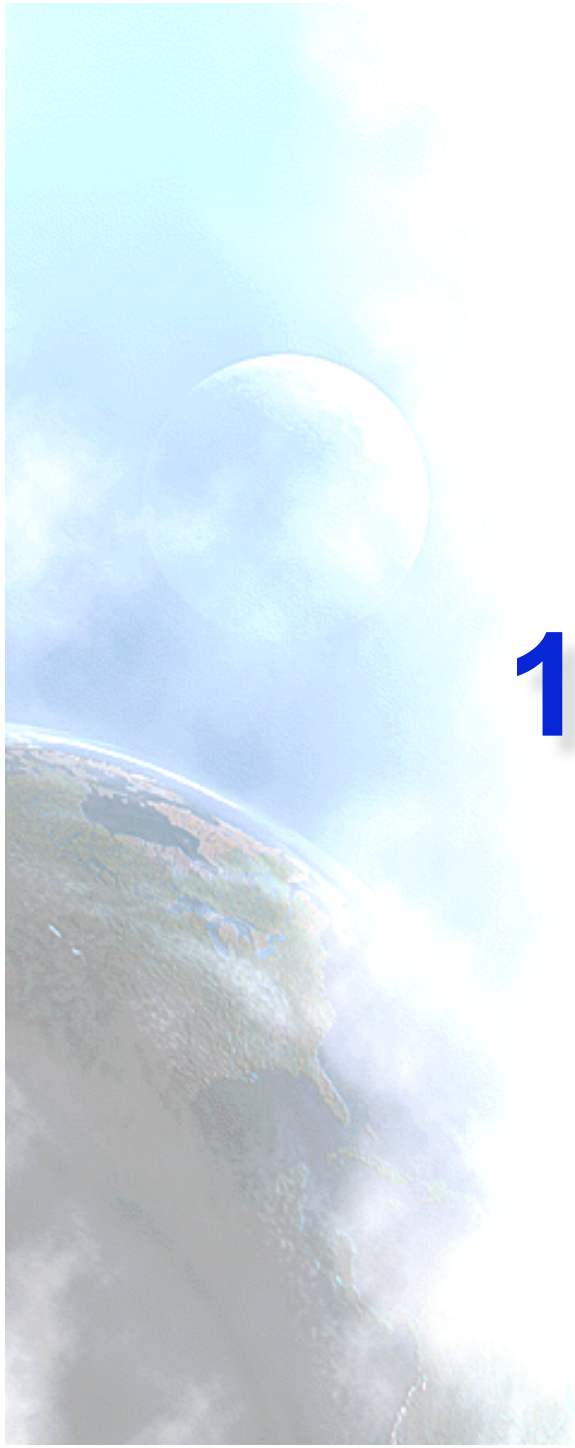
IPv6 & 3G Tutorial

Jordi Palet (jordi.palet@consulintel.es)
Education, Promotion, Public Relations
and Awareness Working Group Chair
IPv6 Forum

Agenda

1. Introduction & Background
2. Header Formats & Packet Size Issues
3. Addressing & Routing
4. Security
5. Quality of Service
6. Mobility
7. ICMPv6 & Neighbor Discovery
8. Multicast
9. IPv4-v6 Coexistence & Transition
10. Porting Applications to IPv6
11. Current Status
12. IPv6 in 3G



IPv6 Tutorial

1. Introduction & Background

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

2. Header Formats & Packet Size Issues

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

Agenda

2.1. Terminology
2.2. IPv6 Header Format
2.3. Packet Size Issues
2.4. Upper-Layer Protocol Issues

2.1. Terminology

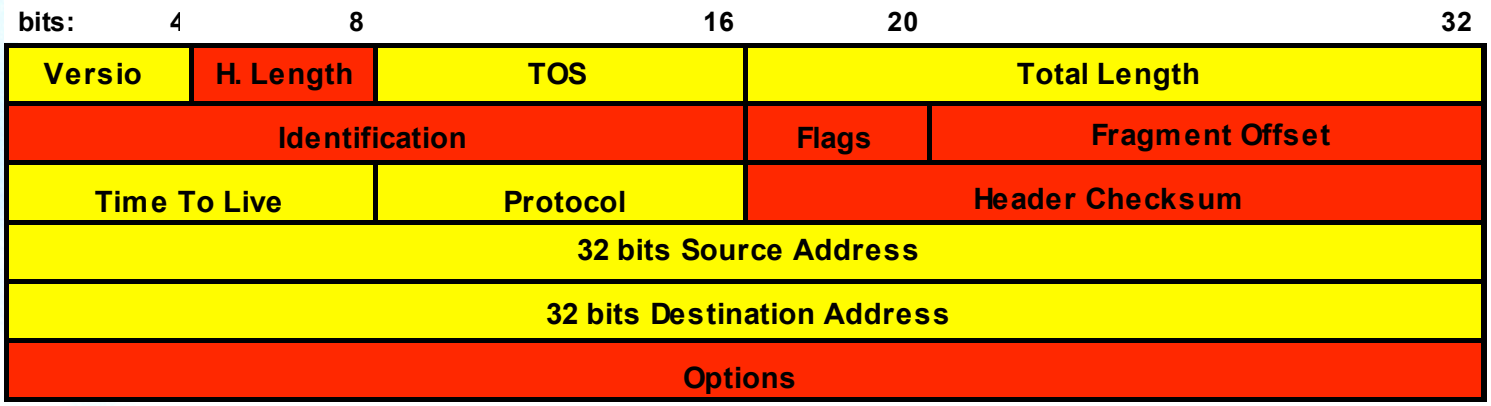
Terminology

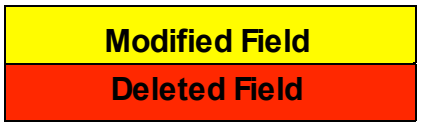
- **Node:** Device that implements IPv6
- **Router:** Node that forwards IPv6 packets
- **Host:** Any node that isn't a router
- **Upper Layer:** Protocol layer immediately above IPv6
- **Link:** Communication Facility or Medium over with nodes can communicate at the link layer
- **Neighbors:** Nodes attached to the same link
- **Interface:** A node´s attachment to a link
- **Address:** An IPv6-layer identification for an interface or a set of interfaces
- **Packet:** An IPv6 header plus payload
- **Link MTU:** Maximum Transmission Unit
- **Path MTU:** Minimum link MTU of all the links in a path between source and destination node's

2.2. IPv6 Header Format

IPv4 Header Format

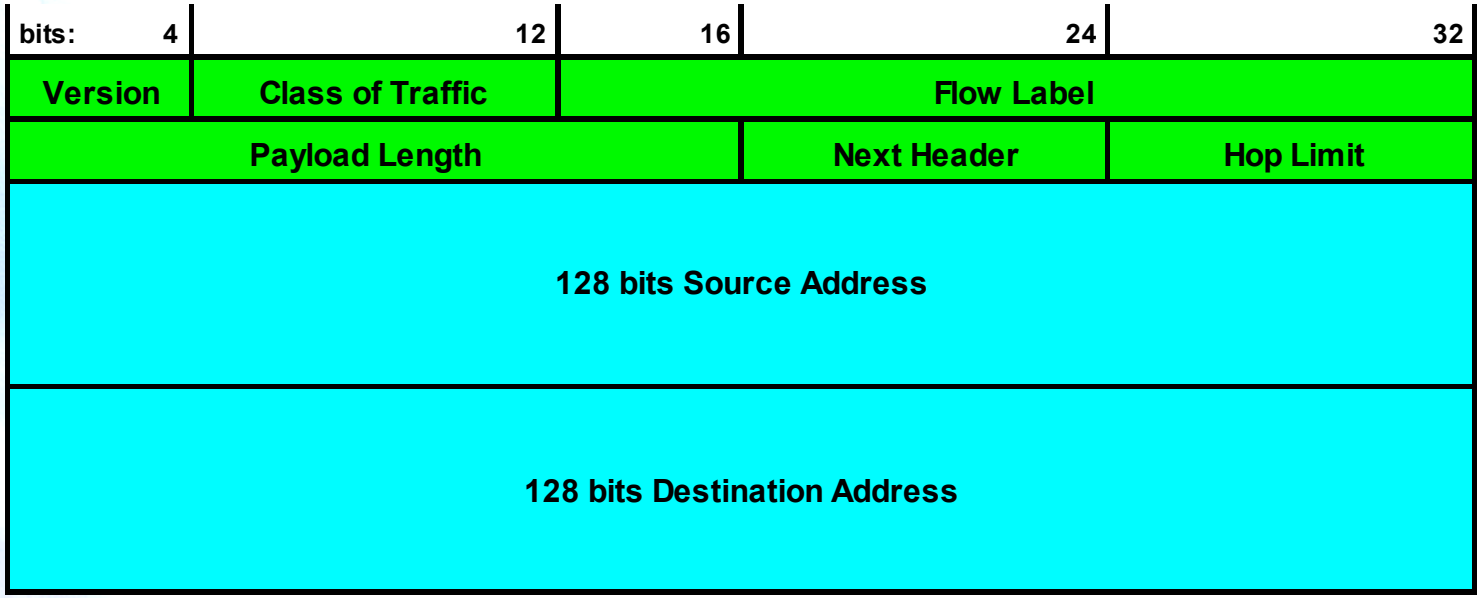
- 20 Bytes + Options





IPv6 Header Format

- From 12 to 8 Fields (40 bytes)



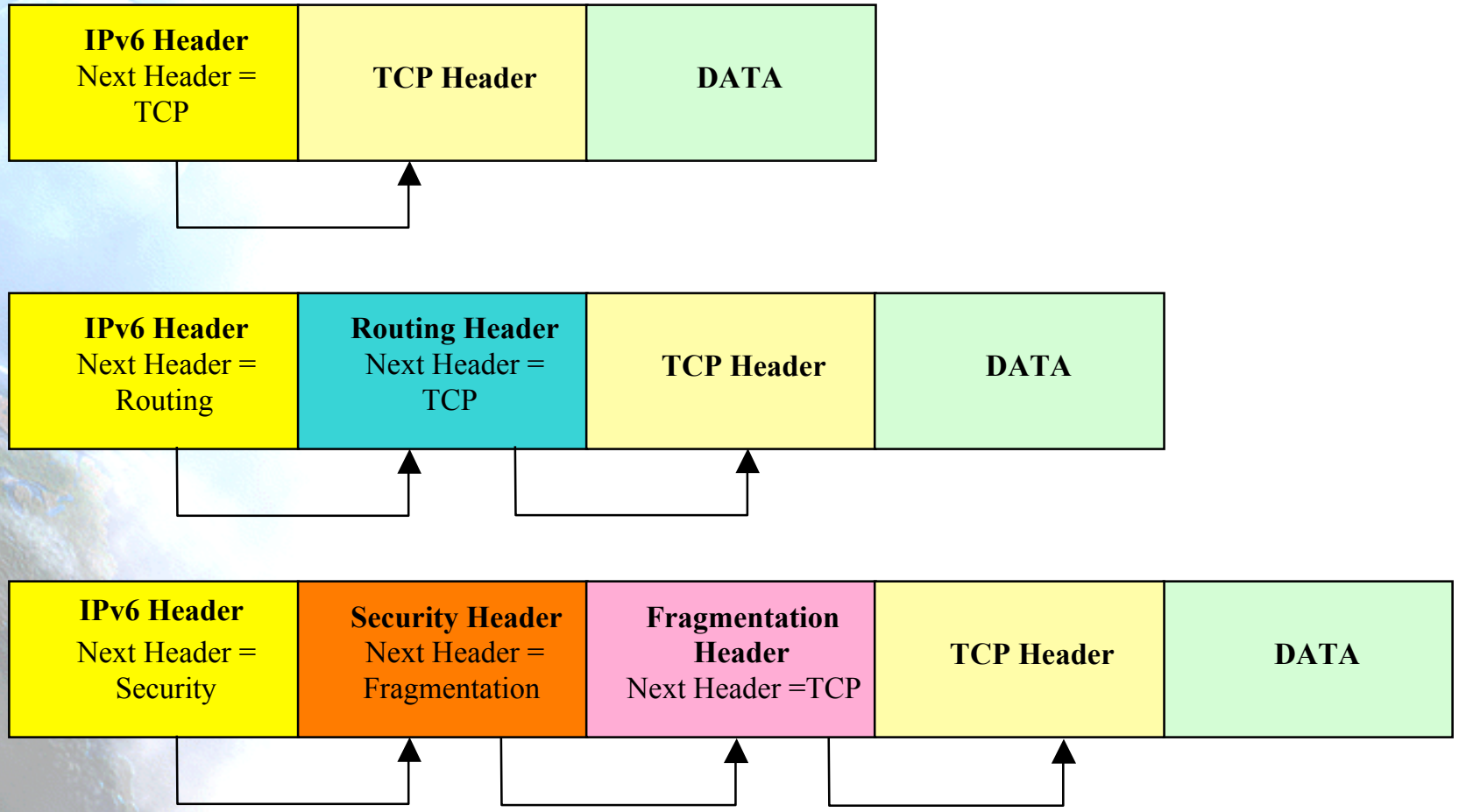
  - Avoid checksum redundancy
  - Fragmentation end to end

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



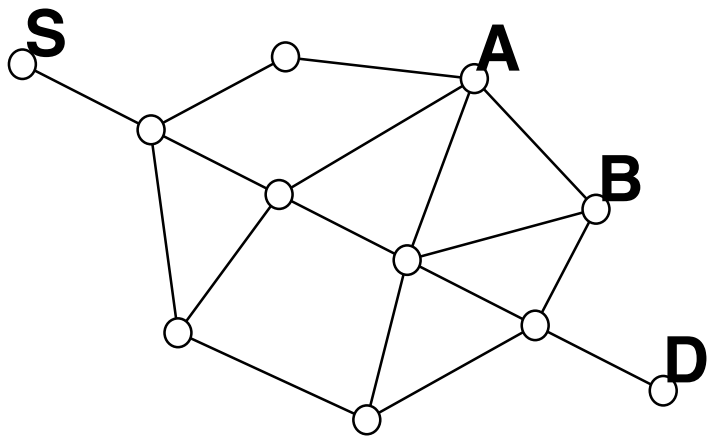
Extension Headers Goodies

- 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

Example: Using the Routing Header

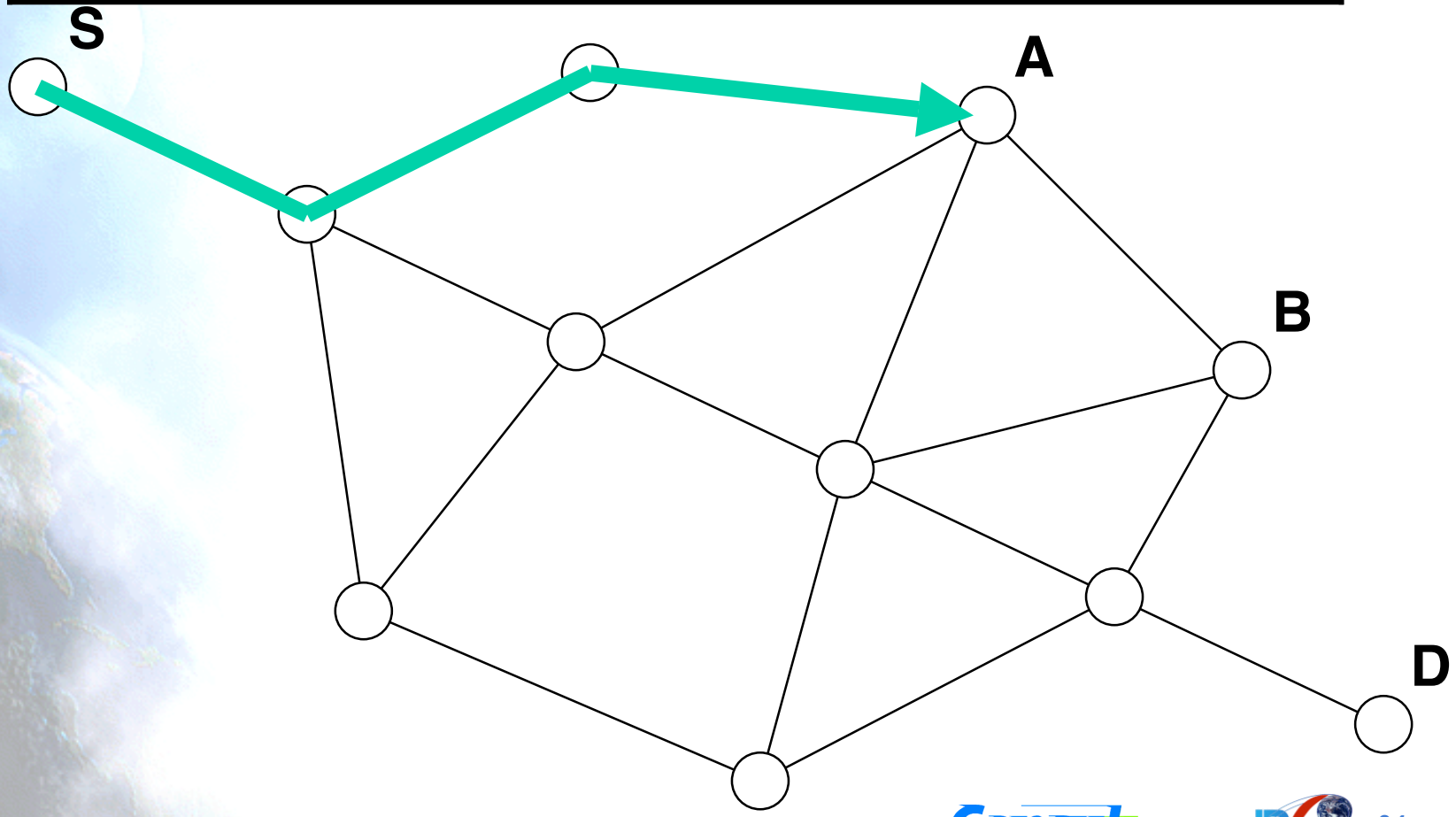
8 bits	8 bits unsigned	8 bits	8 bits unsigned
Next Header	**H. Ext. Length**	**Routing Type = 0**	**Segments Left**
Reserved = 0			
Address 1			
Address 2			
…			
Address n			

- Next Header value = 43
- A type 0 routing header, where:
  - Source Node: S
  - Destination: D
  - Intermediate Nodes: A & B



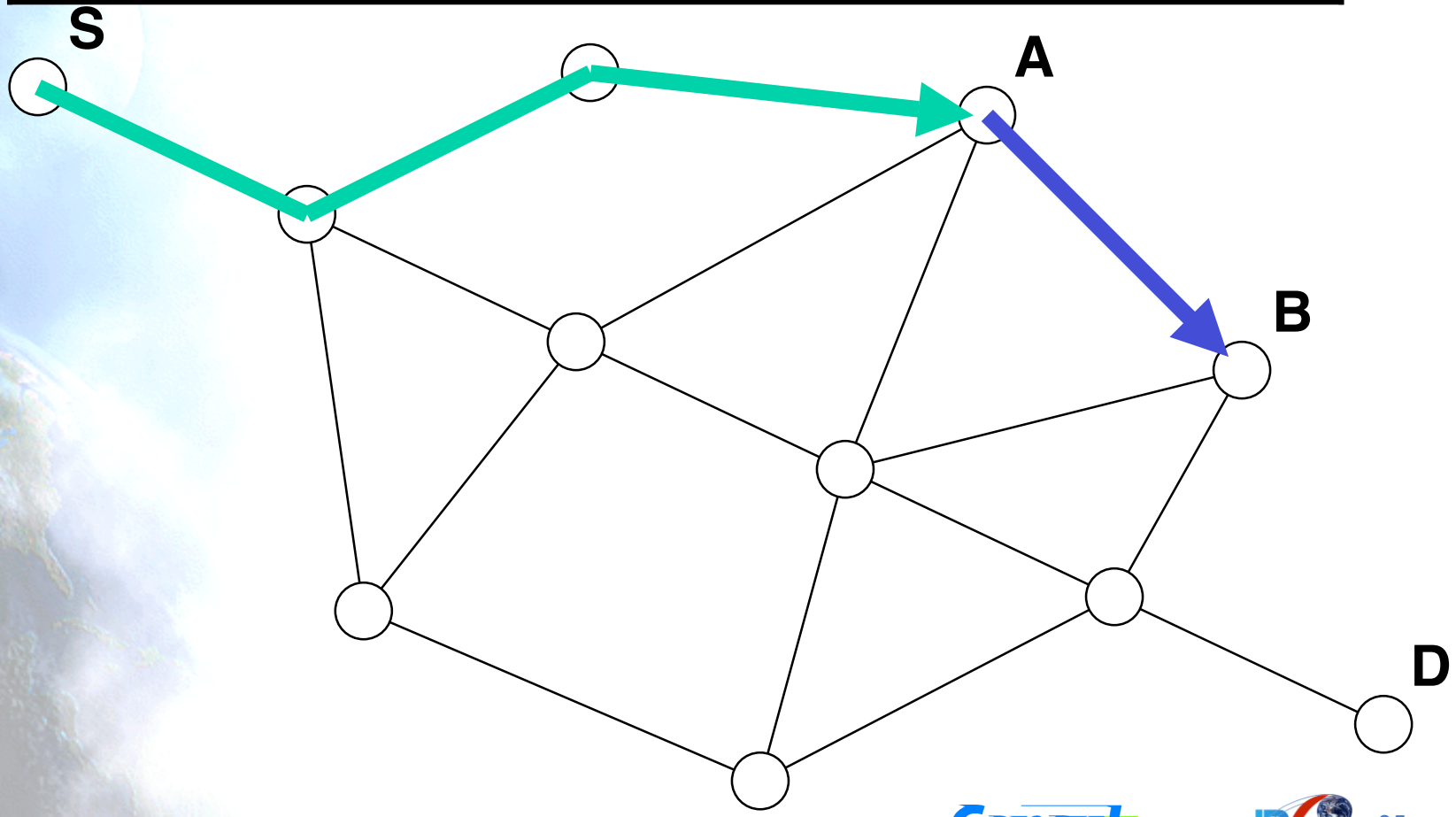
Example: Headers when S to A

IPv6 Base Header	Routing Header
Source Address = S Destination Address = A	H. Ext. Length = 4 Segments Left = 2 Address 1 = B Address 2 = D



