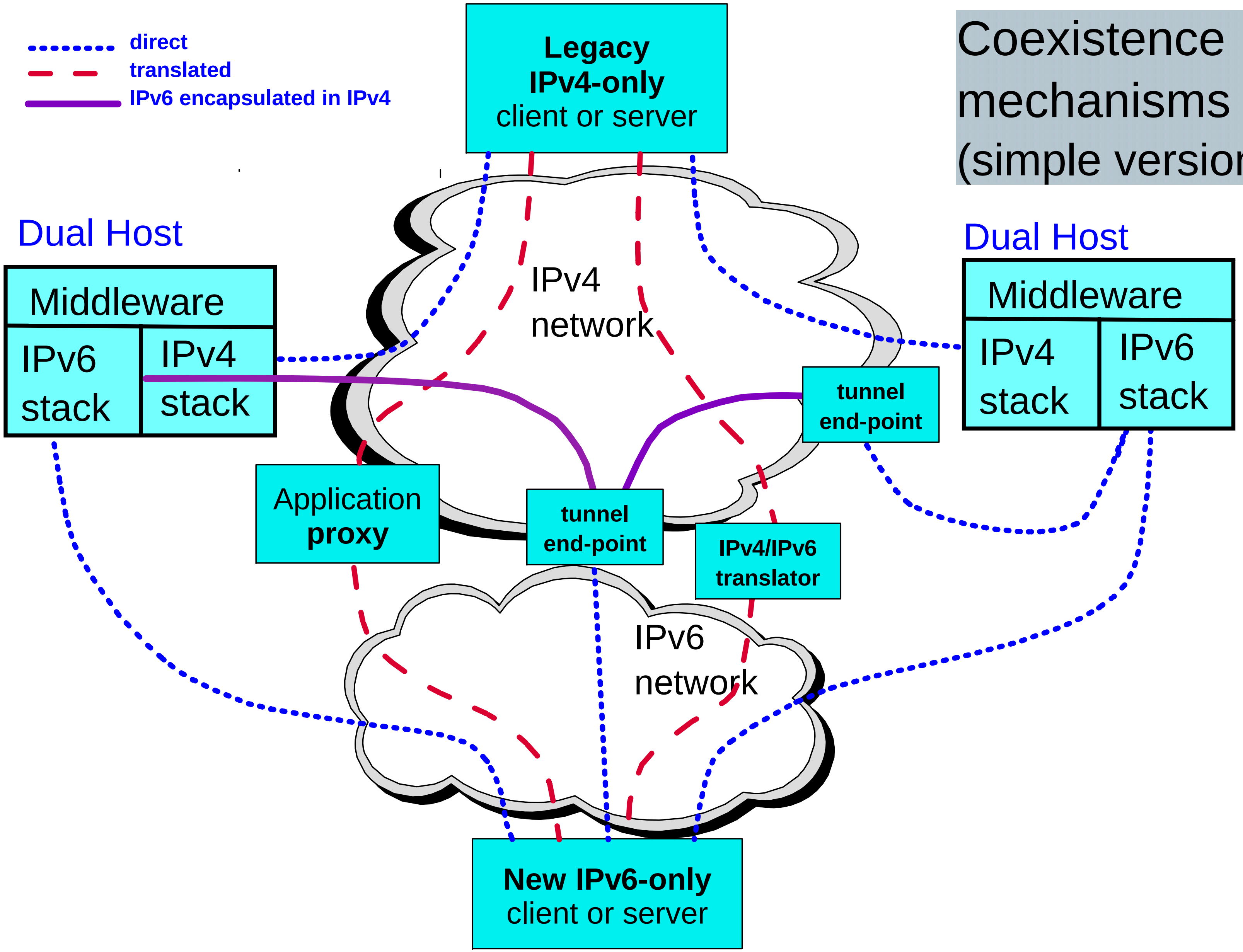
IPsec end to end

- Authentication and confidentiality for all traffic can only be guaranteed by protecting the IP layer itself
  - TLS/SSL only protects TCP traffic
- IPsec for IPv4 has severe challenges in dealing efficiently with NAT
  - workarounds are possible but inefficient
- By using global IPv6 addressing with no NAT, IPsec can be used as designed, to protect traffic all along the route

Local Network Protection

- By combining features of IPv6, such as using both globally routeable addresses and unique local addresses appropriately, a network domain can be effectively protected against many forms of attack at least as well as by using IPv4 NAT, but without the operational disadvantages of NAT.
- RFC 4864
- Ask your router and firewall vendor when they will support such protection.

