IPv6 Basics

Jordi Palet (jordi.palet@consulintel.es)

Why a New IP?

Only *compelling* reason: more addresses!

- for billions of new devices,
  e.g., cell phones, PDAs, appliances, cars, etc.
- for billions of new users,
  e.g., in China, India, etc.
- for "always-on" access technologies,
  e.g., xDSL, cable, ethernet-to-the-home, etc.

But Isn't There Still Lots of IPv4 Address Space Left?

- ~ Half the IPv4 space is unallocated
  - if size of Internet is doubling each year, does this mean only one year's worth?!
- No, because today we deny unique IPv4 addresses to most new hosts
  - we make them use methods like NAT, PPP, etc. to share addresses
- But new types of applications and new types of access need unique addresses!

Why Are NAT's Not Adequate?

- They won't work for large numbers of "servers", i.e., devices that are "called" by others (e.g., IP phones)
- They inhibit deployment of new applications and services
- They compromise the performance, robustness, security, and manageability of the Internet

Incidental Benefits of Bigger Addresses

- Easy address auto-configuration
- Easier address management/delegation
- Room for more levels of hierarchy, for route aggregation
- Ability to do end-to-end IPsec (because NATs not needed)

Incidental Benefits of New Deployment

- Chance to eliminate some complexity, e.g., in IP header
- Chance to upgrade functionality, e.g., multicast, QoS, mobility
- Chance to include new enabling features, e.g., binding updates

Summary of Main IPv6 Benefits

- Expanded addressing capabilities
- Server-less autoconfiguration (“plug-n-play”) and reconfiguration
- More efficient and robust mobility mechanisms
- Built-in, strong IP-layer encryption and authentication
- Streamlined header format and flow identification
- Improved support for options / extensions

Why Was 128 Bits Chosen as the IPv6 Address Size?

- Some wanted fixed-length, 64-bit addresses
  - easily good for 10^{12} sites, 10^{15} nodes, at .0001 allocation efficiency (3 orders of mag. more than IPng requirement)
  - minimizes growth of per-packet header overhead
  - efficient for software processing
- Some wanted variable-length, up to 160 bits
  - compatible with OSI NSAP addressing plans
  - big enough for autoconfiguration using IEEE 802 addresses
  - could start with addresses shorter than 64 bits & grow later
- Settled on fixed-length, 128-bit addresses
  - (340,282,366,920,938,463,463,374,607,431,768,211,456 in all!)

What Ever Happened to IPv5?

0–3		unassigned
4	IPv4	(today's widespread version of IP)
5	ST	(Stream Protocol, not a new IP)
6	IPv6	(formerly SIP, SIPP)
7	CATNIP	(formerly IPv7, TP/IX; deprecated)
8	PIP	(deprecated)
9	TUBA	(deprecated)
10-15		unassigned