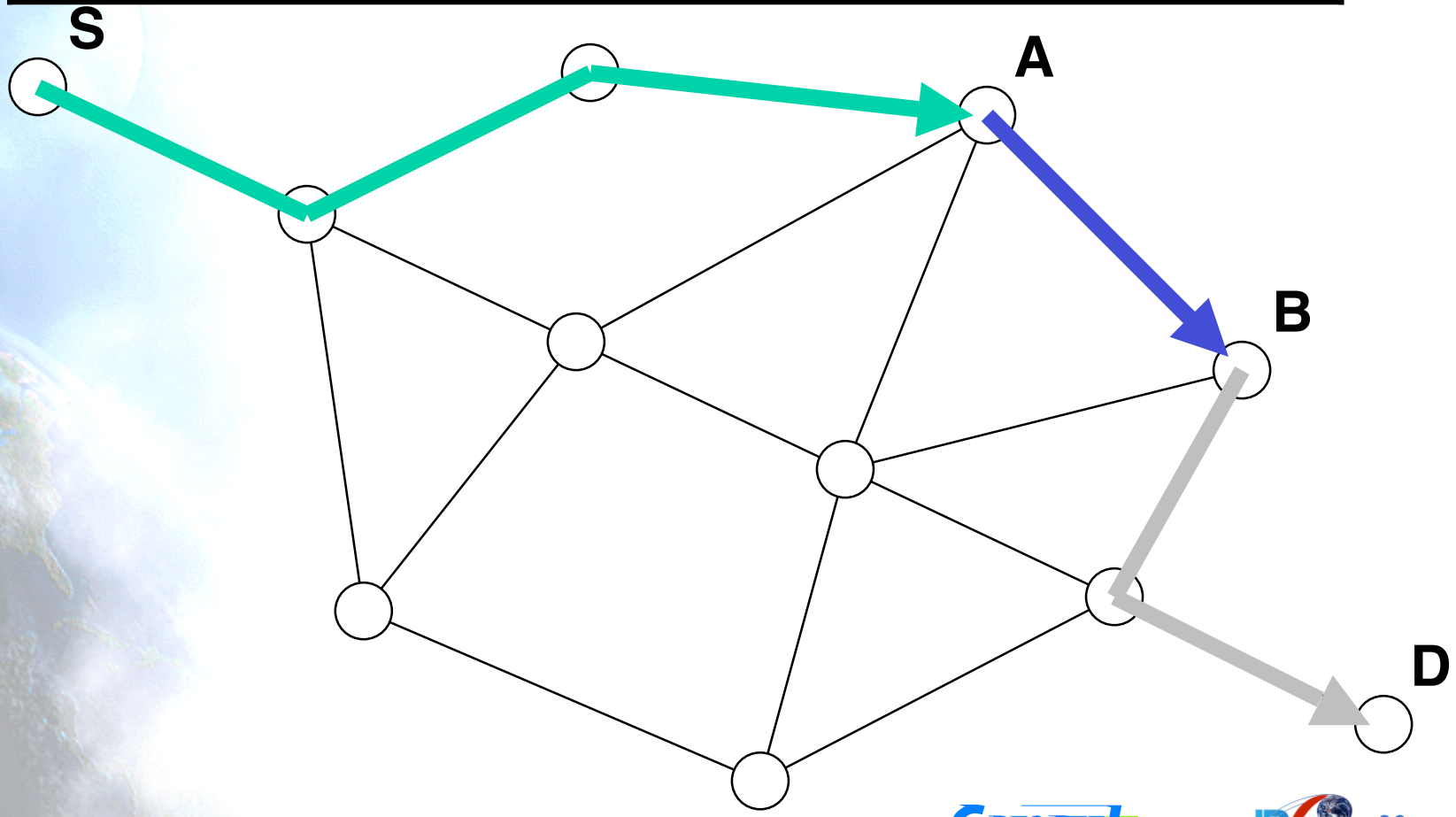
Example: Headers when A to B

IPv6 Base Header	Routing Header
Source Address = S Destination Address = B	H. Ext. Length = 4 Segments Left = 1 Address 1 = A Address 2 = D



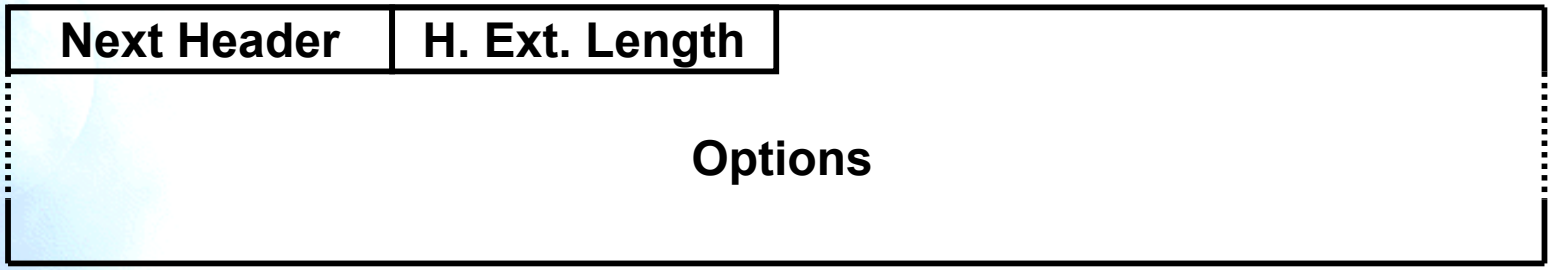
Example: Headers when B to D

IPv6 Base Header	Routing Header
Source Address = S Destination Address = D	H. Ext. Length = 4 Segments Left = 0 Address 1 = A Address 2 = B



Hop-by-Hop & Destination Options Headers

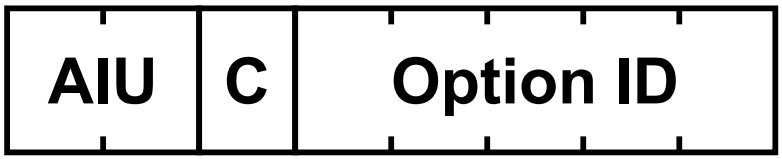
- "Containers" for variable-length options:



- Where Options =

- Next Header values:
  - 0 for Hop-by-Hop Options Header
  - 60 for Destination Options Header

Option Type Encoding



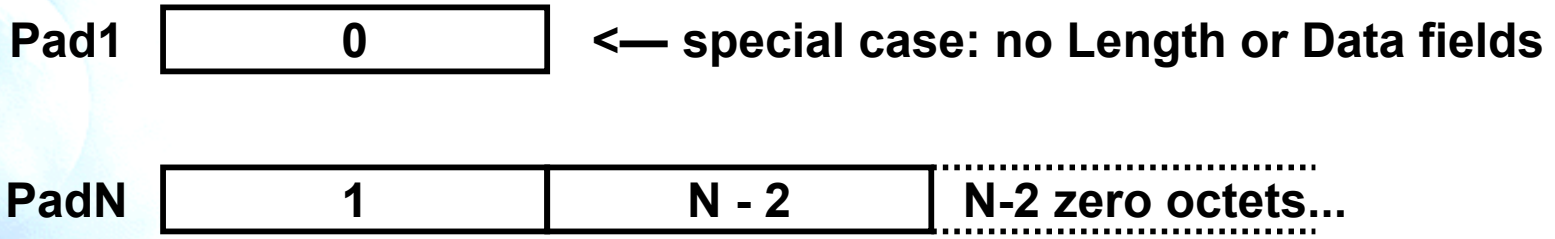
AIU — action if unrecognized:

00 — skip over option
01 — discard packet
10 — discard packet &
send ICMP Unrecognized Type to source
11 — discard packet &
send ICMP Unrecognized Type to source
only if destination was not multicast

C — set (1) if Option Data changes en-route
(Hop-by-Hop Options only)

Option Alignment and Padding

Two Padding Options:



- Used to align options so multi-byte data fields fall on natural boundaries
- Used to pad out containing header to an integer multiple of 8 bytes

Fragment Header

- Used by an IPv6 Source to send a packet larger than would fit in the path MTU to its Destination.
- In IPv6 the Fragmentation is only performed by source nodes, not routers.
- Next Header value = 44

8 bits	8 bits	13 bits unsigned	2 bits	1 bit
Next Header	**Reserved = 0**	**Fragment Offset**	**Res. = 0**	**M**
Identification				

- Original Packet (unfragmented):

Unfragmentable Part	**Fragmentable Part**

Fragmentation Process

- The Fragmentable Part of the original packet is divided into fragments, each, except possibly the last ("rightmost") one, being an integer multiple of 8 octets long. The fragments are transmitted in separate "fragment packets“

Unfragmentable Part	1st Fragment	2nd Fragment	...	Last Fragment

- Fragment Packets:

Unfragmentable Part	Fragment Header	1st Fragment

Unfragmentable Part	Fragment Header	2nd Fragment

• • •

Unfragmentable Part	Fragment Header	Last Fragment

2.3. Packet Size Issues

Minimum MTU

- Link MTU:
  - A link's maximum transmission unit, i.e., the max IP packet size that can be transmitted over the link
- Path MTU:
  - The minimum MTU of all the links in a path between a source and a destination
- Minimum link MTU for IPv6 is 1280 octets vs. 68 octets for v4
- On links with MTU < 1280, link-specific fragmentation and reassembly must be used
- On links that have a configurable MTU, it's recommended a MTU of 1500 bytes

Path MTU Discovery (RFC1981)

- Implementations are expected to perform path MTU discovery to send packets bigger than 1280 octets:
  - for each destination, start by assuming MTU of first-hop link
  - if a packet reaches a link in which it can't fit, will invoke ICMP "packet too big" message to source, reporting the link's MTU; MTU is cached by source for specific destination
  - occasionally discard cached MTU to detect possible increase
- Minimal implementation can omit path MTU discovery as long as all packets kept ≤ 1280 octets
  - e.g., in a boot ROM implementation

Fragment Header

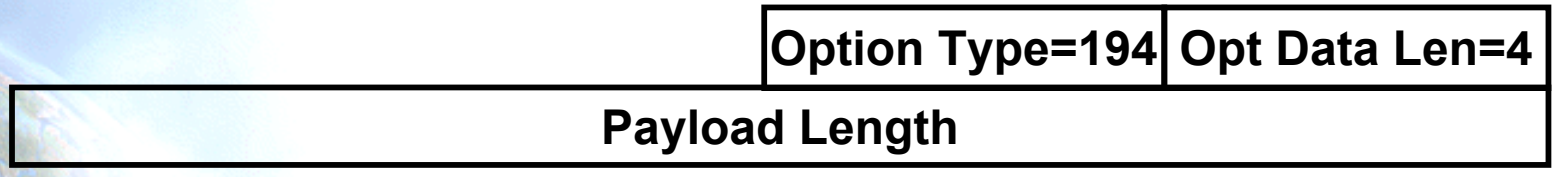
Next Header	Reserved	Fragment Offset	0 0 M
Original Packet Identifier			

- Though discouraged, can use IPv6 Fragment header to support upper layers that do not (yet) do path MTU discovery

- IPv6 fragmentation & reassembly is an end-to-end function; routers do not fragment packets en-route if too big, instead, they send ICMP “packet too big”.

Maximum Packet Size

- Base IPv6 header supports payloads of up to 65,535 bytes (not including 40 byte IPv6 header)
- Bigger payloads can be carried by setting IPv6 Payload Length field to zero, and adding the “jumbogram” hop-by-hop option:



- Can’t use Fragment header with jumbograms (RFC2675)

2.4. Upper-Layer Protocol Issues

Upper-Layer Checksums

- Any transport or other upper-layer protocol that includes the addresses from the IP header in its checksum computation must be modified for use over IPv6, to include the 128-bit IPv6 addresses instead of 32-bit IPv4 addresses.
- TCP/UDP “pseudo-header” for IPv6:

Source Address	
Destination Address	
Upper-Layer Packet Length	
zero	Next Header

- ICMPv6 includes the above pseudo-header in its checksum computation (change from ICMPv4). Reason: Protect ICMP from misdelivery or corruption of those fields of the IPv6 header on which it depends, which, unlike IPv4, are not covered by an internet-layer checksum. The Next Header field in the pseudo-header for ICMP contains the value 58, which identifies the IPv6 version of ICMP.

Maximum Packet Lifetime

- IPv6 nodes are not required to enforce maximum packet lifetime.
- That is the reason the IPv4 "Time to Live" field was renamed "Hop Limit" in IPv6.
- In practice, very few, if any, IPv4 implementations conform to the requirement that they limit packet lifetime, so this is not a “real” change.
- Any upper-layer protocol that relies on the internet layer (whether IPv4 or IPv6) to limit packet lifetime ought to be upgraded to provide its own mechanisms for detecting and discarding obsolete packets.

Maximum Upper-Layer Payload Size

- When computing the maximum payload size available for upper-layer data, an upper-layer protocol must take into account the larger size of the IPv6 header relative to the IPv4 header.
- Example: in IPv4, TCP's MSS option is computed as the maximum packet size (a default value or a value learned through Path MTU Discovery) minus 40 octets (20 octets for the minimum-length IPv4 header and 20 octets for the minimum-length TCP header). When using TCP over IPv6, the MSS must be computed as the maximum packet size minus 60 octets, because the minimum-length IPv6 header (i.e., an IPv6 header with no extension headers) is 20 octets longer than a minimum-length IPv4 header.

Responding to Packets Carrying Routing Headers

- When an upper-layer protocol sends one or more packets in response to a received packet that included a Routing header, the response packet(s) must not include a Routing header that was automatically derived by "reversing" the received Routing header UNLESS the integrity and authenticity of the received Source Address and Routing header have been verified (e.g., via the use of an Authentication header in the received packet).

IPv6 Tutorial

3. 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

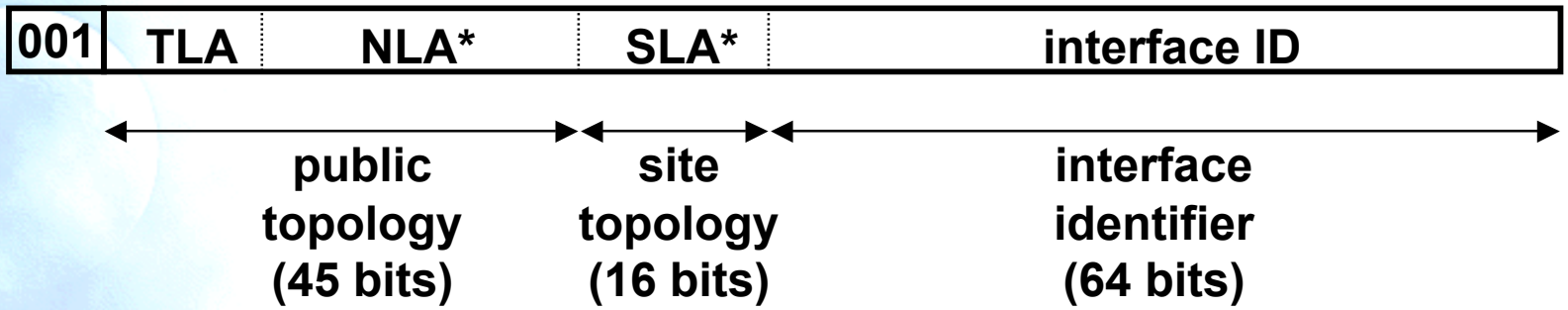
- Unicast (one-to-one)
  - global
  - link-local
  - site-local
  - IPv4-compatible
- Multicast (one-to-many)
- Anycast (one-to-nearest)
- Reserved

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
Multicast	1111 1111

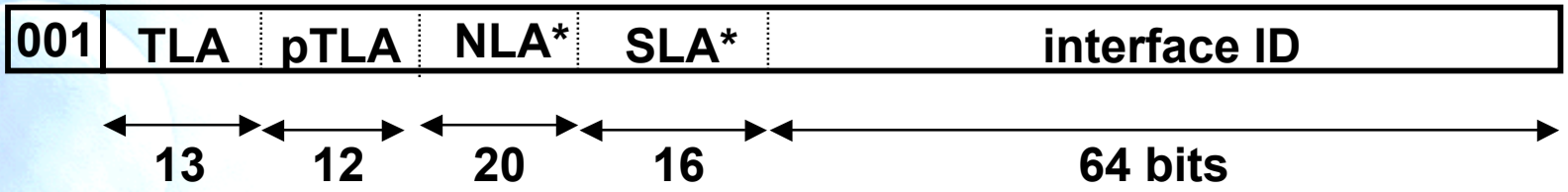
- All other prefixes reserved (approx. 7/8ths of total)
- Anycast addresses allocated from unicast prefixes

Global Unicast Addresses



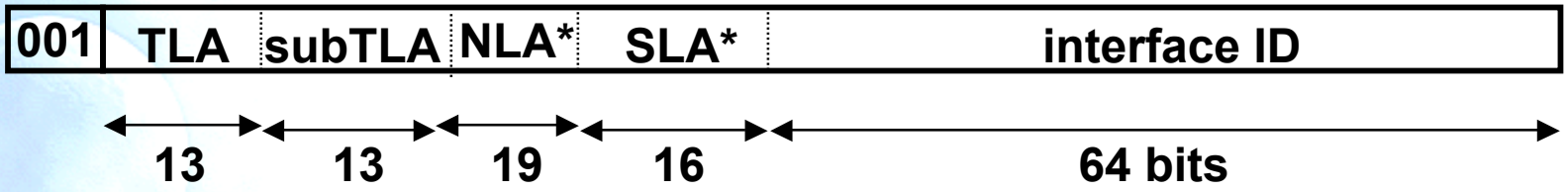
- TLA = Top-Level Aggregator
  NLA* = Next-Level Aggregator(s)
  SLA* = Site-Level Aggregator(s)
- all subfields variable-length, non-self-encoding (like CIDR)
- TLAs may be assigned to providers or exchanges

Global Unicast Addresses for the 6Bone



- 6Bone: experimental IPv6 network used for testing only
- TLA 1FFE (hex) assigned to the 6Bone
  - thus, 6Bone addresses start with 3FFE:
  - (binary 001 + 1 1111 1111 1110)
- Next 12 bits hold a “pseudo-TLA” (pTLA)
  - thus, each 6Bone pseudo-ISP gets a /28 prefix
- Not to be used for production IPv6 service

Global Unicast Addresses for Production Service

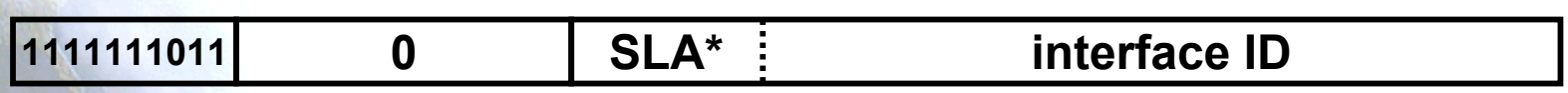


- ISPs start with less space than a TLA; must demonstrate need before getting a TLA ("slow-start" procedure)
- TLA 1 assigned for slow-start allocations
  - thus, initial production addresses start with 2001:
  - (binary 001 + 0 0000 0000 0001)
- Next 13 bits hold a subTLA
  - thus, each new ISP gets a /29 prefix
  - (or even longer, depending on registry policy)

Link-Local & Site-Local Unicast Addresses

Link-local addresses for use during auto-configuration and when no routers are present:

Site-local addresses for independence from changes of TLA / NLA*:



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

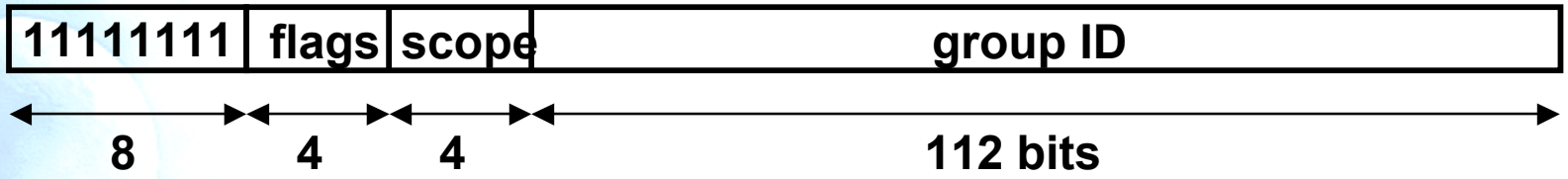
- The unspecified address, used as a placeholder when no address is available:

  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

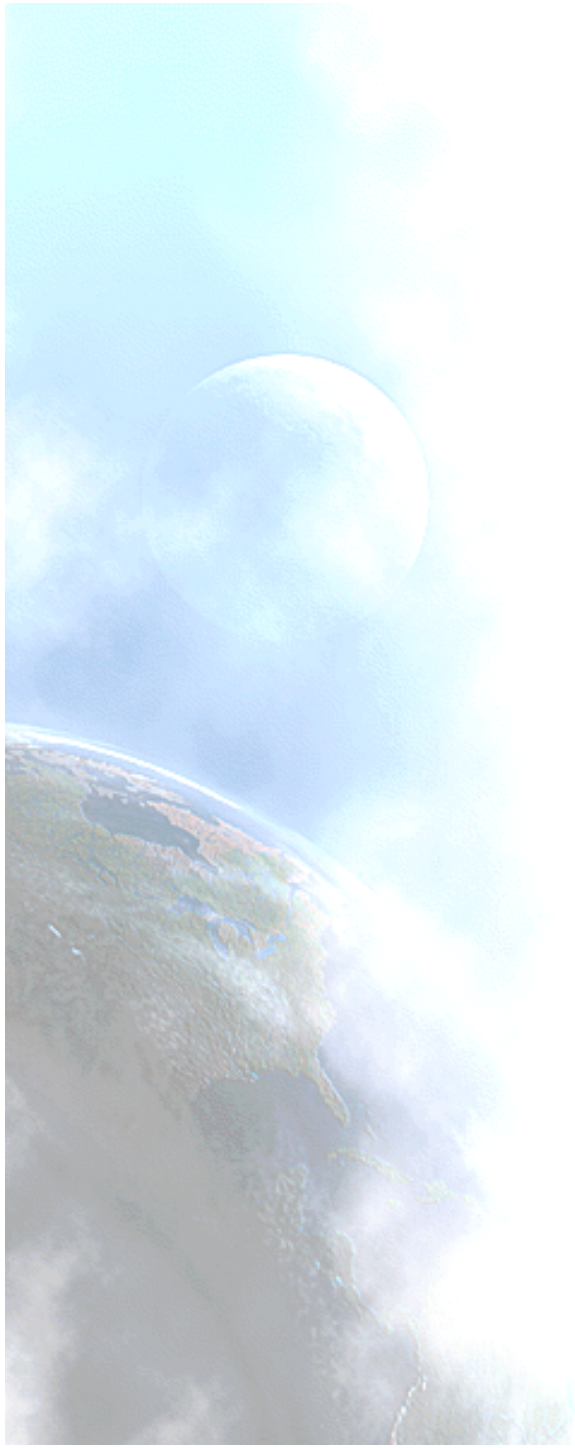


- Low-order flag indicates permanent/transient group; three other flags reserved
- Scope field:
  1 - node local
  2 - link-local
  5 - site-local
  8 - organization-local
  B - community-local
  E - global

  (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

4. Security

Agenda

4.1. Basic Concepts
4.2. Security Associations
4.3. IPsec Headers
4.4. Transport and Tunnel Modes
4.5. Key Management

4.1. Basic Concepts

IP Security

- RFC2401: Base architecture for IPsec compliant systems
- Goal: Provide various security services for traffic at the IP layer, in both IPv4 and IPv6 environments.
  - Security Protocols -- Authentication Header (AH – RFC2402, authentication ONLY) and Encapsulating Security Payload (ESP – RFC2406, encryption + authentication)
  - Security Associations - what they are and how they work, how they are managed, associated processing (RFC2407, RFC2408, RFC2412)
  - Key Management - manual and automatic: The Internet Key Exchange (IKE – RFC2409, ISAKMP, OAKLEY)
  - Algorithms for authentication and encryption

Security Services Set

- Security Services Set:
  - Access control
  - Connectionless integrity
  - Data origin authentication
  - Protection against replays (a form of partial sequence integrity)
  - Confidentiality (encryption)
  - Limited traffic flow confidentiality.
- IPsec is designed to provide interoperable, high quality, cryptographically-based security for IPv4 and IPv6.