Coexistence mechanisms (simple version)



Coexistence Mechanisms (1)

- Dual stack (RFC 2893)
  - Socket API (RFC 3493)
  - DNS supports IPv4 and IPv6 (RFC 1886)
- IPv6 in IPv4 tunnels (RFC 2893)
- NAT-PT translation (RFC 2766)
  - IETF likely to deprecate this
- Tunnel Broker (RFC 3053)
- 6to4 implicit tunnels (RFC 3056)

Coexistence Mechanisms (2)

- Less favored in IETF
  - Bump in the Stack (RFC 2767)
  - Bump in the API (RFC 3338)
  - SOCKS (RFC 3089)
  - Transport relay (RFC 3142)
  - 6over4 using IPv4 multicast (RFC 2529)
  - ISATAP (RFC 4214)
  - Teredo (RFC 4380)

- Still in draft (expired)
  - DSTM

Summing up coexistence

- We have 12 documented coexistence mechanisms
  - Certainly enough to deal with most situations
- The IETF has made an operational analysis of various scenarios.
- Experience matching solutions to scenarios will show whether even more design work is needed.
  - In all probability, mainly tuning and profiling of standards is all that is needed

A few words about DNS

- Dual-stack DNS needs careful thought.
- Need to resolve IPv6 queries over IPv4, and vice versa.
- If a host has an IPv4 address and a few IPv6 addresses, a DNS query should return several answers.
- Which one should we try?
- Getting this right remains tricky

Standards status

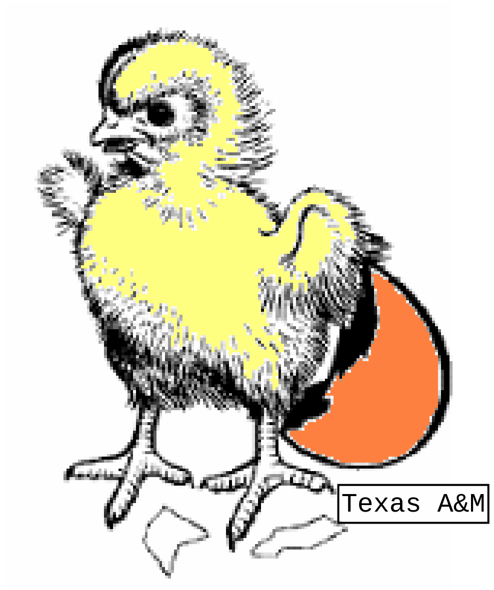
- Basic standards for the protocol, auto-configuration, DHCP, mobility, socket API, DNS, and coexistence mechanisms are done.
- IETF work continues on
  - site multihoming
  - deployment scenarios
  - endless refinements & interactions with other protocols
- IPv6 is assumed by IMS standards for next generation telco networks.
  - telcos understand scaling and long term planning

Implementation status

- All significant operating systems and router vendors now support dual IPv4/IPv6 stacks and socket APIs
- BIND DNS, PowerDNS, etc. support IPv6
- Linux Mobile IPv6 support from IBM LTC
- Java 1.4 and later supports IPv6
- Many public domain applications support IPv6
- The conversion of commercial applications is progressing
- Vista and Longhorn prefer IPv6 to IPv4

Deployment status (1)

- Multiple R&D IPv6 testbeds running around the world
- Numerous commercial IPv6 services on offer, but we have a classical chicken/egg deadlock
  - when will enterprises see the business case?
- Numerous IPv6 Task Forces worldwide.
- Emerging requirement in RFPs
  - Required by ITU NGN
  - US DoD requirement since 10/03
  - USG mandate for 2008.



Deployment status (2)

- About 860 IPv6 prefixes announced in BGP, which mainly belong to ISPs.
  - Hard to know how many offer commercial IPv6 (certainly at least 25, of which ~10 in Japan)
  - Remember that customer prefixes are mainly aggregated behind ISP prefixes: a small number is good news!
  - The pre-production 6BONE officially switched off 6/6/06
  - Major commitments such as CERNET (China) and US DoD and OMB (required operational in June 2008)
  - Connectivity is real, e.g., see
    http://net-stats.ipv6.tilab.com/bgp/
    http://bgp.potaroo.net/index-bgp.html

Topics

- IPv6 Basics
- <u>Practical aspects of enabling middleware for IPv6</u>
- Recent developments

What changes in middleware or applications?