IPv6 Tutorial

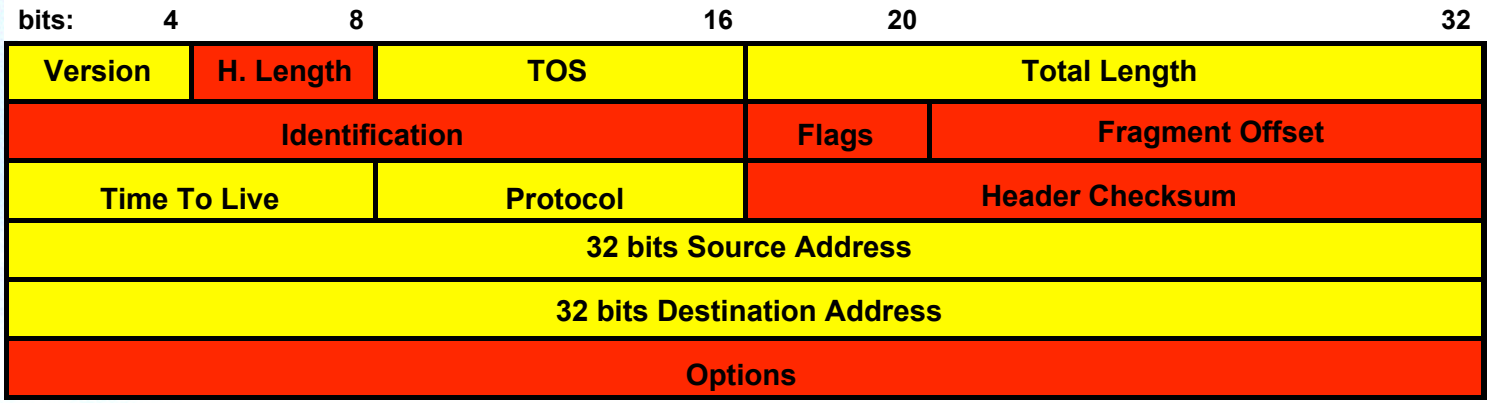
Header Formats

RFC2460

- Internet Protocol, Version 6: Specification
- Changes from IPv4 to IPv6:
  - Expanded Addressing Capabilities
  - Header Format Simplification
  - Improved Support for Extensions and Options
  - Flow Labeling Capability
  - Authentication and Privacy Capabilities

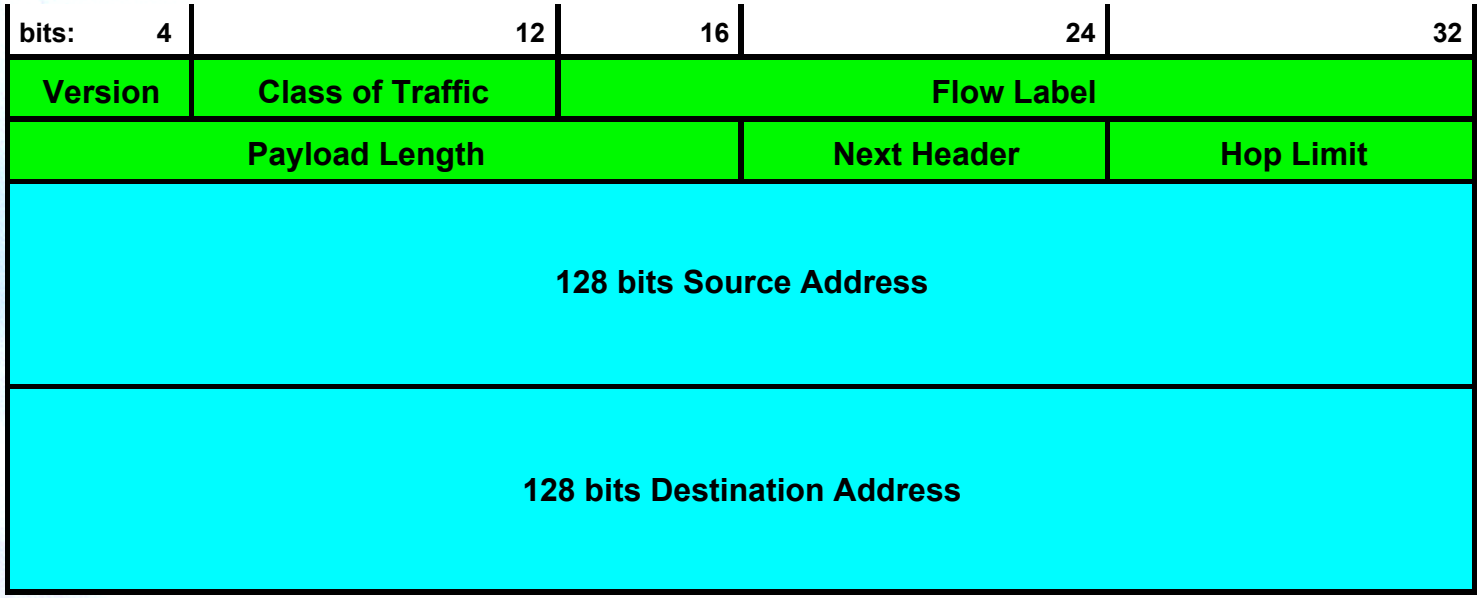
IPv4 Header Format

- 20 Bytes + Options



IPv6 Header Format

- From 12 to 8 Fields (40 bytes)