Traffic Security Protocols

- How to:
  - Authentication Header (AH)
  - Encapsulating Security Payload (ESP)
  - Use of cryptographic key management procedures and protocols.
  - The set of IPsec protocols employed in any context, and the ways in which they are employed, will be determined by the security and system requirements of users, applications, and/or sites/organizations.
    - IPsec allows the user/system administrator to control the granularity at which a security service is offered.
- These mechanisms are designed to be algorithm-independent.
- IPsec can be used to protect one or more "paths" between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

Modes of Use

- AH & ESP may be applied alone or in combination with each other to provide a desired set of security services in IPv4 and IPv6.
- Each protocol supports two modes of use:
  - Transport mode (protection primarily for upper layer protocols)
    - Direct between end-to-end systems
    - Both Remote systems must support IPsec !
  - Tunnel mode (protocols applied to tunneled IP packets)
    - Secure tunnel for encapsulating insecure IP packets
    - Between intermediate systems (not end-to-end)

IPv6 Security

- IPsec is part of the IPv6 “core” specs:
  - All implementations expected to support authentication and encryption headers (“IPsec”)
- Authentication separate from encryption for use in situations where encryption is prohibited or prohibitively expensive
- Key distribution protocols are under development (independent of IP v4/v6)
- Support for manual key configuration required

4.2. Security Associations

The Concept

- Security Association (SA) is a fundamental concept for IPsec:
  - **A simplex "connection" that affords security services to the traffic carried by it.**
- AH & ESP use SA's.
- A major function of IKE is the establishment and maintenance of Security Associations.
- All implementations of AH & ESP MUST support the concept of a Security Association.

SA Identification

- Each SA is uniquely identified by a triple:
  - Security Parameter Index (SPI)
    - Bit String Assigned to the SA (local meaning), as a pointer to a SA Database (SPD or Security Policy Database).
  - IP Destination Address
  - Security protocol (AH or ESP) identifier
- Destination Address may be:
  - Unicast Address
  - IP broadcast address
  - Multicast group address

SA Database (SAD)

- In each IPsec implementation there is a nominal Security Association Database.
- Each entry defines the parameters associated with one SA.
- Each SA has an entry in the SAD.

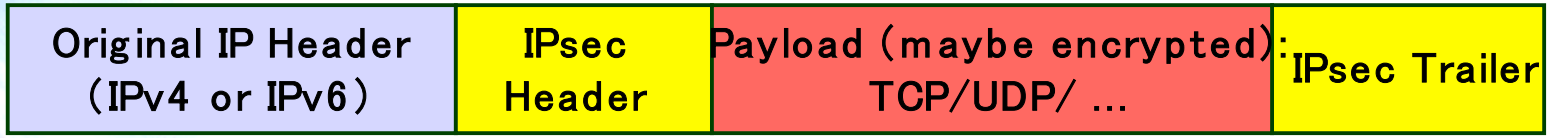
SAD Fields

- **Sequence Number Counter:** 32 bits value used to generate the sequence number transmitted in the AH and ESP headers.
- **Sequence Counter Overflow:** Indicates the action to trigger when the sequence number range is over.
- **Anti-Replay Window:** Window for limiting the acceptance of valid datagrams.
- **AH Information:** Authentication algorithms, keys, lifetimes, etc.
- **ESP Information:** Authentication and Encrypting algorithms, keys, lifetimes, initial values, etc.
- **IPsec Protocol Mode:** Transport, tunnel or wildcard.
- **SA Lifetime:** Time or bytes interval of a SA.
- **Path MTU:** Maximum packet size transmitted without fragmentation.

4.3. IPsec Headers

IPsec Transmission

- IPsec header inserted between the original header and the payload.
- If ESP is used, data is encrypted and am IPsec trailer is appended.

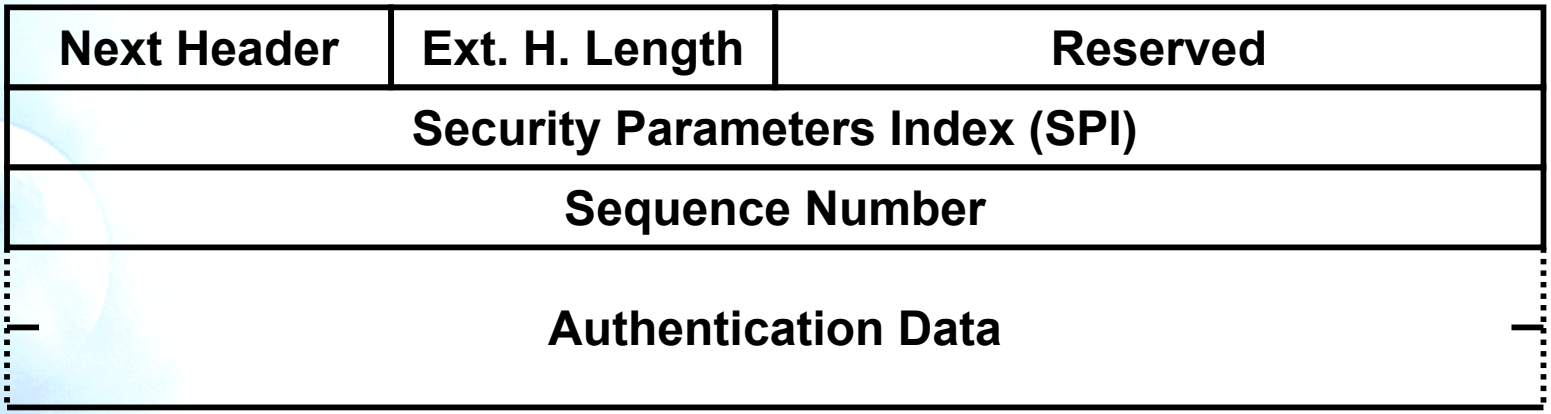


- Next Header value:
  - ESP = 50
  - AH = 51

Authentication Mode (RFC2402)

- Provides authentication and data integrity of the IP fields that don't change en-route:
  - Changes in the content are detected
  - Receivers can authenticate the sender
  - Avoids the IP-Spoofing attack
  - Protection against the replay attack.
- Default algorithms:
  - Keyed MD5
  - SHA-1

Authentication Header (AH)

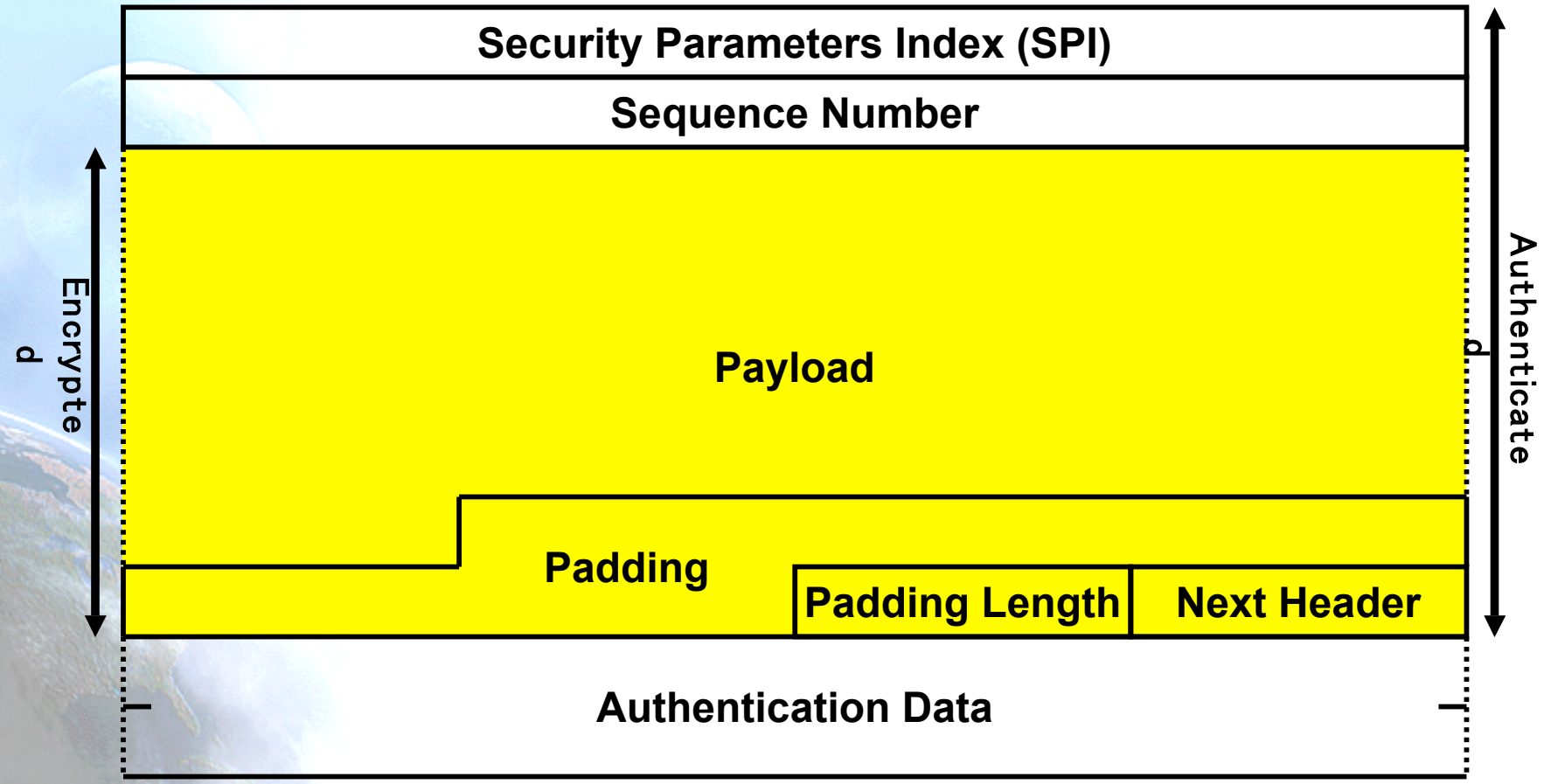


- SPI: Arbitrary 32-bit value that, in combination with the destination IP address and security protocol (AH), uniquely identifies the Security Association for this datagram.
- Sequence Number: Unsigned 32-bit field contains a monotonically increasing counter value.
- Authentication Data: Variable-length field that contains the Integrity Check Value (ICV) for this packet.

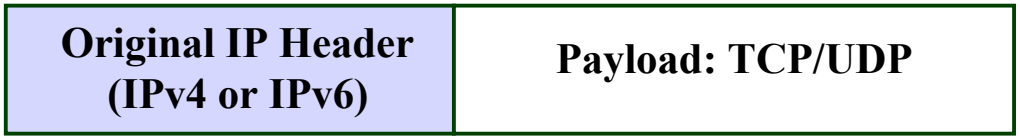
Encryption Mode (RFC2406)

- Provides:
  - Confidentiality
  - Data origin authentication
  - Connectionless integrity
  - Anti-Reply Service (Partial sequence integrity)
  - Limited traffic flow confidentiality

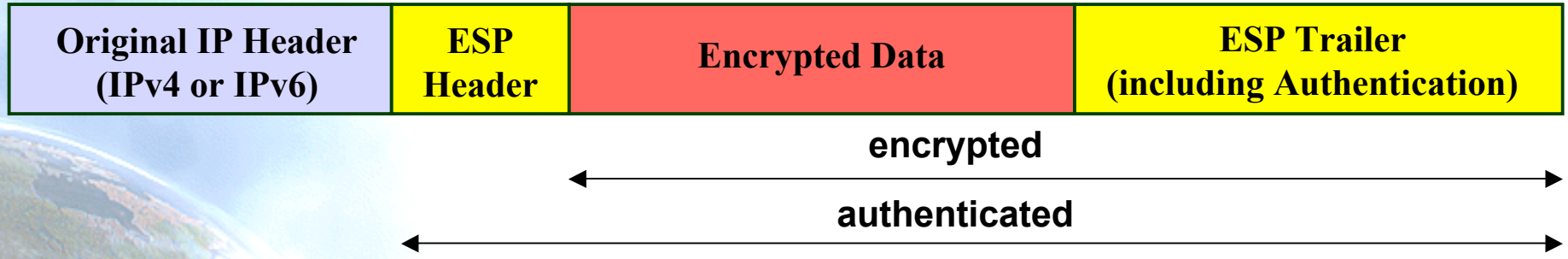
ESP Header


Security Parameters Index (SPI)
Sequence Number
Payload
Padding
Padding Length
Next Header
Authentication Data
Encrypted
Authenticated

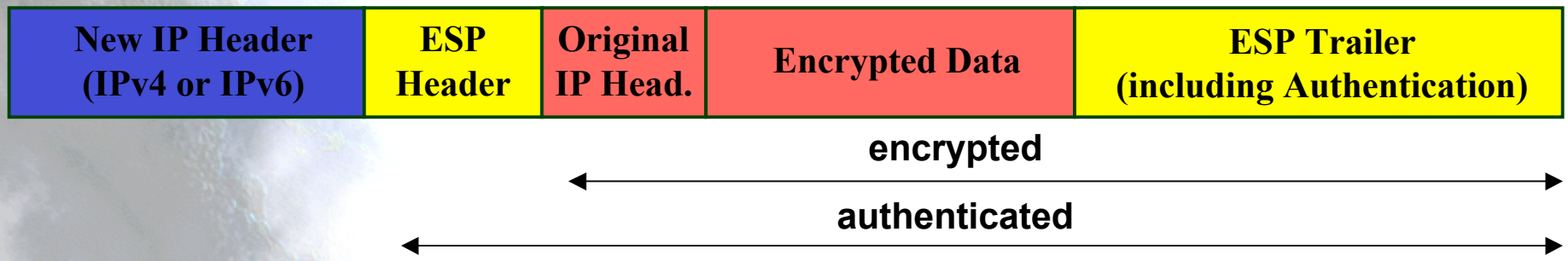
Transport and Tunnel Modes



Transport Mode



Tunnel Mode

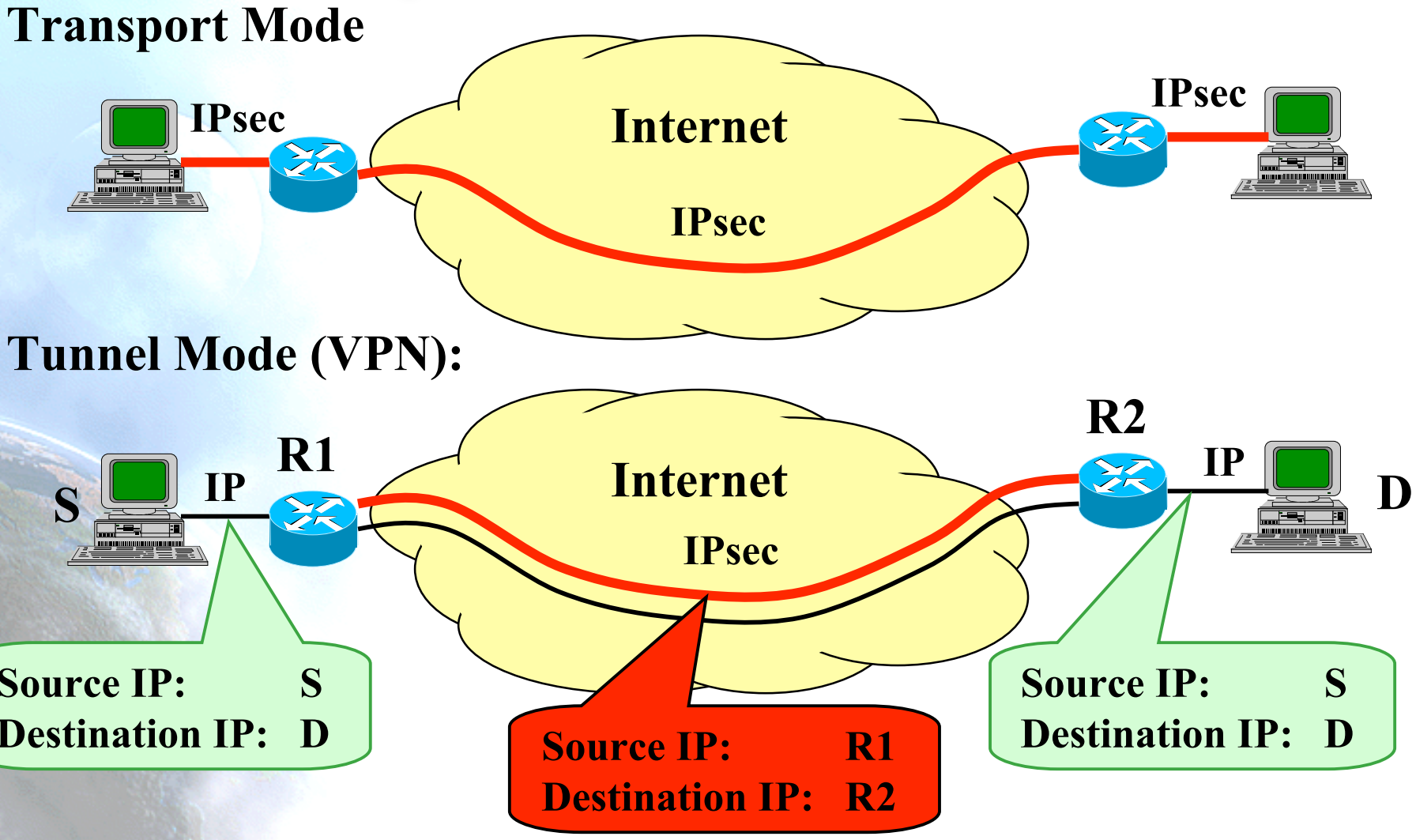


Algorithms

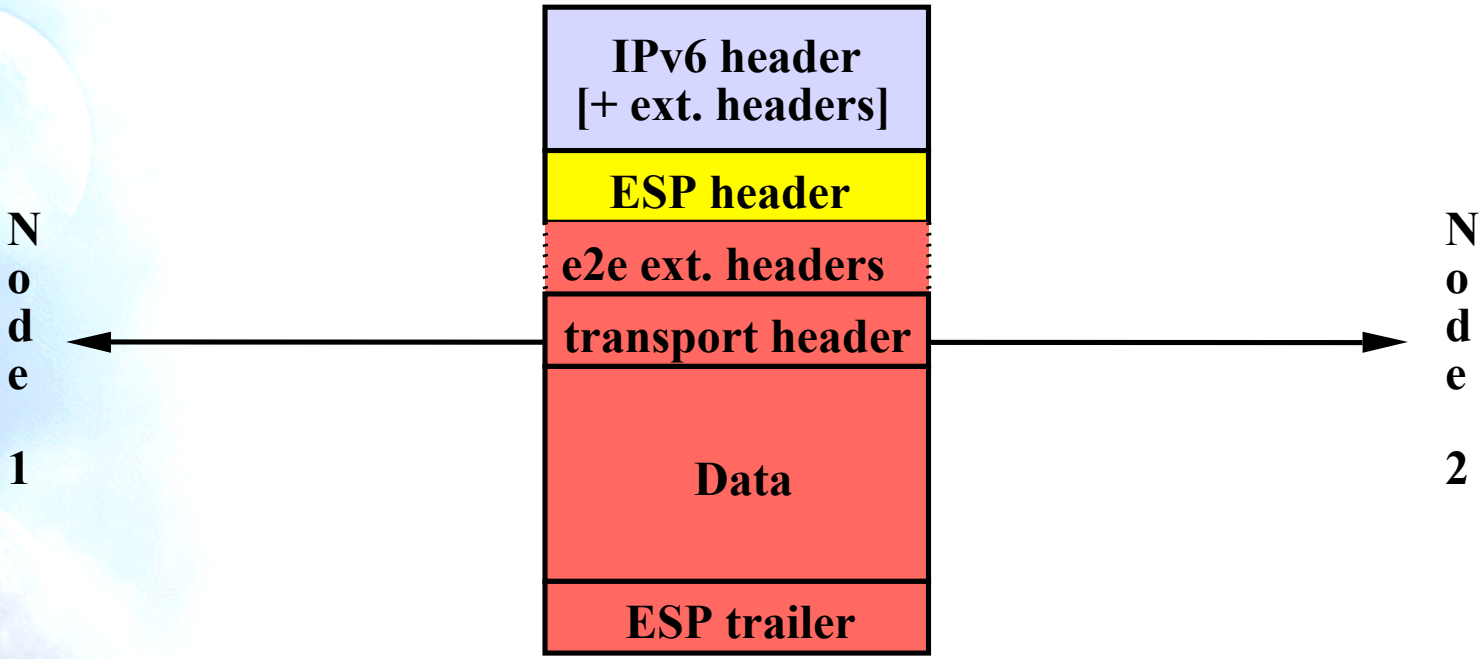
- Specified in the SA
- Encryption: Symmetric algorithms
- Interoperability support:
  - DES with CBC (encryption)
  - MD5 & SHA-1 (authentication)
- Others:
  - Triple DES, RC5, ...