- Principally, the Socket API changes
  - New API supports both IPv4 and IPv6. Code can be version-independent
  - No change to fundamental model
  - No conceptual change to DNS or TCP
  - But an IP address no longer fits into an integer
- Applications do not need to be restructured
  - But all socket API calls, all usage of IP addresses, and URL parsing, must be updated

IPv6 in DNS

- A new record type AAAA is supported
  – Like A record but returns IPv6 address
- Hosts may have both A and AAAA records
  – No relationship between the two
- DNS protocol works identically over IPv4 or IPv6
  – Can send DNS queries for AAAA over IPv4, and for A over IPv6
- Reverse lookups in *ip6.arpa* (changed from *ip6.int*)

Aspects of new C socket API

- Store addresses in *addrinfo* instead of *in_addr*
- Use *localhost* systematically for loopback
- *getnameinfo()* and *getaddrinfo()* replace *gethostbyname()* and *gethostbyaddr()*
  - When *getnameinfo* returns multiple addresses, try to connect first with IPv6, then with IPv4

Java JDK

- Java 1.4 and above "just works" with IPv6 (minor bug fix in JDK 1.5)
  - for full flexibility, code in C!
- JDK includes network preferences for IPv6 (i.e. *java.net.preferIPv4Stack, java.net.preferIPv6Addresses*)

Text representation

- IPv4 9.1.2.3 (dotted decimal)
- IPv6 2002:808d:3871::808d:3871 (colon hexadecimal, :: elides zeros)
- URLs
  - http://9.1.2.3:80/index.html
  - http://[2002:808d:3871::808d:3871]:80/index.html

GUIs and parsers

- Any GUI that displays or accepts IP addresses must support both text formats
- Any URL parser must support literal IPv6 addresses

General assessment

- Thus far I have heard **NO** application or middleware developer report special difficulty in upgrading to support IPv6 as well as IPv4. "It's just work."
- IBM SWG is tackling this, largely in response to the DoD requirements - but it takes time, as every component has to be checked.

Topics

- IPv6 Basics
- Practical aspects of enabling middleware for IPv6
- <u>Recent developments</u>

IPv6 WG in the last 2 years: mainly consolidation

- TCP MIB [update] (RFC 4022)
- IP Tunnel MIB [update] (RFC 4087)
- IPv6 Scoped Address Architecture (RFC 4007)
- Unique Local IPv6 Unicast Addresses (RFC 4193)
- Default Router Preferences (RFC 4191)
- Host-to-Router Load Sharing (RFC 4311)
- IPv6 Addressing Architecture [update] (RFC 4291)
- ICMPv6 [update] (RFC 4443)
- IPv6 Node Requirements (RFC 4294)
- IP MIB [update] (RFC 4293)
- IP Forwarding Table MIB [update] (RFC 4292)
- Neighbor Discovery Proxies (RFC 4389)
- Link-Scoped IPv6 Multicast Addresses [update] (RFC 4489)
- IPv6 Node Information Queries (RFC 4620)