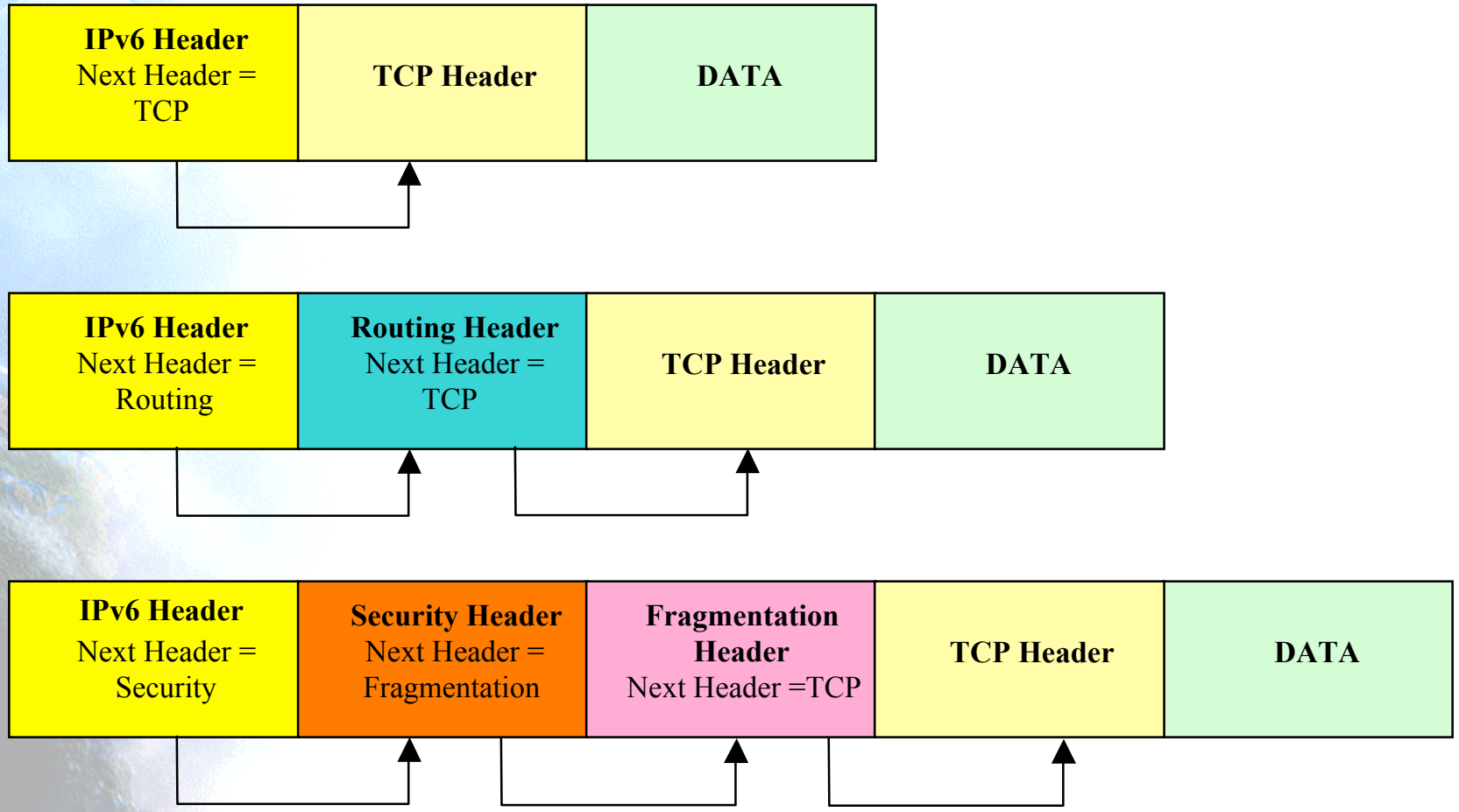
  - Avoid checksum redundancy
  - Fragmentation end to end

Summary of Header Changes

- 40 bytes
- Address increased from 32 to 128 bits
- Fragmentation and options fields removed from base header
- Header checksum removed
- Header length is only payload (because fixed length header)
- New Flow Label field
- TOS -> Traffic Class
- Protocol -> Next Header (extension headers)
- Time To Live -> Hop Limit
- Alignment changed to 64 bits