4.4. Transport and Tunnel Modes

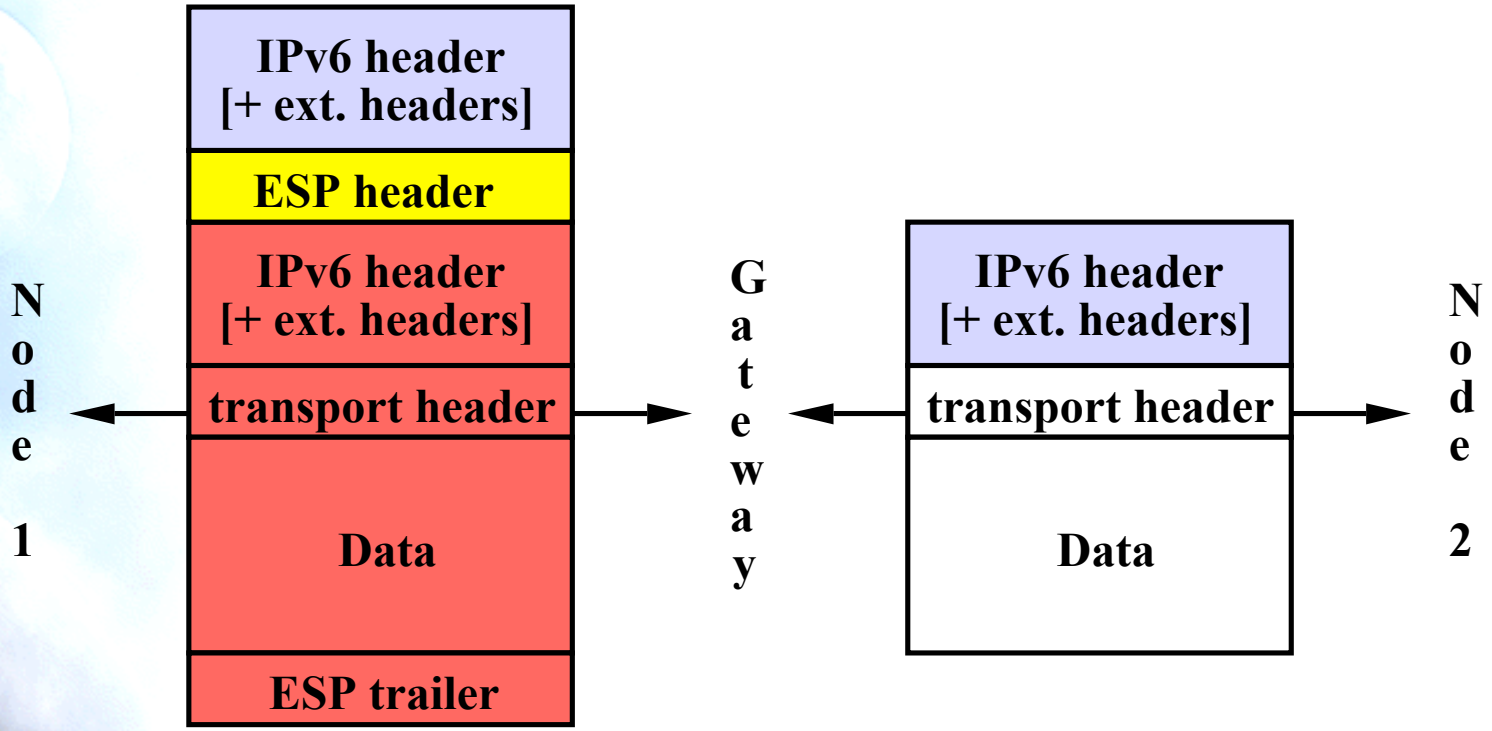
Transport vs. Tunnel Mode



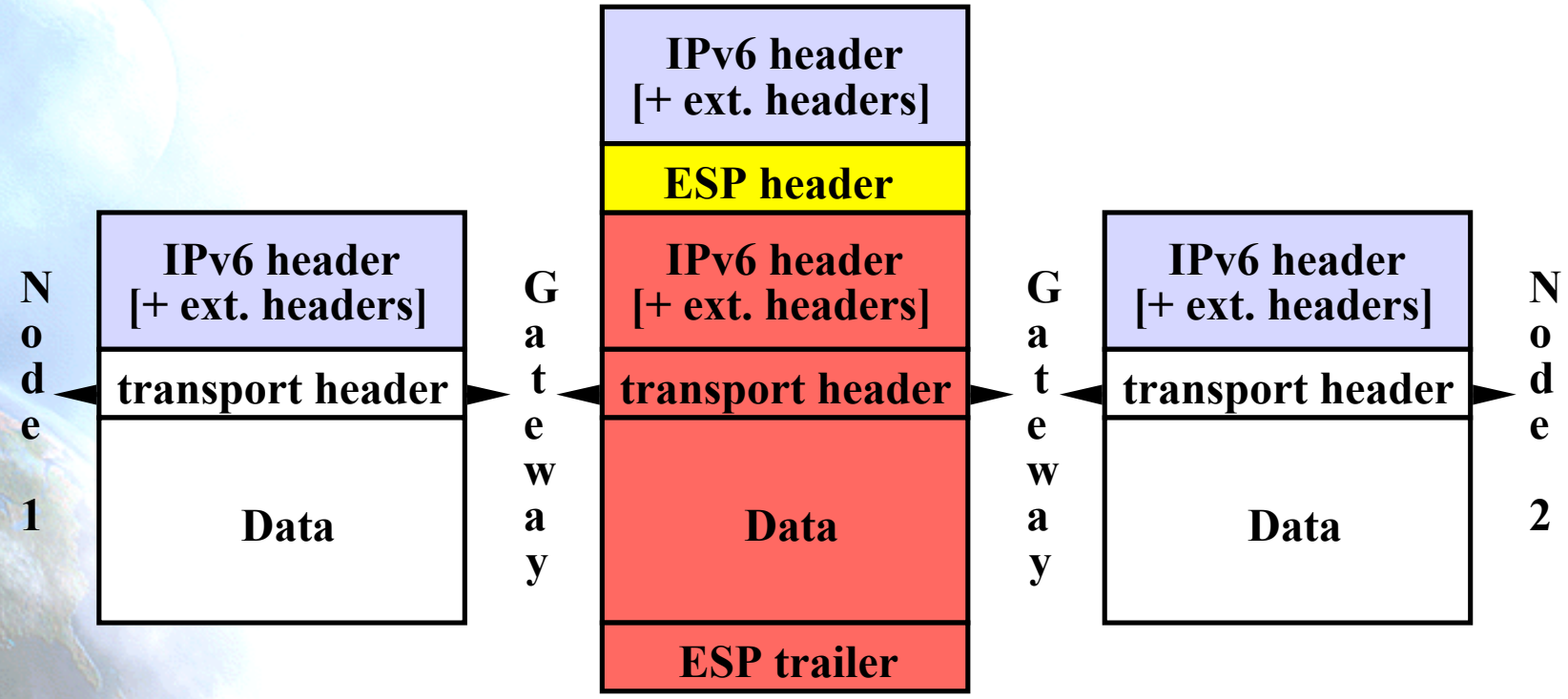
Transport Mode ESP End-to-End



Tunnel Mode ESP
End to Security Gateway



Tunnel Mode ESP
Gateway to Gateway



4.5. Key Management

Key Distribution

- Manual:
  - Simplest form of management.
  - Each system is configured with his own and others keys.
  - Practical in small, static environments.
  - Do not scale well.
- Automatic:
  - On-demand creation of SA´s.
  - The default is IKE - Internet Key Exchange (RFC2409).
  - Other automated SA management protocols MAY be employed.

IKE

- Standard Method to:
  - Dynamically authenticate IPsec peers
  - Negotiate security services
  - Generate shared keys
- Protocols:
  - ISAKMP (Internet Security Association and Key Management Protocol) defines the procedures for authenticating a communicating peer, creation and management of SA's, key generation techniques, and threat mitigation. (RFC2407-2408).
  - OAKLEY: Key exchange protocol (RFC2412).

IPv6 Tutorial

5. Quality of Service

Concept of QoS

- Quality: Reliable delivery of data ("better than normal")
  - Data loss
  - Latency
  - Jittering
  - Bandwidth
- Service: Anything offered to the user
  - Communication
  - Transport
  - Application

Abstract

- “Quality of Service is a measurement of the network behavior with respect to certain characteristics of defined services” !!!!!
- Common concepts to all definitions of QoS:
  - Traffic and type of service differentiation
  - Users may be able to treat one or more traffic classes differently

IP Quality of Service Approaches

Two basic approaches developed by IETF:

- "Integrated Service" (int-serv)
  - fine-grain (per-flow), quantitative promises (e.g., x bits per second), uses RSVP signalling
- "Differentiated Service" (diff-serv)
  - coarse-grain (per-class), qualitative promises (e.g., higher priority), no explicit signalling

IPv6 Support for Int-Serv

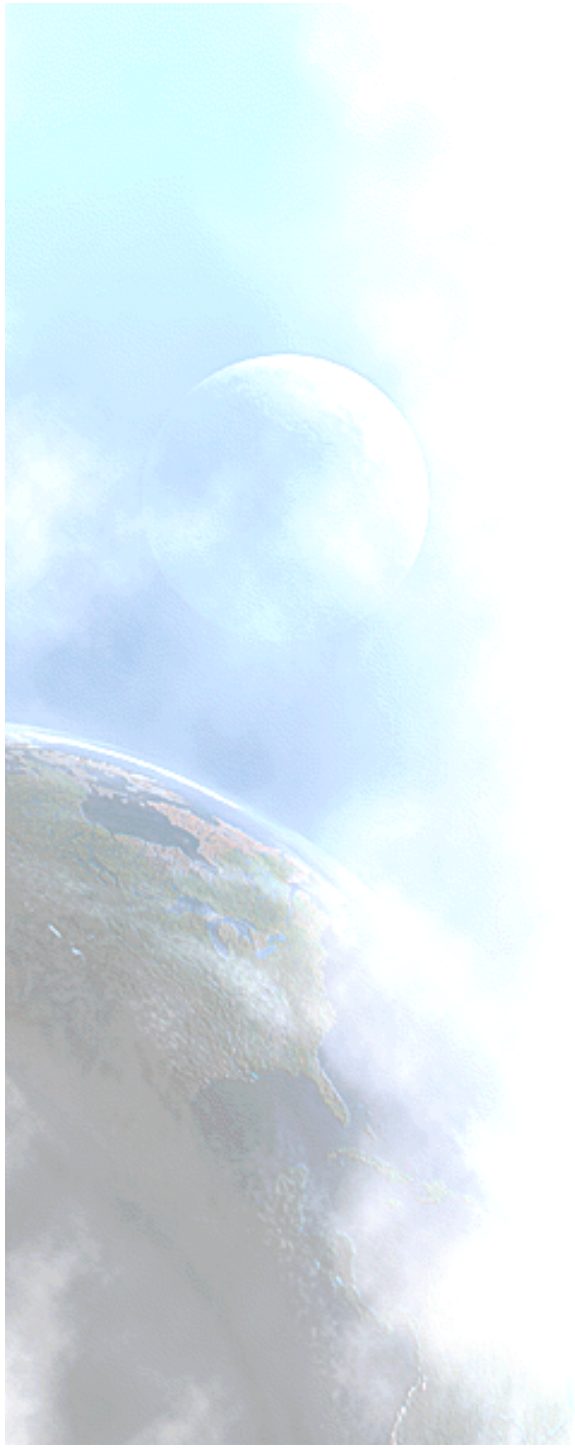
20-bit Flow Label field to identify specific flows needing special QoS

- each source chooses its own Flow Label values; routers use Source Addr + Flow Label to identify distinct flows
- Flow Label value of 0 used when no special QoS requested (the common case today)
- this part of IPv6 is not standardized yet, and may well change semantics in the future

IPv6 Support for Diff-Serv

8-bit Traffic Class field to identify specific classes of packets needing special QoS

- same as new definition of IPv4 Type-of-Service byte
- may be initialized by source or by router enroute; may be rewritten by routers enroute
- traffic Class value of 0 used when no special QoS requested (the common case today)



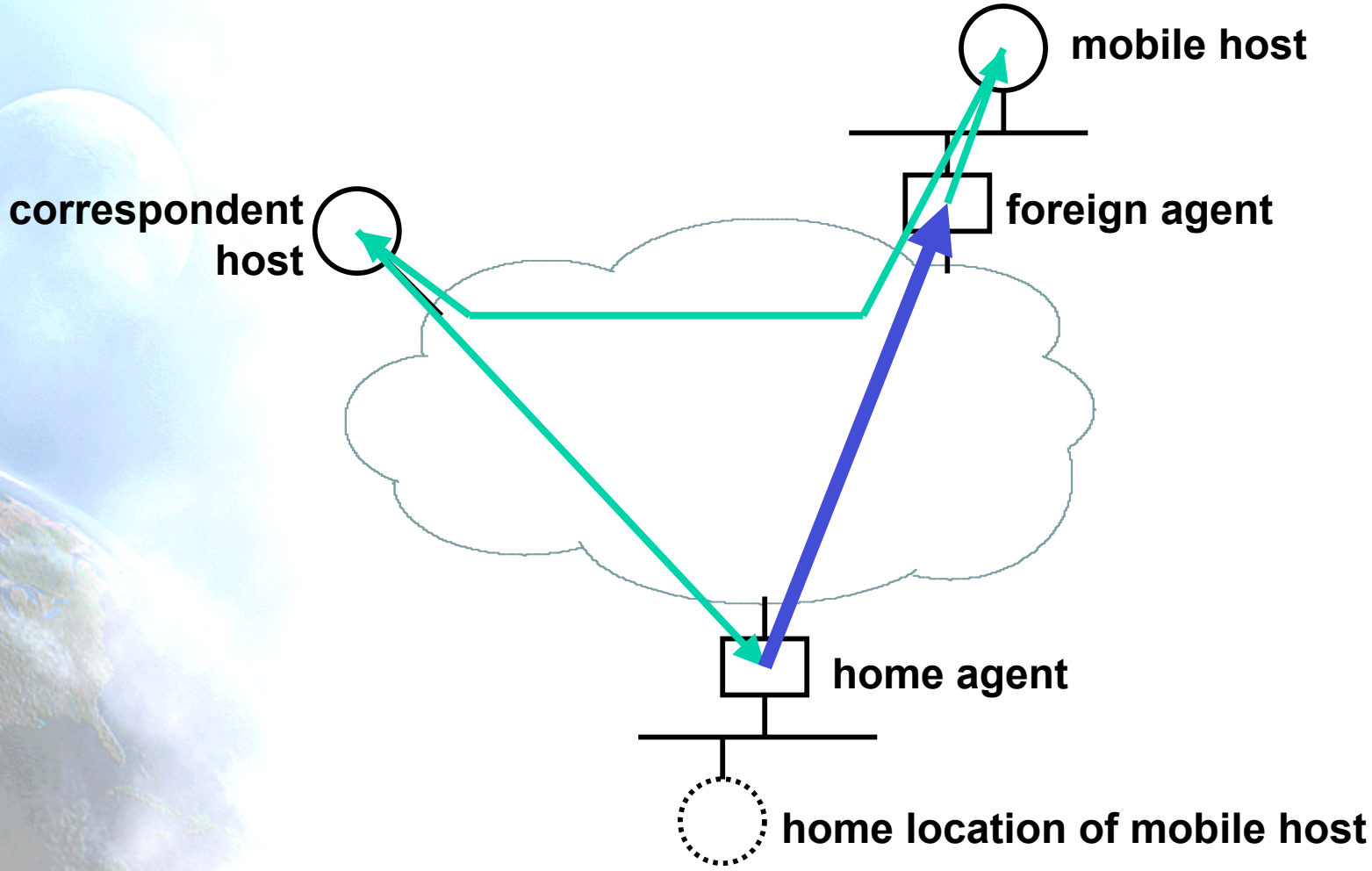
IPv6 Tutorial

6. Mobility

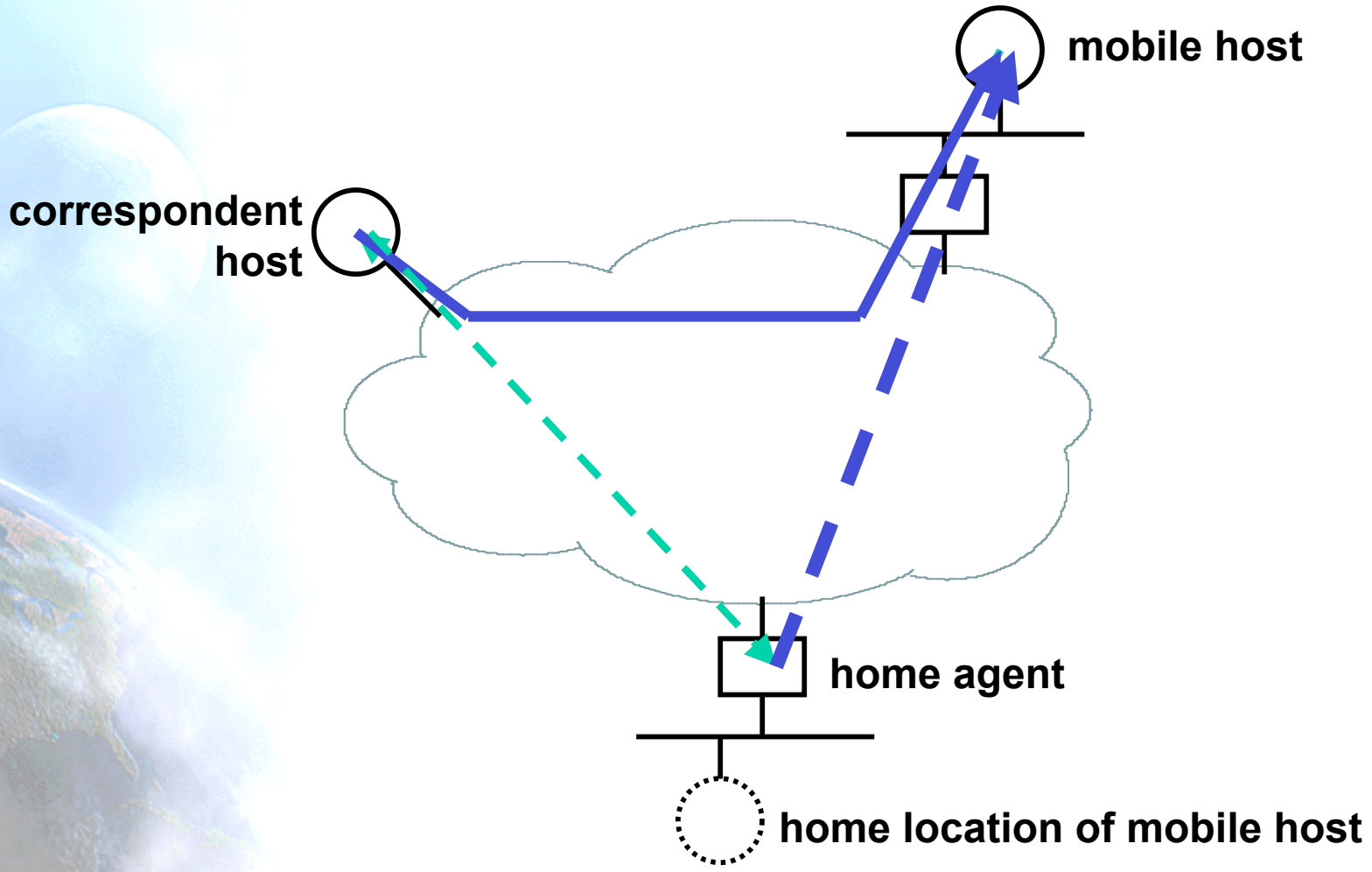
IPv6 Mobility

- 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)