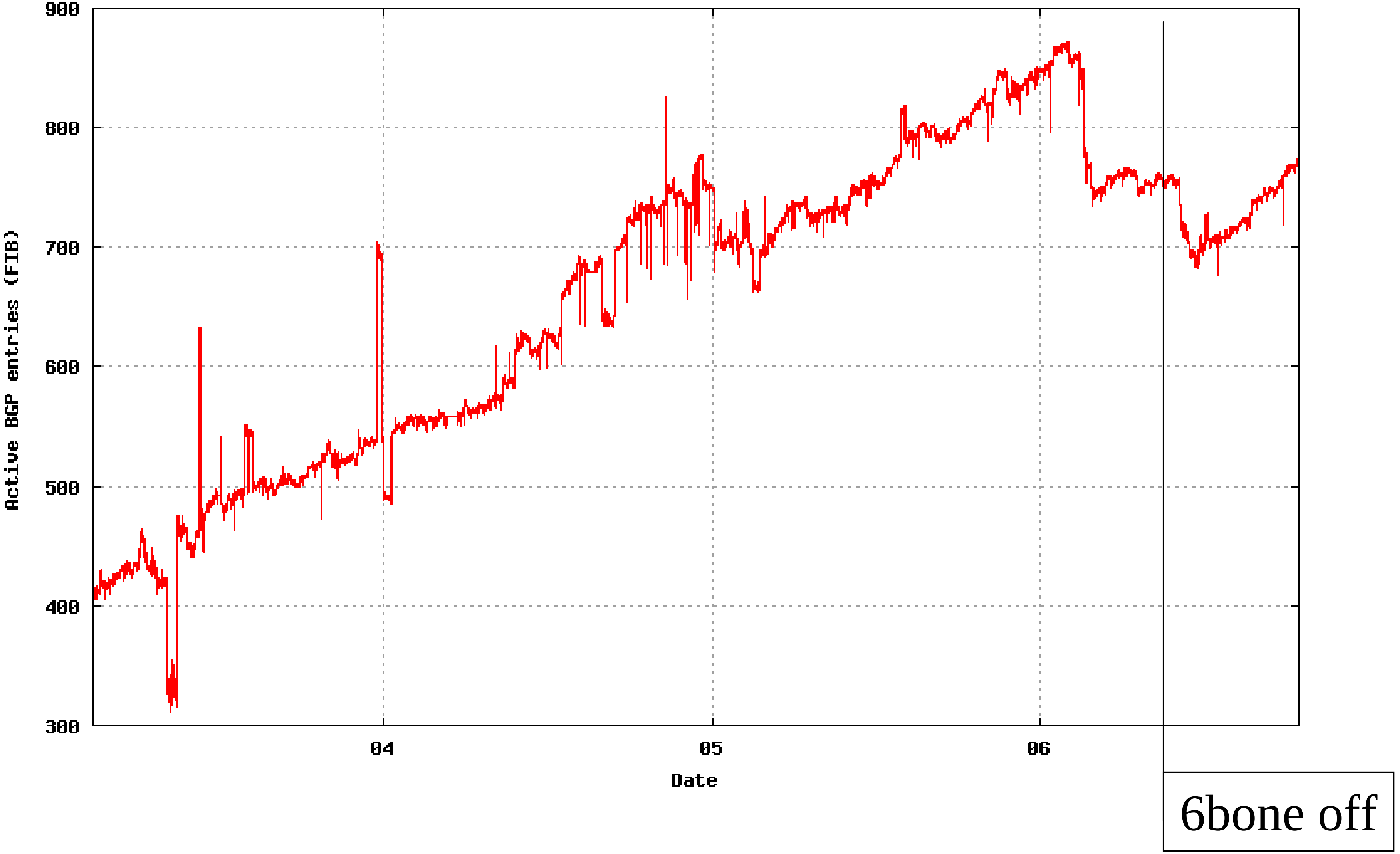
V6OPS WG in the last 2 years: mainly deployment issues

- Security Considerations for 6to4 (RFC 3964)
- Application Aspects of IPv6 Transition (RFC 4038)
- Introducing IPv6 into ISP Networks (RFC 4029)
- IPv6 Enterprise Network Scenarios (RFC 4057)
- Renumbering an IPv6 Network (RFC 4192)
- IPv6 Transition in 3GPP Networks (RFC 4215)
- Basic Transition Mechanisms for IPv6 [update] (RFC 4213)
- VLANs for IPv4-IPv6 Coexistence in Enterprise Networks (RFC 4554)
- ISP IPv6 Deployment Scenarios in Broadband Access Networks (RFC 4779)
- IPv6 Enterprise Network Analysis - IP Layer 3 Focus (RFC 4852)
- Local Network Protection (RFC 4864)
- Recommendations for Filtering ICMPv6 Messages in Firewalls (RFC 4890)
- Using IPsec to Secure IPv6-in-IPv4 Tunnels (RFC 4891)

IPv6 multihoming in the last 2 years

- MULTI6 WG
  - IPv4 Multihoming Practices and Limitations (RFC 4116)
  - Architectural Approaches to Multi-Homing for IPv6 (RFC 4177)
  - Threats relating to IPv6 Multihoming Solutions (RFC 4218)
  - Things MULTI6 Developers Should Think About (RFC 4219)
- SHIM6 WG
  - Working on shim in host IPv6 stack to conceal multihoming events (changes of address) from transport layer
  - No RFCs so far
  - Controversial approach among ISPs

IPv6 routing history



Other IPv6 WGs in progress

- 6lowpan: IPv6 over Low power WPAN
- mip6: Mobility for IPv6
- monami6: Mobile Nodes and Multiple Interfaces in IPv6
- softwire: Softwires. IPv4 over an IPv6 core.
- *plus increasing attention to IPv6 in all other current protocol designs*

What's left to do?

- A big problem known since about 1992 remains open - how to make Internet-wide area routing scale adequately for a ten billion node network?
  - serious concern that BGP4 (the current inter-ISP routing protocol) will run out of steam within 5 years
  - IPv6 does nothing to fix this
- IPv6 is not the end of the story
  - Expect more change in the future
  - But based on IPv6, not IPv4

Pointers

- IETF WGs

www.ietf.org/html.charters/

ipv6-charter.html

v6ops-charter.html

shim6-charter.html

➢ (drafts and RFCs are linked from these sites)

- IPv6 Forum

www.ipv6forum.org