Extension Headers

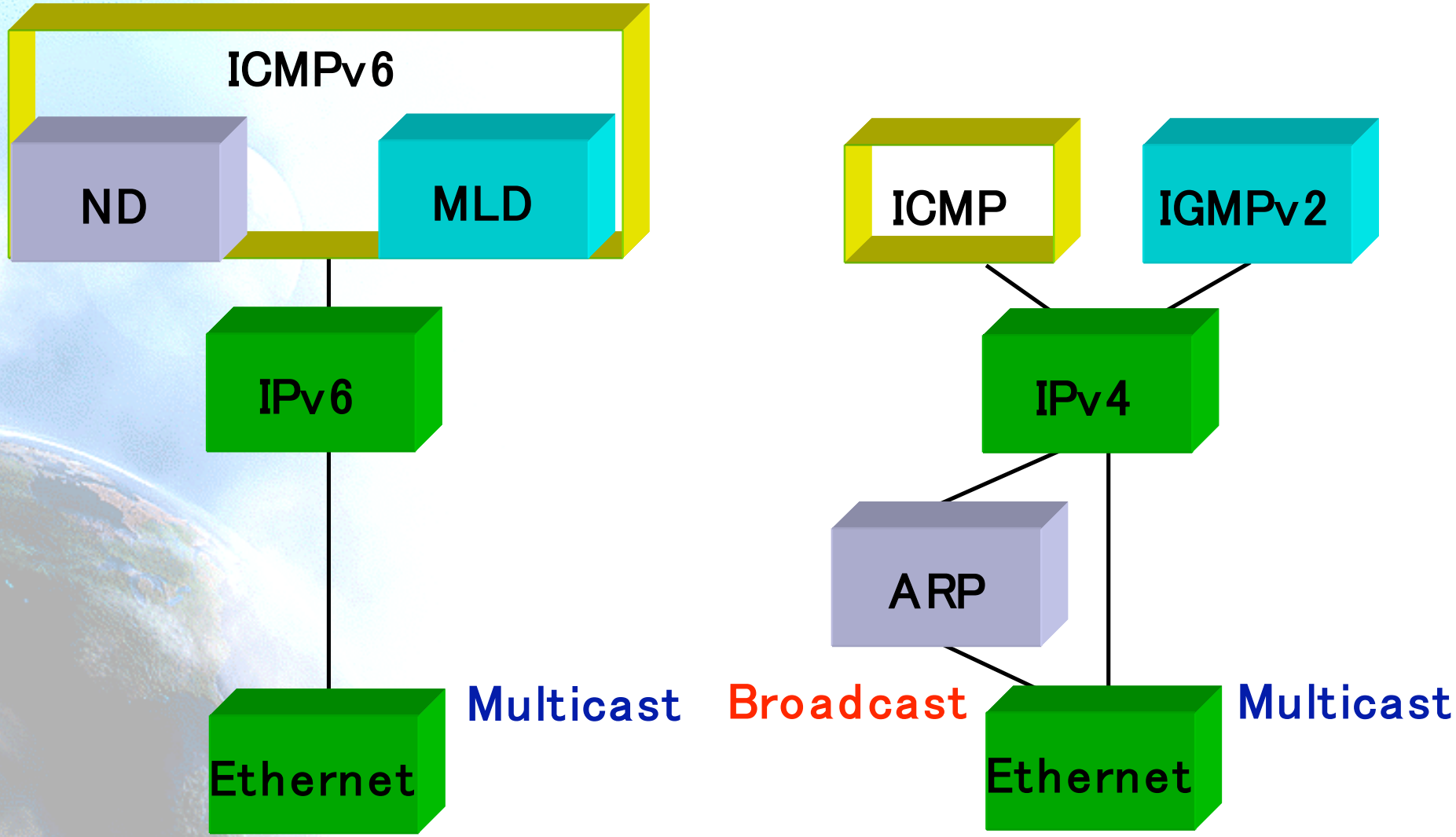
- "Next Header" Field



Extension Headers Goodies

- Processed Only by Destination Node
  - Exception: Hop-by-Hop Options Header
- No more “40 byte limit” on options (IPv4)
- Extension Headers defined currently:
  - Hop-by-Hop Options
  - Routing
  - Fragment
  - Authentication (RFC 2402, next header = 51)
  - Encapsulating Security Payload (RFC 2406, next header = 50)
  - Destination Options

Control Plane IPv4 vs. IPv6



IPv6 Tutorial

Addressing and Routing

Text Representation of Addresses

“Preferred” form:	1080:0:FF:0:8:800:200C:417A
Compressed form:	FF01:0:0:0:0:0:0:43
	becomes FF01::43
IPv4-compatible:	0:0:0:0:0:0:13.1.68.3
	or ::13.1.68.3
URL:	http://[FF01::43]/index.html