Standards

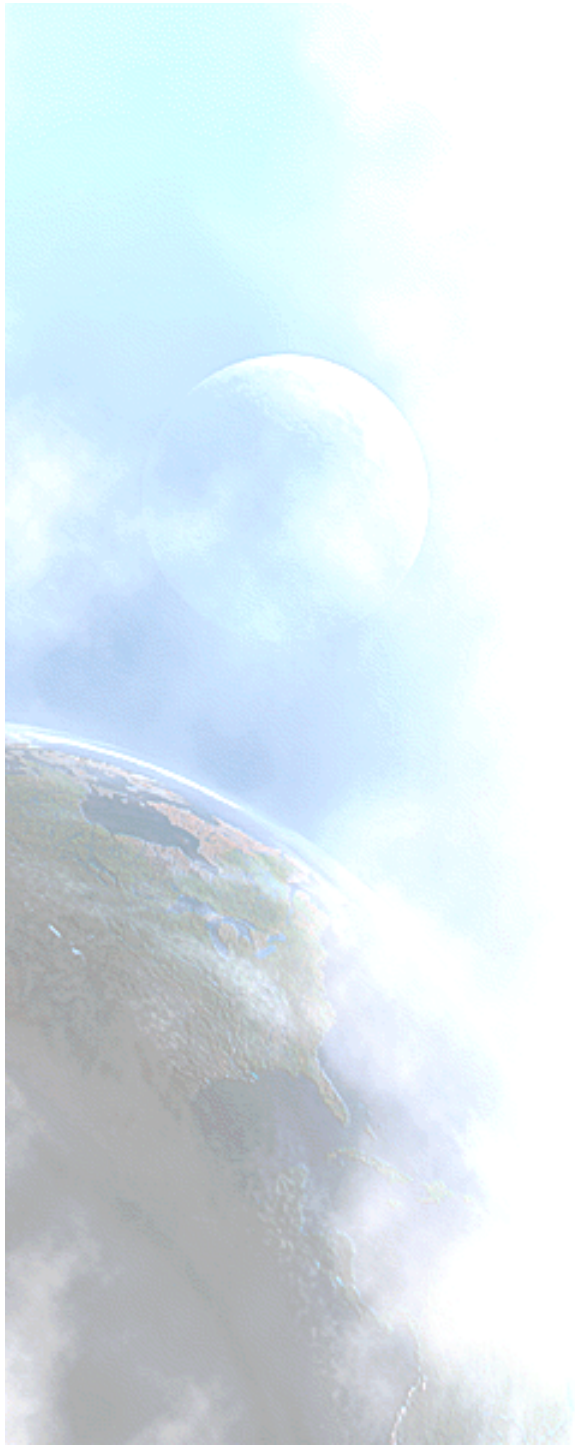
- Mobility Support in IPv6
  - RFC3775 – June 2004
- Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents
  - RFC3776 – June 2004

IPv6 Tutorial

7. ICMPv6 & Neighbor Discovery

Agenda

7.1. ICMPv6
7.2. Neighbor Discovery
7.3. Autoconfiguration
7.4. DHCPv6
7.5. Router Renumbering
7.6. Multi-Homing



7.1. ICMPv6

RFC2463

- IPv6 uses the Internet Control Message Protocol (ICMP) as defined for IPv4 (RFC792)
- Some changes for IPv6: ICMPv6.
- Next Header value = 58.
- ICMPv6 is used by IPv6 nodes to report errors encountered in processing packets, and to perform other internet-layer functions, such as diagnostics (ICMPv6 "ping").
- ICMPv6 is an integral part of IPv6 and MUST be fully implemented by every IPv6 node.

ICMPv6 Messages

- Grouped into two classes:
  - Error messages
  - Informational messages.

bits 8	16	32
Type	Code	Checksum
Message Body		

- Error messages have a zero in the high-order bit of their message Type field values (message Types from 0 to 127)
- Informational messages have message Types from 128 to 255

Message Source Address Determination

- A node that sends an ICMPv6 message has to determine both the Source and Destination IPv6 Addresses in the IPv6 header before calculating the checksum.
- If the node has more than one unicast address, it must choose the Source Address of the message as follows:

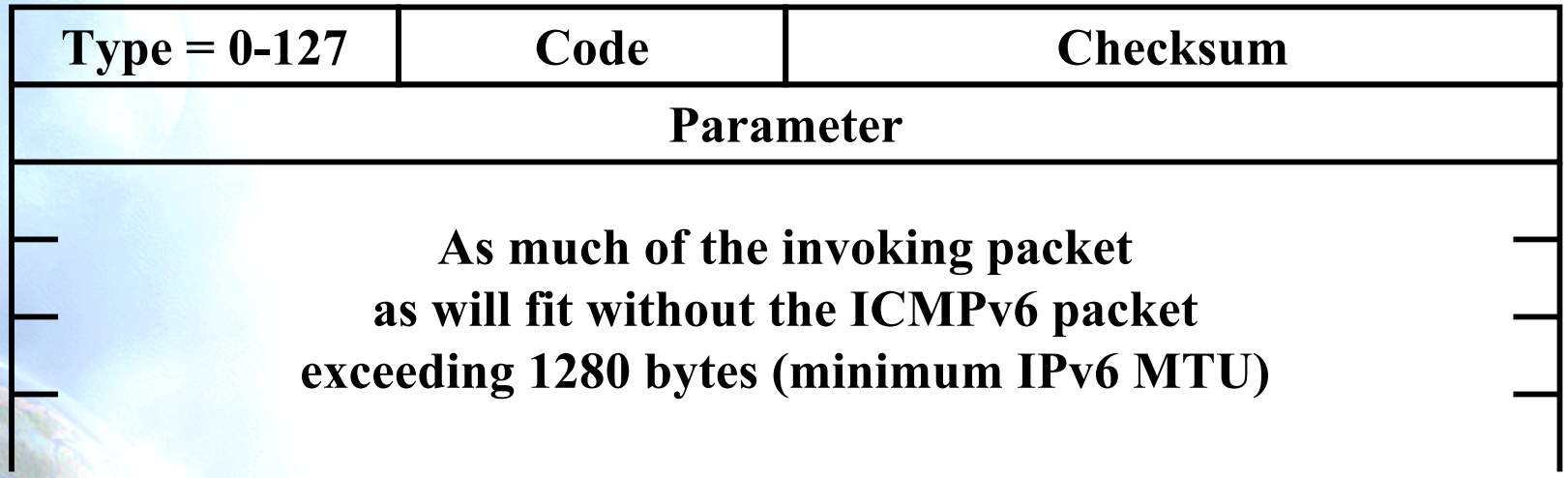
a) Message responding to a message sent to one of the node's unicast addresses, then Reply Source Address = Same Address.

b) Message responding to a message sent to a multicast or anycast group in which the node is a member, then Reply Source Address = unicast address belonging to the interface on which the multicast or anycast packet was received.

c) Message responding to a message sent to an address that does not belong to the node, then Source Address = unicast address belonging to the node that will be most helpful in diagnosing the error.

d) Otherwise, the node's routing table must be examined to determine which interface will be used to transmit the message to its destination, message Source Address = unicast address belonging to that interface.

ICMP Error Messages



ICMP Error Messages Types

- Destination Unreachable (type = 1, parameter = 0)
  - No route to destination (code = 0)
  - Communication with destination administratively prohibited (code = 1)
  - Not Assigned (code = 2)
  - Address Unreachable (code = 3)
  - Port Unreachable (code = 4)
- Packet Too Big (type = 2, code = 0, parameter = next hop MTU)
- Time Exceeded (type = 3, parameter = 0)
  - Hop Limit Exceeded in Transit (code = 0)
  - Fragment Reassembly Time Exceeded (code = 1)
- Parameter Problem (type = 4, parameter = offset to error)
  - Erroneous Header Field (code = 0)
  - Unrecognized Next Header Type (code = 1)
  - Unrecognized IPv6 Option (code = 2)

ICMP Informational Messages

- Echo Request (type =128, code = 0)
- Echo Reply (type = 129, code = 0)

Type = 128-255	Code	Checksum
Maximum Response Delay		Reserved
Multicast Address		

- Multicast listener discovery messages:
  - Query, report, done (like IGMP for IPv4):

7.2. Neighbor Discovery

RFC2461

- Defines the Neighbor Discovery (ND) protocol for IPv6.
- Nodes (hosts and routers) use Neighbor Discovery to determine the link-layer addresses for neighbors known to reside on attached links and to quickly purge cached values that become invalid.
- Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf.
- Nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed link-layer addresses.

Autoconfiguration Foundation

- ND is a very complete and sophisticated foundation to enable the autoconfiguration mechanism in IPv6.
- Enable extended support for proxy services, anycast addresses, load sharing balancing, among others.
- RFC2461 describes a conceptual model of one possible data structure organization that hosts (and to some extent routers) will maintain in interacting with neighboring nodes.

Interaction Between Nodes

- Defines mechanism to solve:
  - Router Discovery.
  - Prefix Discovery.
  - Parameter Discovery.
  - Address Autoconfiguration.
  - Address Resolution.
  - Next-hop Determination.
  - Neighbor Unreachability Detection (NUD).
  - Duplicate Address Detection (DAD).
  - First-Hop Redirect.

New ICMP Packet Types

- ND defines 5 packet types:
  - Router Solicitation.
  - Router Advertisement.
  - Neighbor Solicitation.
  - Neighbor Advertisement.
  - Redirect.

Router Advertisements

- On multicast-capable links, each router periodically multicasts a Router Advertisement packet.
- A host receives Router Advertisements from all routers, building a list of default routers.
- A separate Neighbor Unreachability Detection algorithm provides failure detection.
- Router Advertisements contain a list of prefixes used for on-link determination and/or autonomous address configuration.
- Router Advertisements allow routers to inform hosts how to perform Address Autoconfiguration.

Router Advertisement "Session"

"I am a router" (implied) list of:

lifetime as default (1 sec - 18,2 hr) » prefix

"get addresses from DHCP" flag » prefix length

"get other stuff from DHCP" flag » valid lifetime

router's link-layer address » preferred lifetime

link MTU » on-link flag

suggested hop limit » autoconfig OK flag

Router Advertisement Format

Bits 8			16	32
Type = 134			Code = 0	Checksum
Cur Hop Limit	M	O	Reserved = 0	Router Lifetime
Reachable Time				
Retrans Timer				
Options …				

- Cur Hop Limit: default value that should be placed in the Hop Count field of the IP header for outgoing IP packets.
- M: 1-bit "Managed address configuration" flag.
- O: 1-bit "Other stateful configuration" flag.
- Router Lifetime: 16-bit unsigned integer.
- Reachable Time 32-bit unsigned integer.
- Retrans Timer 32-bit unsigned integer.
- Possible Options: Source link-layer address, MTU, …

Router Solicitation

- At Start-up, hosts send Router Solicitations in order to prompt routers to generate Router Advertisements quickly.
- Sent to all routers multicast address (link scope).

Bits 8	16	32
Type = 133	**Code = 0**	**Checksum**
Reserved = 0		
Options …		

- Possible Options: Source link-layer address.

Neighbor Solicitation

- Nodes send Neighbor Solicitations to request the link-layer address of a target node while also providing their own link-layer address to the target.
- Neighbor Solicitations are multicast when the node needs to resolve an address and unicast when the node seeks to verify the reachability of a neighbor.

Bits 8	16	32
Type = 135	**Code = 0**	**Checksum**
Reserved = 0		
Target Address		
Options …		

- Target Address: IP address of the target of the solicitation. It MUST NOT be a multicast address.
- Possible Options: Source link-layer address.

Neighbor Advertisement

- A node sends Neighbor Advertisements in response to Neighbor Solicitations and sends unsolicited Neighbor Advertisements in order to (unreliably) propagate new information quickly.

Bits 8	16	32
Type = 136	Code = 0	Checksum
R S O	Reserved = 0	
Target Address		
Options …		

- Target Address: For solicited advertisements, the Target Address field in the Neighbor Solicitation message that prompted this advertisement. For an unsolicited advertisement, the address whose link-layer address has changed. The Target Address MUST NOT be a multicast address.

Redirect

- Routers send Redirect packets to inform a host of a better first-hop node on the path to a destination.
- Hosts can be redirected to a better first-hop router but can also be informed by a redirect that the destination is in fact a neighbor.

Bits 8	16	32
Type = 137	**Code = 0**	**Checksum**
Reserved = 0		
Target Address		
Destination Address		
Options …		

- Target Address: IP address that is a better first hop to use for the ICMP Destination Address.
- Destination Address: IP address of the destination which is redirected to the target.

7.3. Autoconfiguration

RFC2462

- The document specifies the steps a host takes in deciding how to autoconfigure its interfaces in IPv6.
- The autoconfiguration process includes creating a link-local address and verifying its uniqueness on a link, determining what information should be autoconfigured (addresses, other information, or both), and in the case of addresses, whether they should be obtained through the stateless mechanism, the stateful mechanism, or both.
- IPv6 defines both a stateful and stateless address autoconfiguration mechanism.
- Stateless autoconfiguration requires no manual configuration of hosts, minimal (if any) configuration of routers, and no additional servers.

Stateless or Serverless Autoconfiguration

- Stateless mechanism allows a host to generate its own addresses using a combination of locally available information and information advertised by routers.
- Routers advertise prefixes that identify the subnet(s) associated with a link.
- Hosts generate an "interface identifier" that uniquely identifies an interface on a subnet, locally generated, e.g., using MAC address.
- An address is formed by combining the both.
- In the absence of routers, a host can only generate link-local addresses.
- Link-local addresses are sufficient for allowing communication among nodes attached to the same link.

Stateful Autoconfiguration

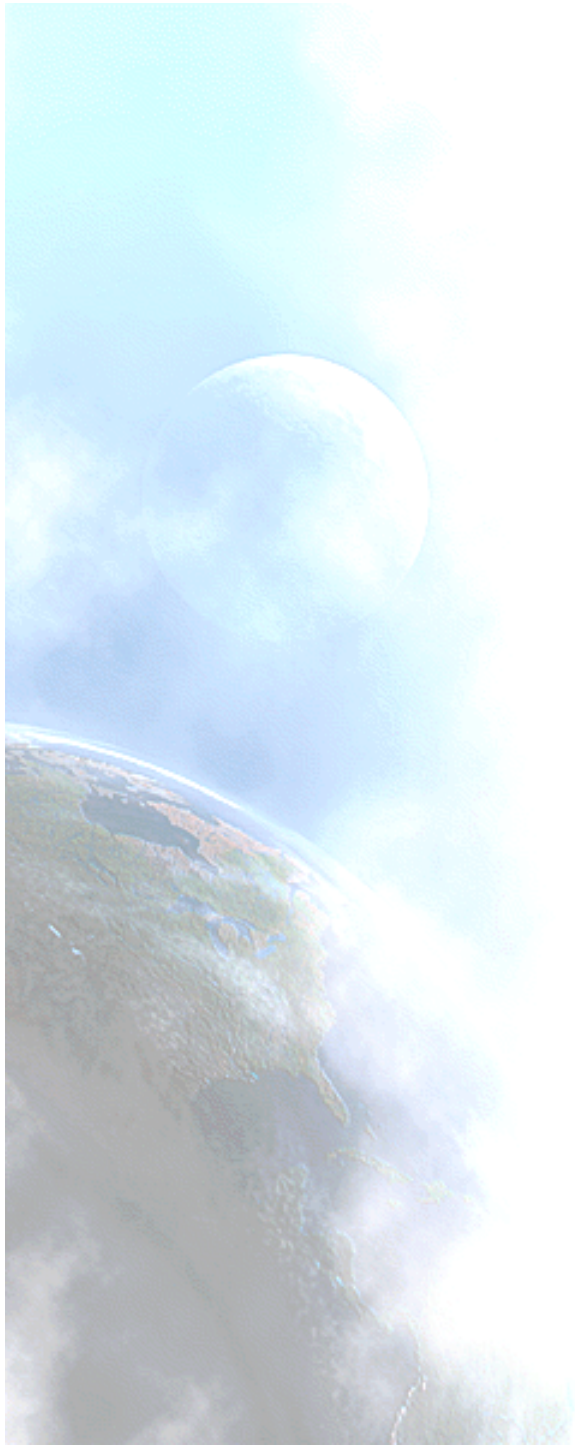
- Hosts obtain interface addresses and/or configuration information and parameters from a server.
- Servers maintain a database that keeps track of which addresses have been assigned to which hosts.
- Stateless and stateful autoconfiguration complement each other.
- Both stateful and stateless address autoconfiguration may be used simultaneously.
- The site administrator specifies which type of autoconfiguration to use through the setting of appropriate fields in Router Advertisement messages.

Address Life Time

- IPv6 addresses are leased to an interface for a fixed (possibly infinite) length of time, that indicates how long the address is bound to an interface.
- When a lifetime expires, the binding (and address) become invalid and the address may be reassigned to another interface elsewhere in the Internet.
- To handle the expiration of address bindings gracefully, an address goes through two distinct phases while assigned to an interface.
  - Initially, an address is "preferred", meaning that its use in arbitrary communication is unrestricted.
  - Later, an address becomes "deprecated" in anticipation that its current interface binding will become invalid.

Duplicate Address Detection

- To insure that all configured addresses are likely to be unique on a given link, nodes run a "duplicate address detection" algorithm on addresses before assigning them to an interface.
- The Duplicate Address Detection algorithm is performed on all addresses, independent of whether they are obtained via stateless or stateful autoconfiguration.
- The procedure for detecting duplicate addresses uses Neighbor Solicitation and Advertisement messages.
- Since host autoconfiguration uses information advertised by routers, routers will need to be configured by some other means. However, it is expected that routers will generate link-local addresses using the same mechanism.
- Routers are expected to successfully pass the Duplicate Address Detection procedure on all addresses prior to assigning them to an interface.



7.4. DHCPv6

draft-ietf-dhc-dhcpv6-17

- DHCP for IPv6 (DHCPv6) is an UDP client/server protocol designed to reduce the cost of management of IPv6 nodes in environments where network managers require more control over the allocation of IPv6 addresses and configuration of network stack parameters than that offered by "IPv6 Stateless Autoconfiguration".
- DHCP reduces the cost of ownership by centralizing the management of network resources rather than distributing such information in local configuration files among each network node.
- DHCP is designed to be easily extended to carry new configuration parameters through the addition of new DHCP "options" defined to carry this information.

DHCPv6 Goals

- Is a mechanism rather than a policy.
- Is compatible with IPv6 stateless autoconfiguration.
- Does not require manual configuration of network parameters on DHCP clients.
- Does not require a server on each link.
- Coexists with statically configured, non-participating nodes and with existing network protocol implementations.
- DHCP clients can operate on a link without IPv6 routers present.
- DHCP will provide the ability to renumber network(s).
- A DHCP client can make multiple, different requests.
- DHCP will contain the appropriate time out and retransmission mechanisms to efficiently operate in environments with high latency and low bandwidth characteristics.

New User Features with DHCPv6

- Configuration of Dynamic Updates to DNS.
- Address deprecation, for dynamic renumbering.
- Relays can be preconfigured with server addresses, or use of multicast.
- Authentication.
- Clients can ask for multiple IP addresses.
- Addresses can be reclaimed using the Reconfigure-init message.
- Integration between stateless and stateful address autoconfiguration.
- Enabling relays to locate off-link servers.

7.5. Router Renumbering

RFC2894

- IPv6 Neighbor Discovery and Address Autoconfiguration make initial assignments of address prefixes to hosts.
- These two mechanisms also simplify the reconfiguration of hosts when the set of valid prefixes changes.
- The Router Renumbering ("RR") mechanism allows address prefixes on routers to be configured and reconfigured almost as easily as the combination of Neighbor Discovery and Address Autoconfiguration works for hosts.
- Provides a means for a network manager to make updates to the prefixes used by and advertised by IPv6 routers throughout a site.

Functional Overview

- Router Renumbering Command packets contain a sequence of Prefix Control Operations (PCOs).
- Each PCO specifies an operation, a Match-Prefix, and zero or more Use-Prefixes.
- A router processes each PCO, checking each of its interfaces for an address or prefix which matches the Match-Prefix.
- Applied for every interface on which a match is found.
- The operation is one of ADD, CHANGE, or SET-GLOBAL to instruct the router to respectively add the Use-Prefixes to the set of configured prefixes, remove the prefix which matched the Match-Prefix and replace it with the Use-Prefixes, or replace all global-scope prefixes with the Use-Prefixes.