Address Types

- Unicast (one-to-one)
  - global
  - link-local
  - site-local (deprecated)
  - IPv4-compatible
- Multicast (one-to-many)
- Anycast (one-to-nearest)
- Reserved

Address Type Prefixes

address type	binary prefix
IPv4-compatible	0000...0 (96 zero bits)
Global unicast	001
Link-local unicast	1111 1110 10
Site-local unicast	1111 1110 11 (deprecated)
Multicast	1111 1111

- All other prefixes reserved (approx. 7/8ths of total)
- Anycast addresses allocated from unicast prefixes

Interface IDs

The lowest-order 64-bit field of unicast addresses may be assigned in several different ways:

- auto-configured from a 48-bit MAC address (e.g., Ethernet address), expanded into a 64-bit EUI-64
- assigned via DHCP
- manually configured
- auto-generated pseudo-random number (to counter some privacy concerns)
- possibly other methods in the future

Some Special-Purpose Unicast Addresses

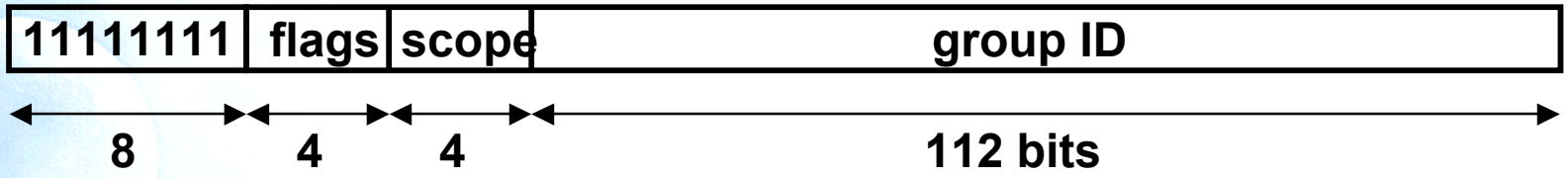
- The unspecified address, used as a placeholder when no address is available:

  0:0:0:0:0:0:0:0

- The loopback address, for sending packets to self:

  0:0:0:0:0:0:0:1

Multicast Addresses

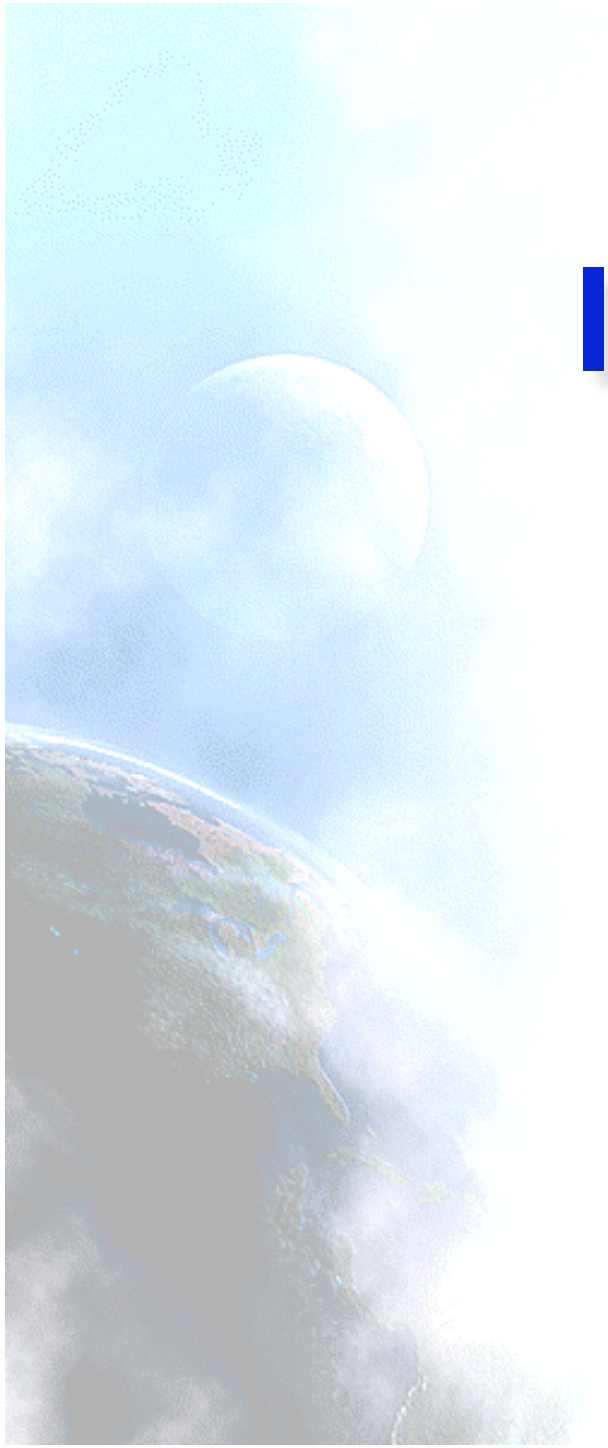


- Low-order flag indicates permanent/transient group; three other flags reserved
- Scope field:
  1 - node local
  2 - link-local
  5 - site-local
  8 - organization-local
  B - community-local
  E - global

  (all other values reserved)

Routing

- Uses same “longest-prefix match” routing as IPv4 CIDR
- Straightforward changes to existing IPv4 routing protocols to handle bigger addresses
  - –unicast: OSPF, RIP-II, IS-IS, BGP4+, …
  - –multicast: MOSPF, PIM, …
- Can use Routing header with anycast addresses to route packets through particular regions
  - –e.g., for provider selection, policy, performance, etc.