7.6. Multi-Homing

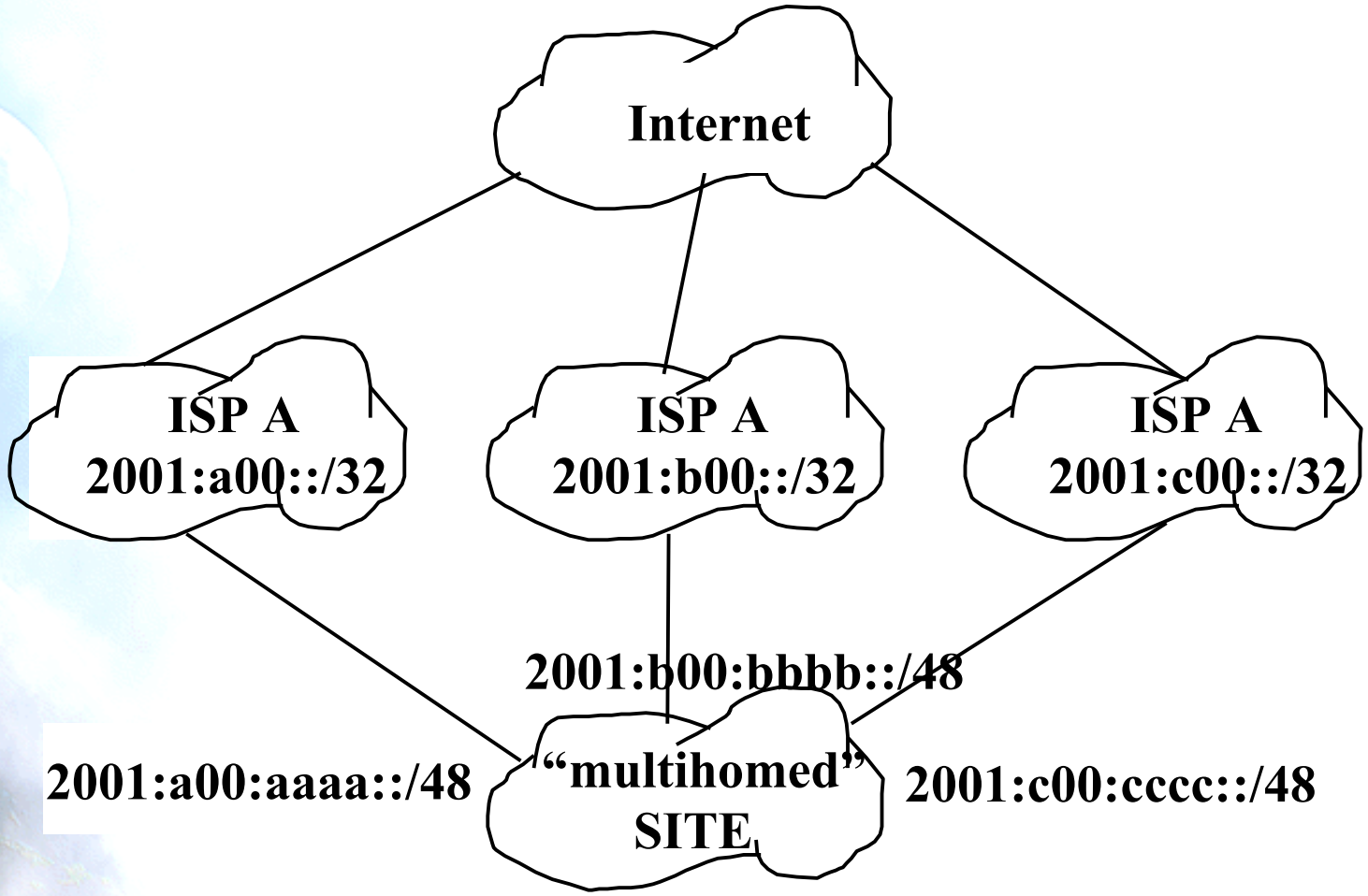
Motivations

- Connectivity to more than one Internet Service Providers (ISPs) is becoming relevant, for the purpose of redundancy and traffic load distribution.
- Large numbers of Multi-homed sites impose a direct challenge on routing aggregation and consequently on global Internet routing scalability.
- IPv6 multihoming also:
  - Provides redundancy and load sharing for the multi-homed sites
  - Facilitate the scalability of the global IPv6 Internet routing table
  - Is simple and operationally manageable.
- Uses existing routing protocol and implementation thus no new protocol or changes are needed.

Multihoming Mechanism

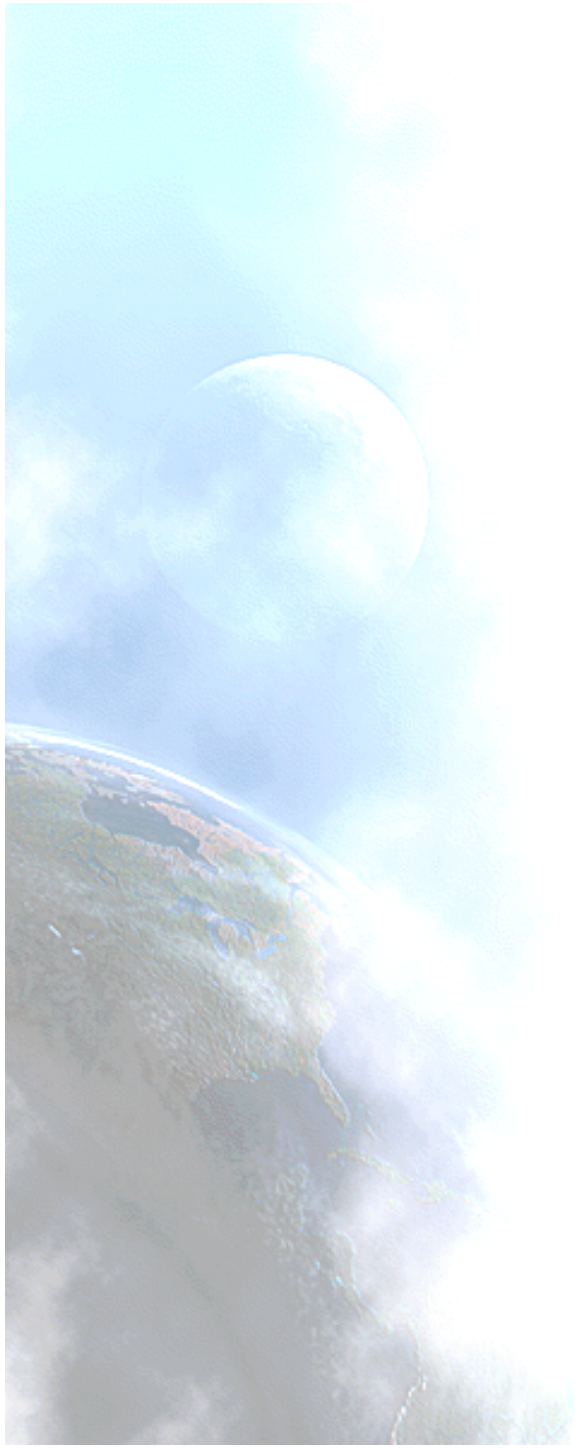
- Two types of multihoming connections:
  1. Site multi-homed to a single ISP, commonly at different geography locations
  2. Site multi-homed to more than one ISPs.
- The specific routes associated with the multihomed site 1 will not be visible outside of the particular ISP network and thus there is no real impact on the global routing: No special mechanism is needed for multihoming in this scenario.
- To obtain IP addresses, a multi-homed site will designate one of its ISPs as its primary ISP and receive IP address assignment from the primary ISP's IPv6 aggregation block.

Multi-Homing Example



Current Status

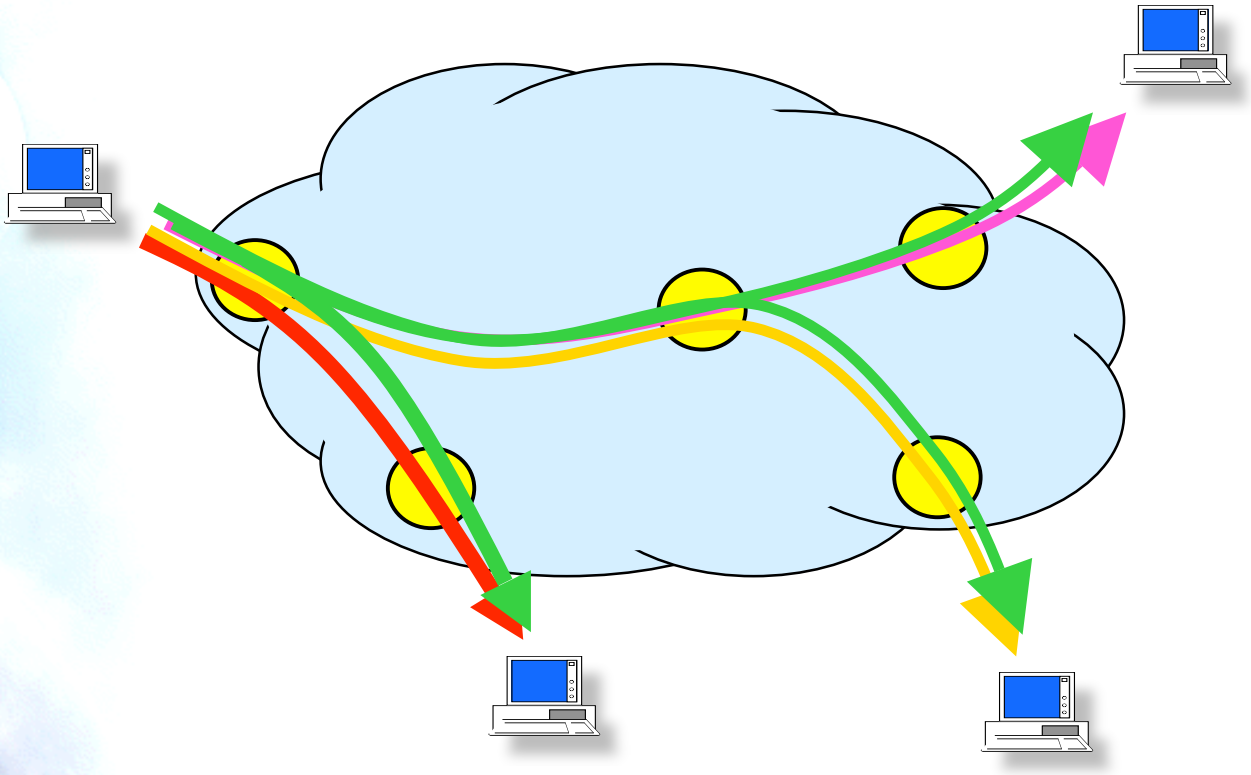
- Available IPv6 multi-homing solution:
  - RFC2260 (same as IPv4): Tunnels, but limited fault tolerance, not provides load sharing, performance and policying.
- IETF multi6 Working Group:
  - Work in Progress (IPv6 site multihoming requirements).



IPv6 Tutorial

8. Multicast

What's Multicast?



Applications

- Distributed systems
- Video on Demand (VoD)
- Radio/TV Diffusion
- Multipoint Conferencing (voice/video)
- Network Gaming
- Network level functions

How it Works ?

- The host joins/signoff the multicast group
- No restriction about number of groups or members per group
- Sending to the group don't means belonging to it
- The destination address is a group address (multicast address)
- Connection-Less service

IPv4 vs. IPv6

- IPv4
  - Broadcast
    - Limited: 255.255.255.255
    - Directed: <network>11..1
  - Multicast
    - D Class:

      224.0.0.0 -239.255.255.255
- IPv6
  - Multicast

Reserved Multicast Addresses (I)

- Node-Local Scope
  - FF01::1 All Nodes Address
  - FF01::2 All Routers Address

- Link-Local Scope
  - FF02::1 All Nodes Address
  - FF02::2 All Routers Address
  - FF02::4 DVMRP Routers
  - FF02::5 OSPFIGP
  - FF02::6 OSPFIGP Designated Routers
  - FF02::9 RIP Routers
  - FF02::B Mobile-Agents
  - FF02::D All PIM Routers
  - FF02::1:2 All-DHCP-agents
  - FF02::1:FFXX:XXXX Solicited-Node Address

Reserved Multicast Addresses (II)

- Site-Local Scope
  - FF05::2 All Routers Address
  - FF05::1:3 All-DHCP-servers
  - FF05::1:4 All-DHCP-relays

- Variable Scope Multicast Addresses
  - FF0X::101 Network Time Protocol (NTP)
  - FF0X::129 Gatekeeper
  - FF0X::2:0000-FF0X::2:7FFD Multimedia Conference Calls
  - FF0X::2:7FFE SAPv1 Announcements
  - FF0X::2:8000-FF0X::2:FFFF SAP Dynamic Assignments

Important Multicast Addresses

- FF01::1, FF02::1 All-nodes
- FF01::2, FF02::2, FF05::2 All routers

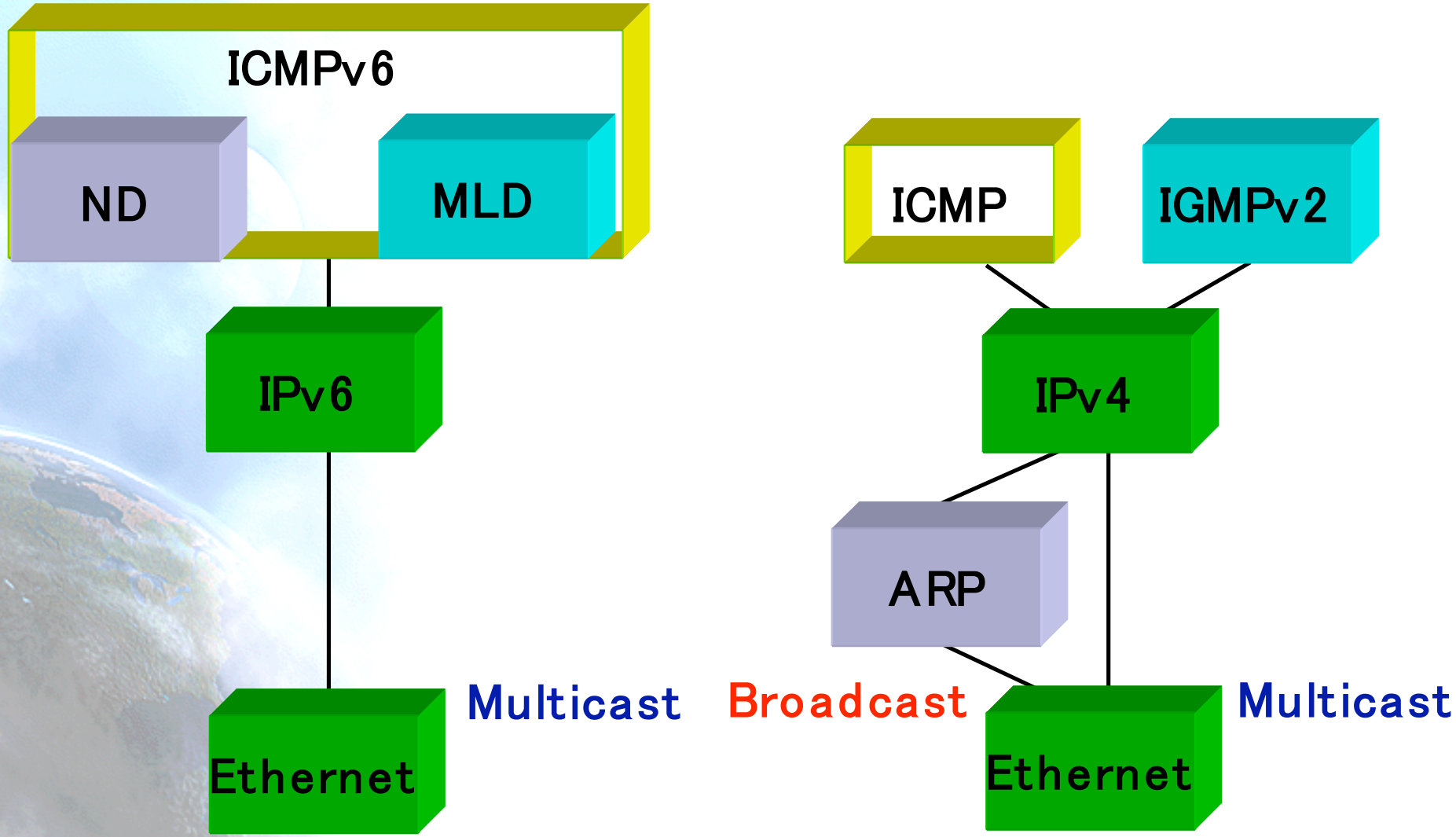
- Solicited Node (SN) address from a unicast one
  - For the address that finish with "XY:ZTUV"
  - the SN is FF02::1:FFXY:ZTUV

- Every IPv6 node must join SN for all its unicast and anycast addresses, and to "all-nodes"

Multicast Listener Discovery

- MLD (RFC2710) enables each IPv6 router to learn which multicast addresses have listeners on each of its directly attached links
- This is a mandatory function in IPv6 nodes
- Is used instead of IGMP

Control Plane IPv4 vs. IPv6



Multicast Routing

- Routers listen all the groups
- Multicast Routing Protocols:
- Dense Mode:
  - DVMRP
  - PIM-DM
  - MOSPF
- Sparse Mode:
  - CBT
  - PIM-SM

- Allow multicast tunnels over IPv6 unicast networks

IPv6 Tutorial

9. 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 or A6 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

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

IPv6 Tutorial

10. Porting Applications to IPv6

The Porting Issue

- Network layer change is not transparent
  - IPv4 applications need to be modified for IPv6
- Best practice is to turn IPv4 apps into protocol-independent apps
- Usually not difficult
  - Simple apps (e.g. telnet) take only hours to port

Main Changes From IPv4

- Address Size
  - 32 bits (IPv4) to 128 bits (IPv6)
- API changes
  - Address size issues
  - Protocol independence
- Dependencies on IP header size
- Dependencies on particular addresses

Not All Applications Need to be Changed

- Many applications don't talk to the network directly, but rather use library functions to carry out those tasks. In some cases, only the underlining library needs to be changed.
- Examples:
  - RPC
  - DirectPlay

Address Storage Issues

- Problem: you can't store a 128 bit value in a 32 bit space.
- Most applications today store and reference IP addresses as either:
  - sockaddrs (good)
  - in_addrs (okay)
  - ints (bad)
- Storage versus reference

Anatomy of a sockaddr

```
struct sockaddr {
    u_short sa_family;  // Address family
    char sa_data[14];  // Address data
};
```

- The sa_family field contains a value which indicates which type of address this is (IPv4, IPv6, etc).

sockaddr_in

```
struct sockaddr_in {
    short sin_family;
    u_short sin_port;
    struct in_addr sin_addr;
    char sin_zero[8];
};
```

sockaddr_in6

```
struct sockaddr_in6 {
    short sin6_family; // AF_INET6
    u_short sin6_port;
    u_long sin6_flowinfo;
    struct in_addr6 sin6_addr;
    u_long sin6_scope_id;
};
```

API Changes

- Most of the socket APIs don't need to change – they were originally designed to be protocol independent, and thus take pointers to sockaddrs as input or output.
  - bind, connect, getsockname, getpeername, etc.
- The name resolution APIs are the big offenders that need to be changed
  - gethostbyname, gethostbyaddr.

New Name Resolution APIs

- getaddrinfo – for finding the addresses and/or port numbers that corresponds to a given host name.and service.
- getnameinfo – for finding the host name and/or service name that corresponds to a given address or port number.
- Both of these APIs are protocol-independent – they work for both IPv4 and IPv6

Getaddrinfo

```
int
getaddrinfo(
    IN const char FAR * nodename,
    IN const char FAR * servicename,
    IN const struct addrinfo FAR * hints,
    OUT struct addrinfo FAR * FAR * res
);
```

Anatomy of Addrinfo

```
typedef struct addrinfo {
    int ai_flags;
    int ai_family;   // PF_xxx.
    int ai_socktype;   // SOCK_xxx.
    int ai_protocol;   // IPPROTO_xxx.
    size_t ai_addrlen;
    char *ai_canonname;
    struct sockaddr *ai_addr;
    struct addrinfo *ai_next;
} ADDRINFO, FAR * LPADDRINFO;
```

Getnameinfo

```
int
getnameinfo(
        IN  const struct sockaddr FAR * sa,
        IN  socklen_t salen,
        OUT char FAR * host,
        IN  DWORD hostlen,
        OUT char FAR * service,
        IN  DWORD servlen,
        IN  int flags
);
```

Header Size Dependencies

- Problem: The IPv6 header is 20 bytes larger than (the minimal) IPv4 header.
- Programs that calculate their datagram payload size by computing MTU – (UDP header size + IP header size) need to know that the IP header size has changed.

IPv4 Address Dependencies

- Some programs “know” certain addresses (e.g. loopback = IPv4 address 127.0.0.1).
- Programs whose purpose is to manipulate addresses (e.g. Network Address Translators, or NATs) obviously have innate knowledge of IPv4 addresses.
- Only an issue for those sorts of programs.
- NATs are evil anyway, so who cares.

IPv6 Tutorial

11. Current Status

Standards

- Core IPv6 specifications are IETF Draft Standards => well-tested & stable
  - IPv6 base spec, ICMPv6, Neighbor Discovery, PMTU Discovery, IPv6-over-Ethernet, IPv6-over-PPP,...
- Other important specs are further behind on the standards track, but in good shape
  - mobile IPv6, header compression, A6 DNS support,...
  - for up-to-date status: playground.sun.com/ipng
- UMTS R5 cellular wireless standards mandate IPv6

Implementations

- Most IP stack vendors have an implementation at some stage of completeness
  - some are shipping supported product today, e.g., 3Com, *BSD(KAME), Cisco, Epilogue, Ericsson Telebit, IBM, Hitachi, NEC, Nortel, Sun, Trumpet, Linux, Microsoft, HP, Juniper, Apple
  - others have beta releases now, supported products soon
  - others rumored to be implementing, but status unkown (to me), e.g., Bull, Mentat, Novell, SGI
  - (see playground.sun.com/ipng for most recent status reports)
- good attendance at frequent testing events

Deployment

- Experimental infrastructure: the 6bone
  - for testing and debugging IPv6 protocols and operations (see www.6bone.net)
- European IST projects
  - 6NET & Euro6IX
- Production infrastructure in support of education and research: the 6ren
  - CAIRN, Canarie, CERNET, Chunahwa Telecom, Dante, ESnet, Internet 2, IPFNET, NTT, Renater, Singren, Sprint, SURFnet, vBNS, WIDE
    (see www.6ren.net, www.6tap.net)
- Commercial infrastructure
  - a few ISPs (IIJ, NTT, SURFnet, Trumpet, Zama,…) have announced commercial IPv6 service or service trials

Deployment (cont.)