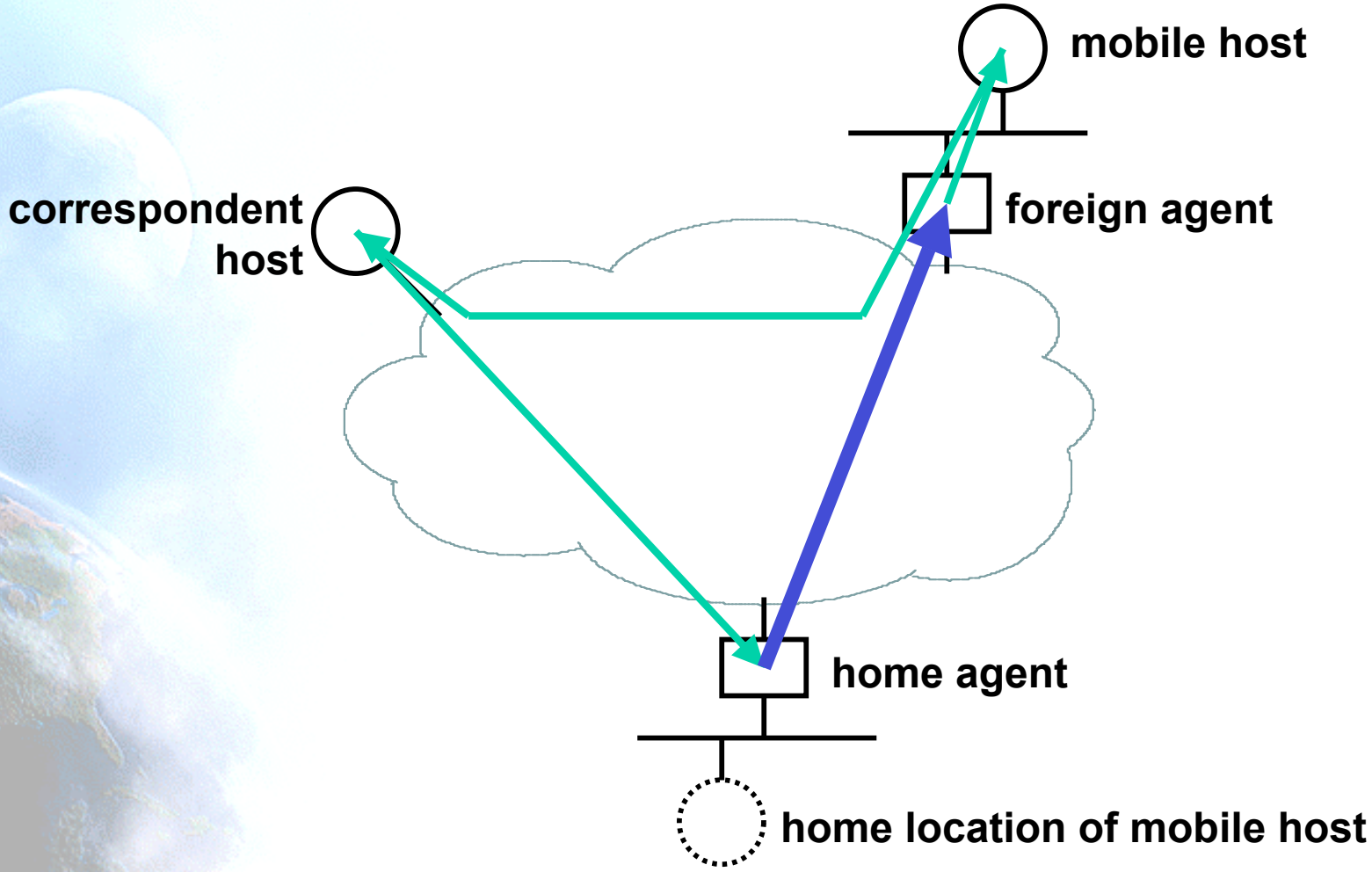
IPv6 Tutorial

Mobility

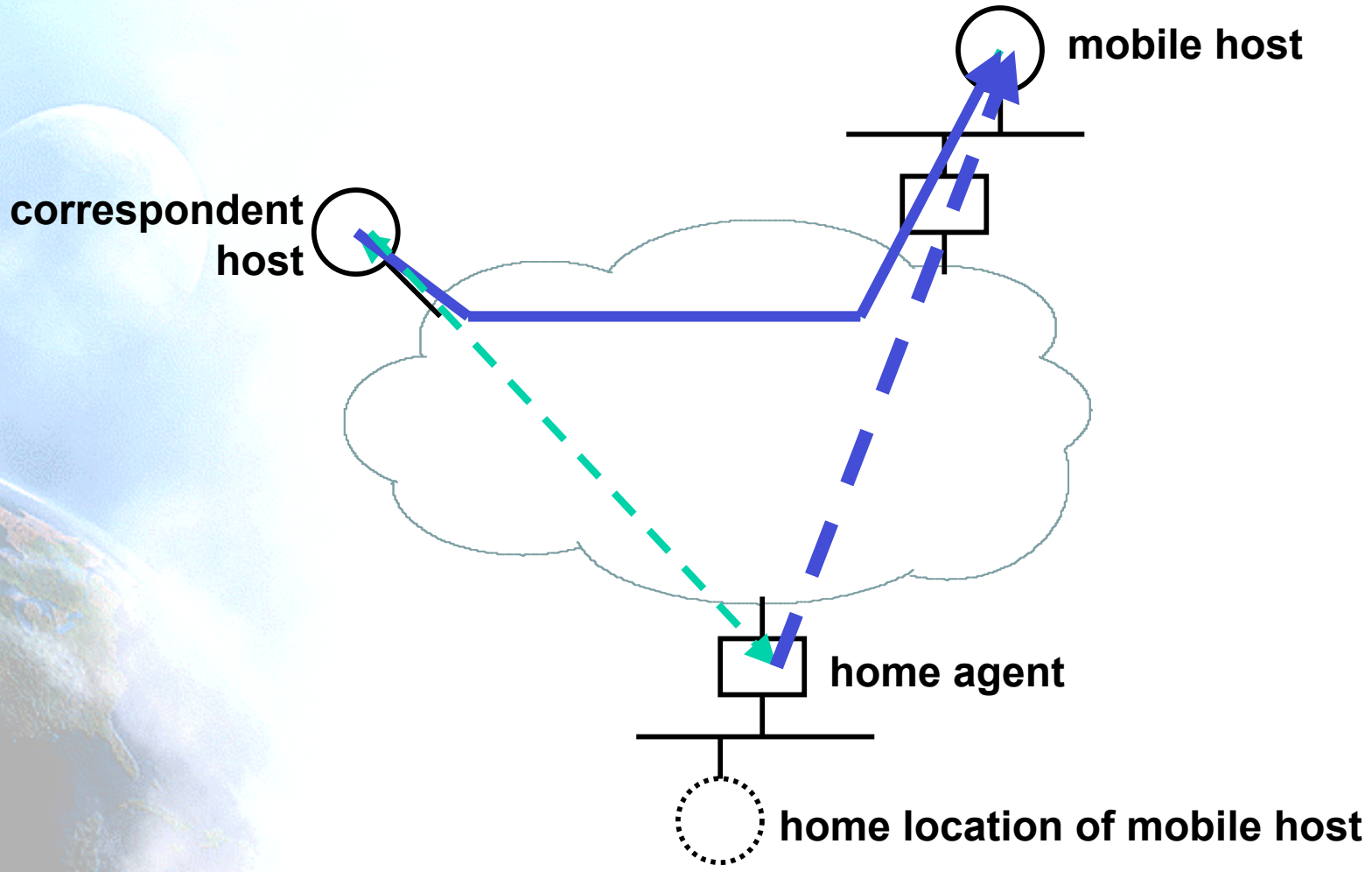
IPv6 Mobility

- A mobile host has one or more home address(es)
  - relatively stable; associated with host name in DNS
- When it discovers it is in a foreign subnet (i.e., not its home subnet), it acquires a foreign address
  - uses auto-configuration to get the address
  - registers the foreign address with a home agent, i.e, a router on its home subnet
- Packets sent to the mobile's home address(es) are intercepted by home agent and forwarded to the foreign address, using encapsulation

Mobile IP (v4 version)



Mobile IP (v6 version)



IPv6 Tutorial

IPv4-IPv6 Coexistence & Transition

Transition / Co-Existence Techniques

A wide range of techniques have been identified and implemented, basically falling into three categories:

(1) dual-stack techniques, to allow IPv4 and IPv6 to co-exist in the same devices and networks

(2) tunneling techniques, to avoid order dependencies when upgrading hosts, routers, or regions

(3) translation techniques, to allow IPv6-only devices to communicate with IPv4-only devices

Expect all of these to be used, in combination

Dual-Stack Approach

- When adding IPv6 to a system, do not delete IPv4
  - this multi-protocol approach is familiar and well-understood (e.g., for AppleTalk, IPX, etc.)
  - note: in most cases, IPv6 will be bundled with new OS releases, not an extra-cost add-on
- Applications (or libraries) choose IP version to use
  - when initiating, based on DNS response:
    - if (dest has AAAA record) use IPv6, else use IPv4