- IPv6 address allocation
  - 6bone procedure for test address space
  - regional IP address registries (APNIC, ARIN, RIPE-NCC) for production address space
- Deployment advocacy (a.k.a. marketing)
  - IPv6 Forum: www.ipv6forum.com

Much Still To Do

Though IPv6 today has all the functional capability of IPv4,

- Implementations are not as advanced
  (e.g., with respect to performance, multicast support, compactness, instrumentation, etc.)
- Deployment has only just begun
- Much work to be done moving application, middleware, and management software to IPv6
- Much training work to be done
  (application developers, network administrators, sales staff,…)
- Many of the advanced features of IPv6 still need specification, implementation, and deployment work

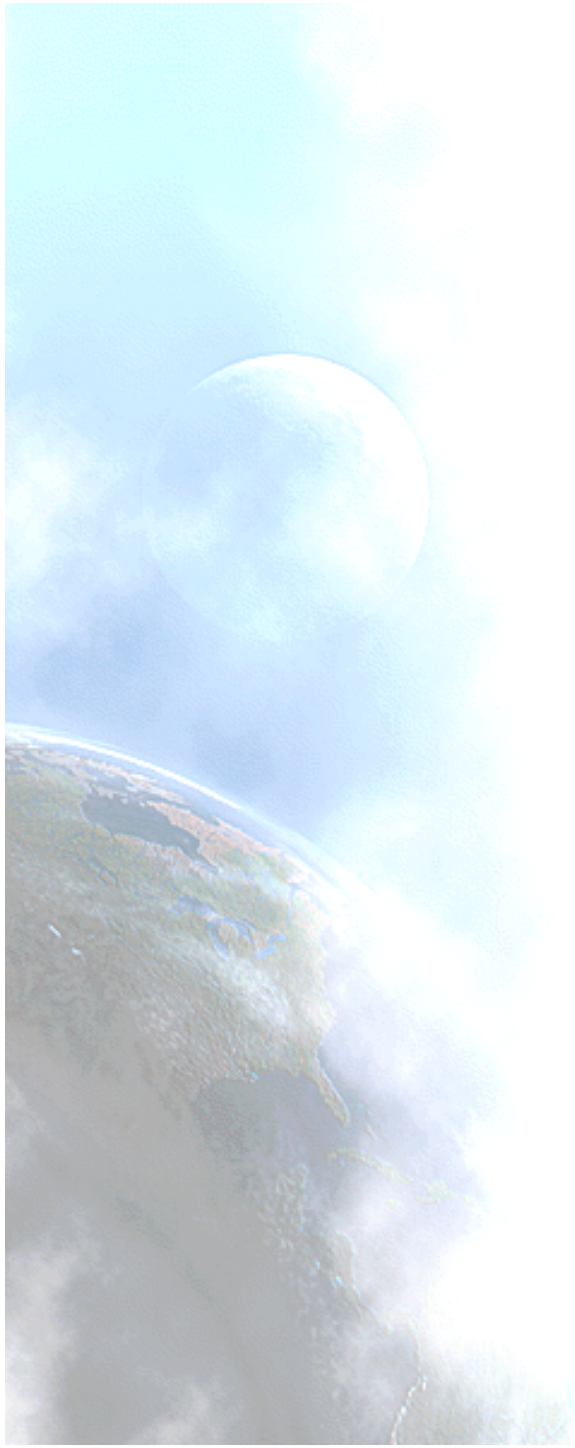
IPv6 Advanced Features

- Plug-and-play
  - we have most of the pieces for IP and DNS layers; still need work on auto-configuration of applications and services
- Mobility
  - to get most efficient routing in all cases, need to deploy key distribution infrastructure
- Security
  - though IPv6 enables end-to-end use of IPsec protocols (because it eliminates NATs), also dependant on key distribution infrastructure
- Quality of Service
  - IPv6 QoS features are same as IPv4's, but less widely implemented

Recent IPv6 “Hot Topics” in IETF

- multihoming / address selection
- address allocation
- DNS discovery
- anycast addressing
- scoped address architecture
- flow-label semantics
- API issues
  - (flow label, traffic class, PMTU discovery, scoping,…)
- enhanced router-to-host info
- site renumbering procedures
- temp. addresses for privacy
- inter-domain multicast routing
- address propagation and AAA issues of different access scenarios
  - (always-on, dial-up, mobile,…)
- and, of course, transition / co-existence / interoperability with IPv4

Note: this indicates vitality, not incompleteness, of IPv6!



IPv6 Tutorial

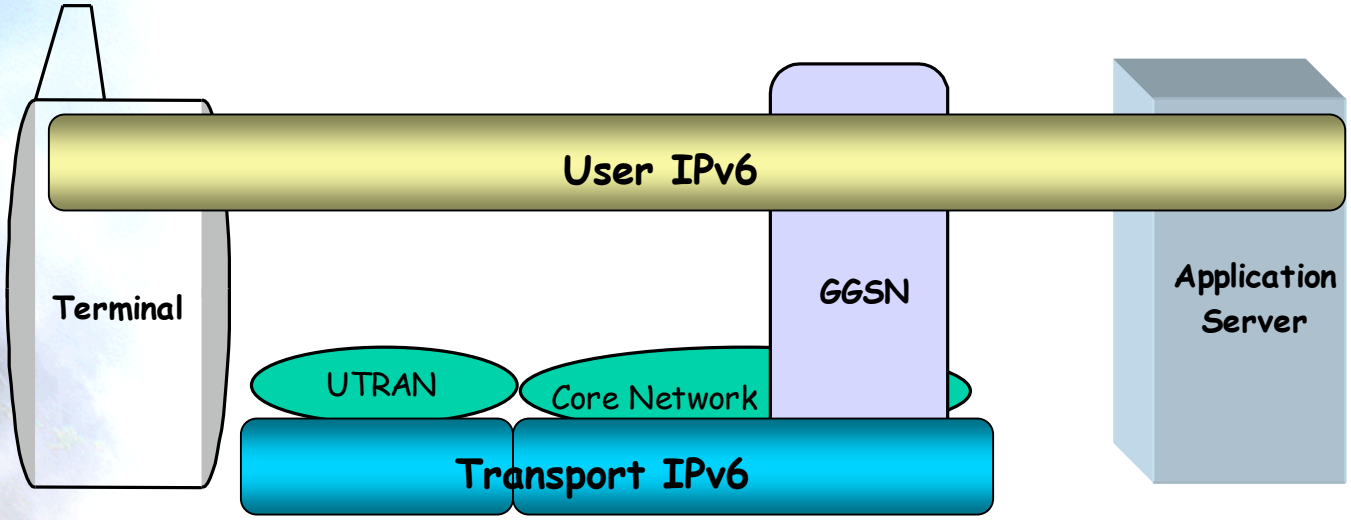
12. IPv6 in 3G

RFC3314: Rec. for IPv6 in 3GPP

- Informational: Recommendations from the IETF (IPv6 WG) to 3GPP community regarding the use of IPv6 in the 3GPP standards.
- Specifically, this document recommends that the 3GPP specify that:
  - Multiple prefixes may be assigned to each primary PDP context
  - Require that a given prefix must not be assigned to more than one primary PDP context
  - Allow 3GPP nodes to use multiple identifiers within those prefixes, including randomly generated identifiers.
- The IPv6 WG supports the use of IPv6 within 3GPP
- Recommendations in a spirit of open cooperation
- Since the original publication of this document as an Internet-Draft, the 3GPP has adopted the primary recommendations of this document.

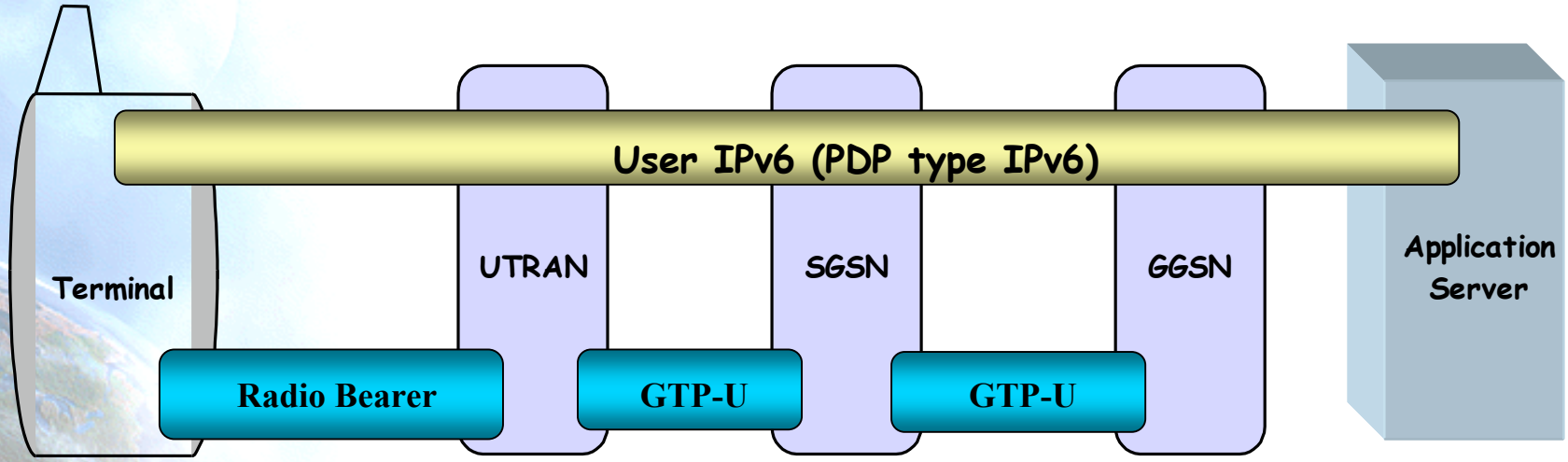
User plane vs. transport plane

- User and transport planes are completely independent:
  - –The transport plane can run on a different IP version than the user plane
- UTRAN and Core Network transport can also run on different IP versions

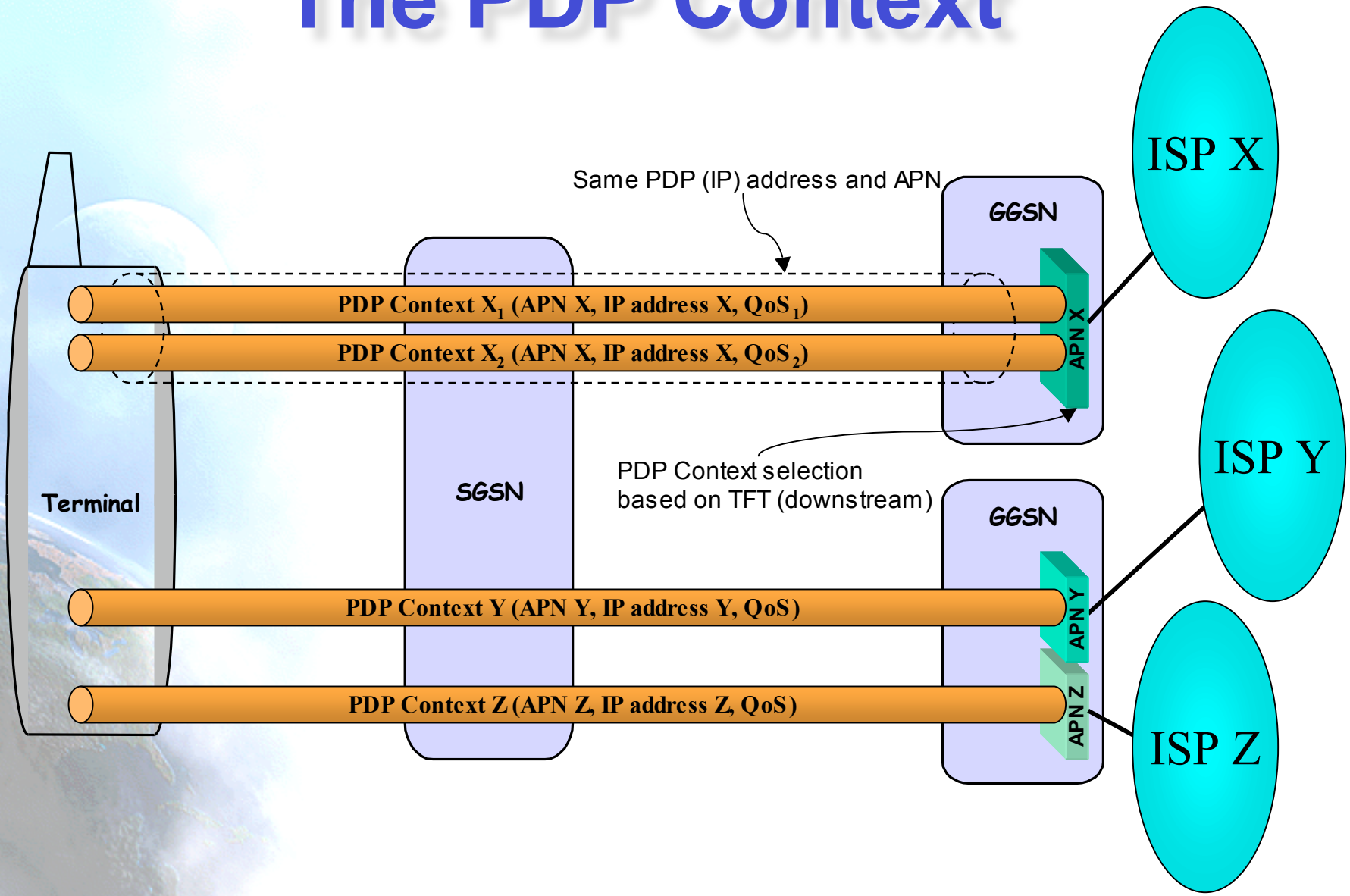


Transport of user IP packets

- IP packets to/from the terminal are tunneled through the UMTS network, they are not routed directly at the IP level.



The PDP Context



The PDP CONTEXT

- When an MS attaches to the Network, the SGSN creates a Mobility Management context containing information pertaining to e.g., mobility and security for the MS.
- At PDP Context Activation (PDP - Packet Data Protocol), the SGSN and GGSN create a PDP context, containing information about the session (e.g. IP address, QoS, routing information , etc.).
- Each Subscriber may activate several PDP Contexts towards the same or different GGSNs.
- When activated towards the same GGSN, they can use the same or different IP addresses.

The Access Point Name - APN

- The APN is a logical name referring to a GGSN. The APN also identifies an external network.
- The syntax of the APN corresponds to a fully qualified name.
- At PDP context activation, the SGSN performs a DNS query to find out the GGSN(s) serving the APN requested by the terminal.
- The DNS response contains a list of GGSN addresses from which the SGSN selects one address in a round-robin fashion (for this APN).

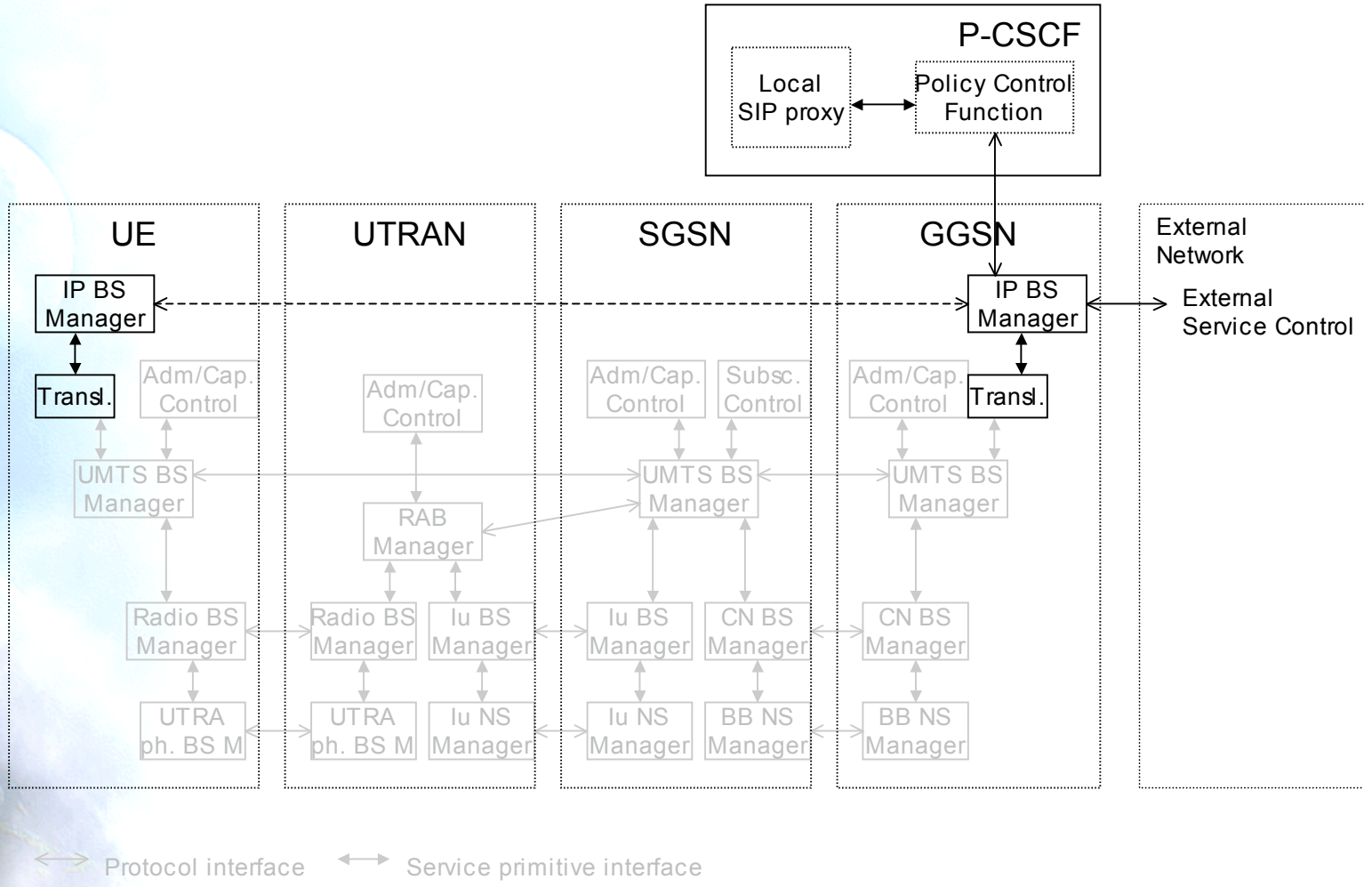
Traffic Flow Template (TFT)

- A TFT is a packet filter allowing the GGSN to classify packets received from the external network into the proper PDP context.
- A TFT consists of a set of packet filters, each containing a combination of the following attributes:
  - Source Address and Subnet Mask
  - Destination Port Range
  - Source Port Range
  - IPsec Security Parameter Index (SPI)
  - Type of Service (TOS) (IPv4) / Traffic Class (IPv6) and Mask
  - Flow Label (IPv6)

GPRS Tunneling Protocol

- GTP is a simple tunneling protocol based on UDP/IP, used both in GSM/GPRS and UMTS.
- A GTP tunnel is identified at each end by a Tunnel Endpoint Identifier (TEID)
- For every MS, one GTP-C tunnel is established for signaling and a number of GTP-U tunnels, one per PDP context (i.e. session), are established for user traffic.

QoS Management Functions



IP BS Manager

- Used to control the external IP bearer service to provide IP QoS end-to-end.
- Communicates with the UMTS BS manager through the translation function.
- Uses standard IP mechanisms to manage the IP bearer service.
- May exist both in the UE and the Gateway node, and it is possible that these IP BS Managers communicate directly with each other by using relevant signaling protocols, e.g., RSVP.
- Policy enforcement point for Service-based Local Policy control.

Policy Control Function (PCF)

- Logical entity that is co-located with the P-CSCF (the interface between the P-CSCF and PCF is not standardized in Release 5).
- Logical policy decision element which uses standard IP mechanisms to implement Service-based Local Policy in the bearer level.
- Enables coordination between events in the SIP session level and resource management in the bearer level.
- Makes policy decisions based on information obtained from the P-CSCF.
- Has a protocol interface with GGSN (Go interface) which supports the transfer of information and policy decisions between the policy decision point and the IP BS Manager in the GGSN (following COPS framework).

IP BS Manager capability in the UE and GGSN

Table 1: IP BS Manager capability in the UE and GGSN

Capability	UE	GGSN
DiffServ Edge Function	Optional	Required
RSVP/Intserv	Optional	Optional
IP Policy Enforcement Point	Optional	Required (*)

(*) Although the capability of IP policy enforcement is required within the GGSN, the control of IP policy through the GGSN is a network operator choice.