  - when responding, based on version of initiating packet
- This allows indefinite co-existence of IPv4 and IPv6, and gradual app-by-app upgrades to IPv6 usage
- A6 record is experimental

Tunnels to Get Through IPv6-Ignorant Routers

- Encapsulate IPv6 packets inside IPv4 packets (or MPLS frames)
- Many methods exist for establishing tunnels:
  - manual configuration
  - "tunnel brokers" (using web-based service to create a tunnel)
  - "6-over-4" (intra-domain, using IPv4 multicast as virtual LAN)
  - "6-to-4" (inter-domain, using IPv4 addr as IPv6 site prefix)
- Can view this as:
  - IPv6 using IPv4 as a virtual link-layer, or
  - an IPv6 VPN (virtual public network), over the IPv4 Internet (becoming "less virtual" over time, we hope)

Translation

- May prefer to use IPv6-IPv4 protocol translation for:
  - new kinds of Internet devices (e.g., cell phones, cars, appliances)
  - benefits of shedding IPv4 stack (e.g., serverless autoconfig)
- This is a simple extension to NAT techniques, to translate header format as well as addresses
  - IPv6 nodes behind a translator get full IPv6 functionality when talking to other IPv6 nodes located anywhere
  - they get the normal (i.e., degraded) NAT functionality when talking to IPv4 devices
  - methods used to improve NAT functionality (e.g, RSIP) can be used equally to improve IPv6-IPv4 functionality

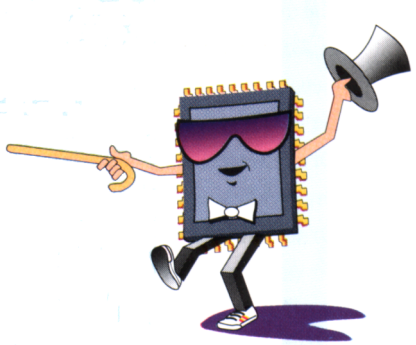
Thanks !

Contact:

- Jordi Palet Martínez (Consulintel): jordi.palet@consulintel.es

Madrid 2005 IPv6 Summit, info available at:

www.ipv6-es.com



THE IPv6 PORTAL

Consulintel
Consultores Integrales en Telecomunicaciones