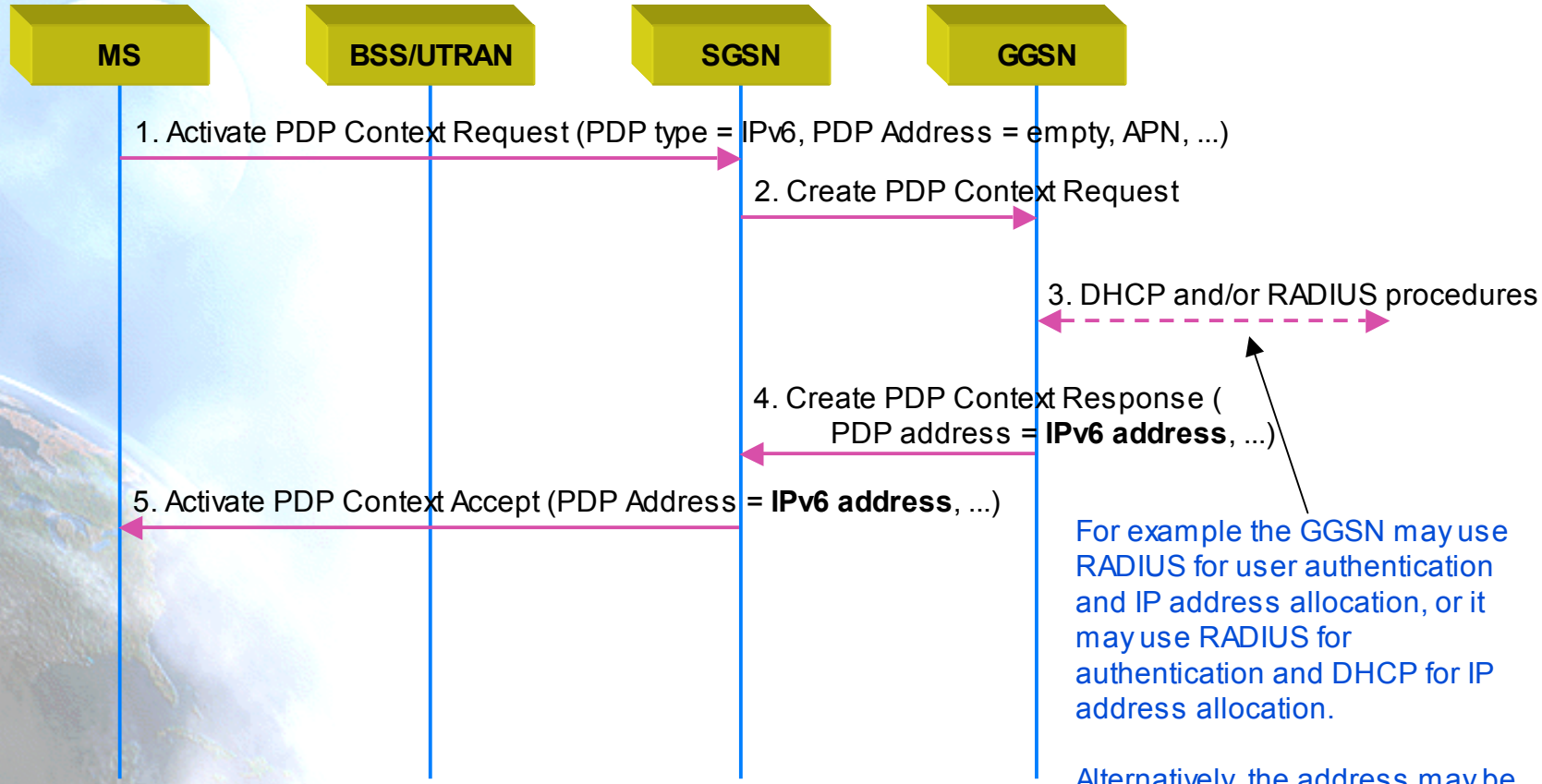
IPv6 History in UMTS

- IPv6 in the 3GPP standards
  - User plane:
    - PDP Type IPv6 introduced in GPRS R'97
  - Transport plane:
    - IPv6 is optional
  - UTRAN:
    - IP transport study is being conducted right now
  - IMS:
    - The IP Multimedia Core Network Subsystem has been standardized to be based on the following IPv6 support:
      - The architecture shall make optimum use of IPv6.
      - The IM CN subsystem shall exclusively support IPv6.
      - The UE shall exclusively support IPv6 for the connection to services provided by the IM CN subsystem.

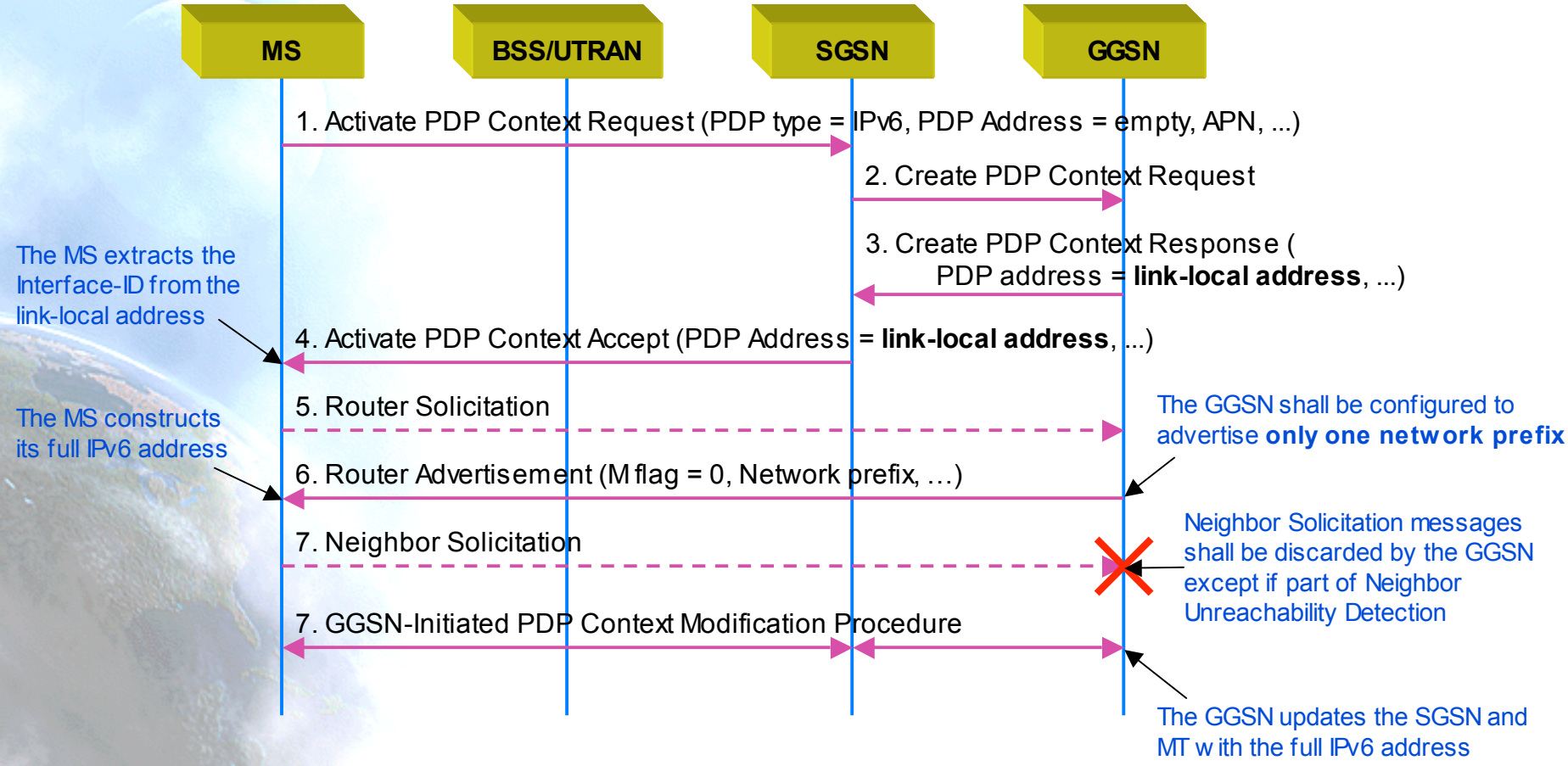
IPv6 Address Allocation Methods

- Stateless Address Autoconfiguration
  - Introduced in GPRS R'99
- Stateful Address Autoconfiguration
  - DHCPv6 client in the terminal
  - Requires DHCPv6 relay agent in the GGSN
- GPRS-specific Address Configuration
  - Static Address Configuration
    - The MS provides its statically configured IPv6 address at PDP context activation
  - Dynamic Address Allocation
    - The IPv6 address is provided by the GGSN at PDP context activation

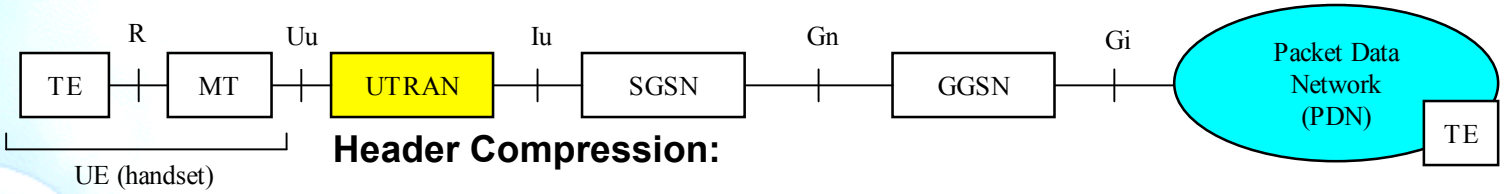
Dynamic Address Allocation in UMTS/GPRS



Stateless Address Auto-configuration in UMTS/GPRS



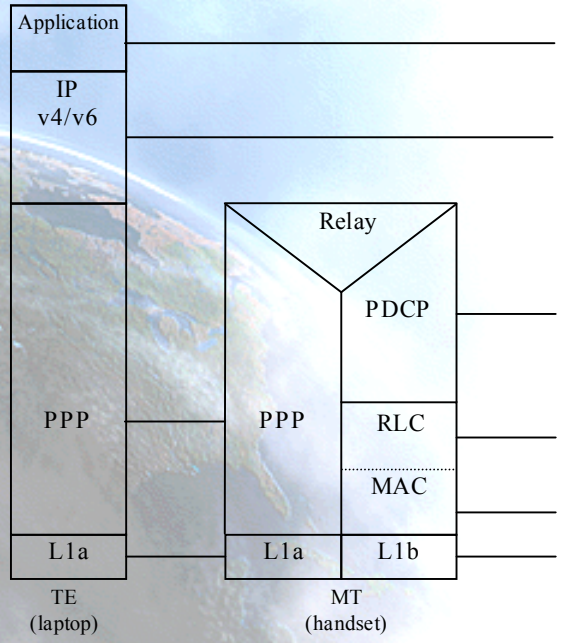
Header Compression



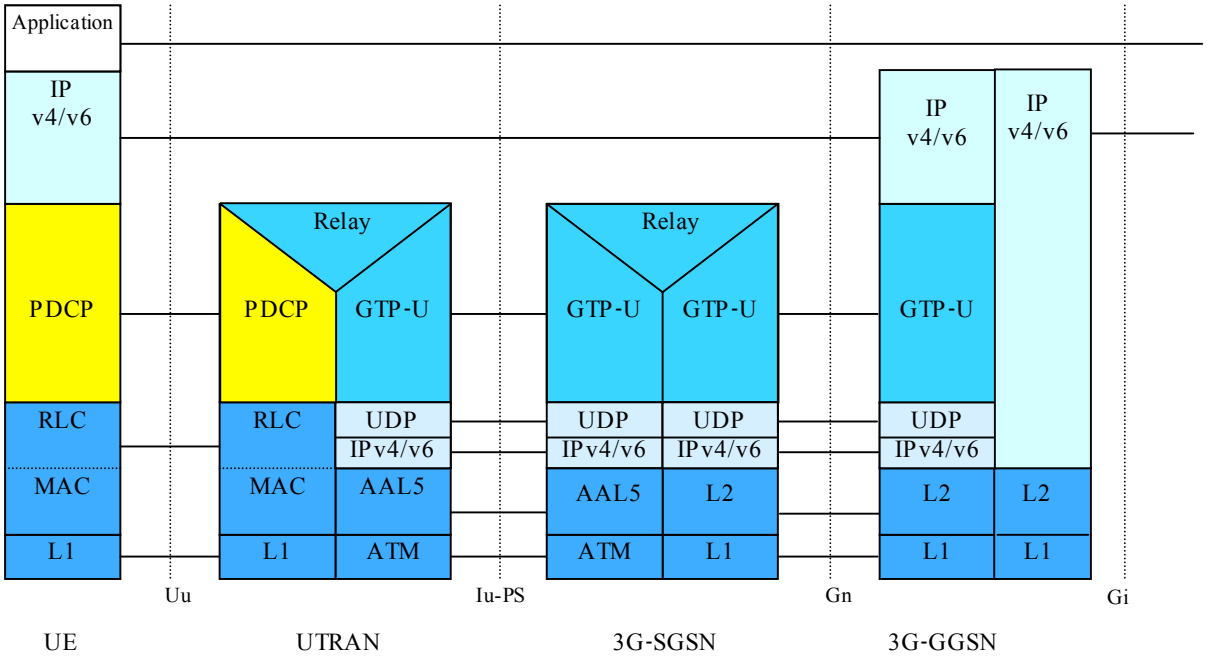
Header Compression:

- RFC2507, RFC…

Laptop+Handset



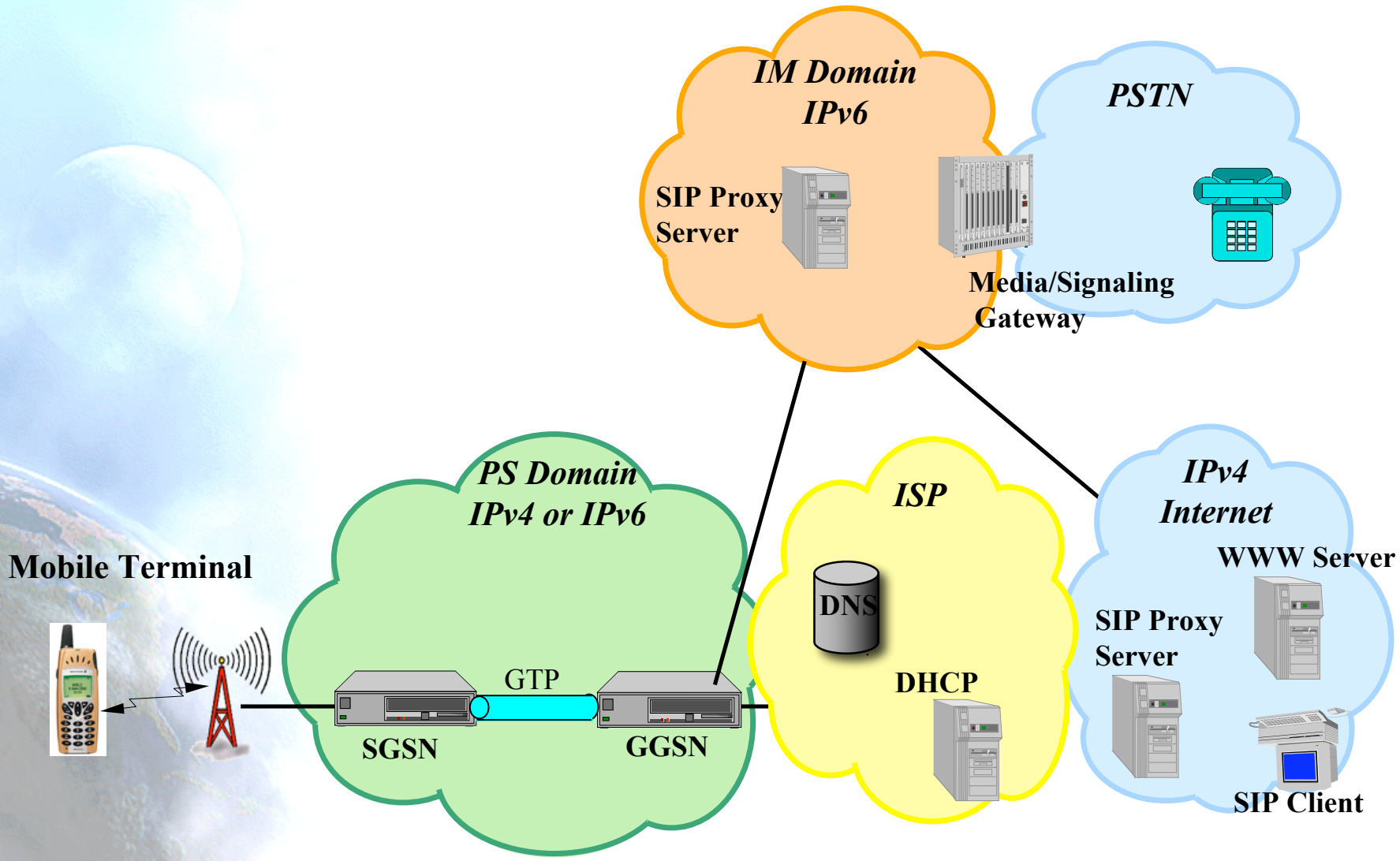
PS Domain User Plane protocol stack



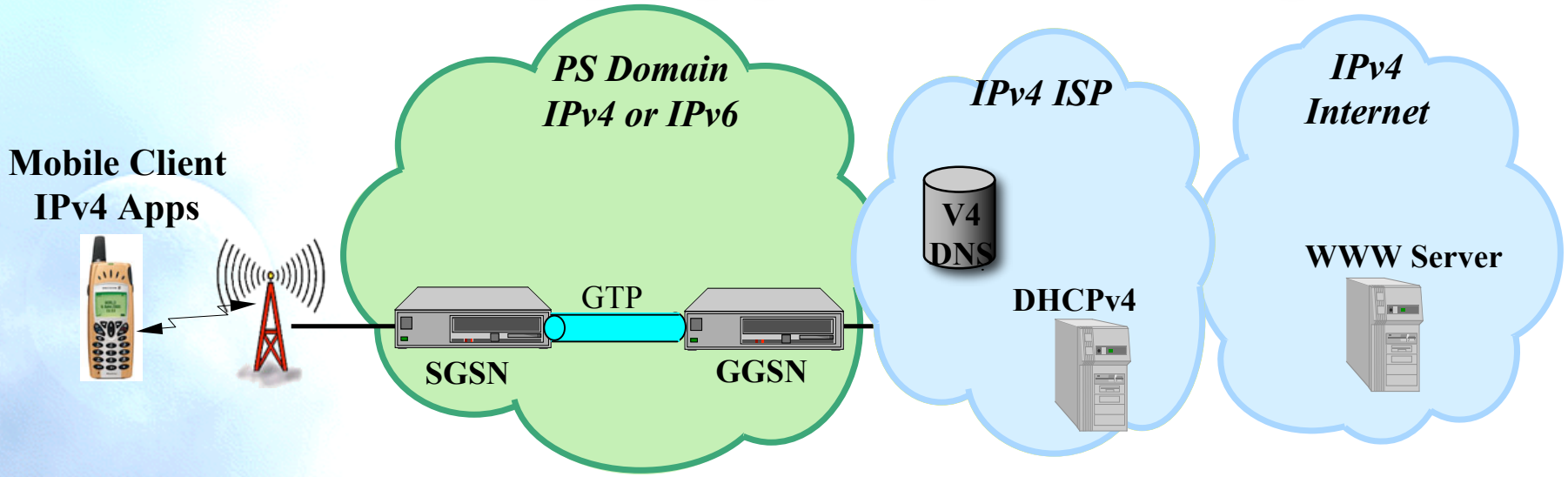
IPv4/IPv6 Transition

- Text in 23.221 shows examples of transition:
  - Dual Stack
  - NAT/PT
  - Tunneling
- These are only examples to show how transition could be done.
- They are not mandatory to implement/deploy.

3G Rel. 5 Architecture: Services



WWW Access via IPv4 ISP



Access to operators GGSN

IPv4 Address allocation

Dynamic DNS update

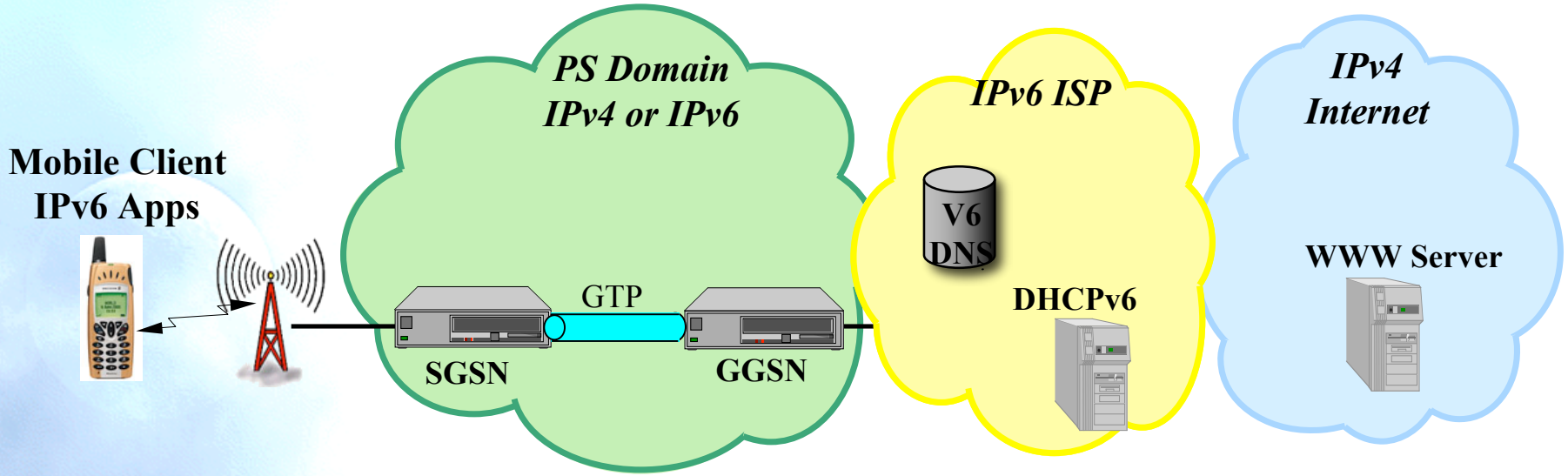
GGSN Allocated IPv4 Address

IPv4 Address

IPv4 Data

This IPv4 address is probably not permanent

WWW Access via IPv6 ISP



Access to operators GGSN

IPv6 Address allocation

Dynamic DNS update

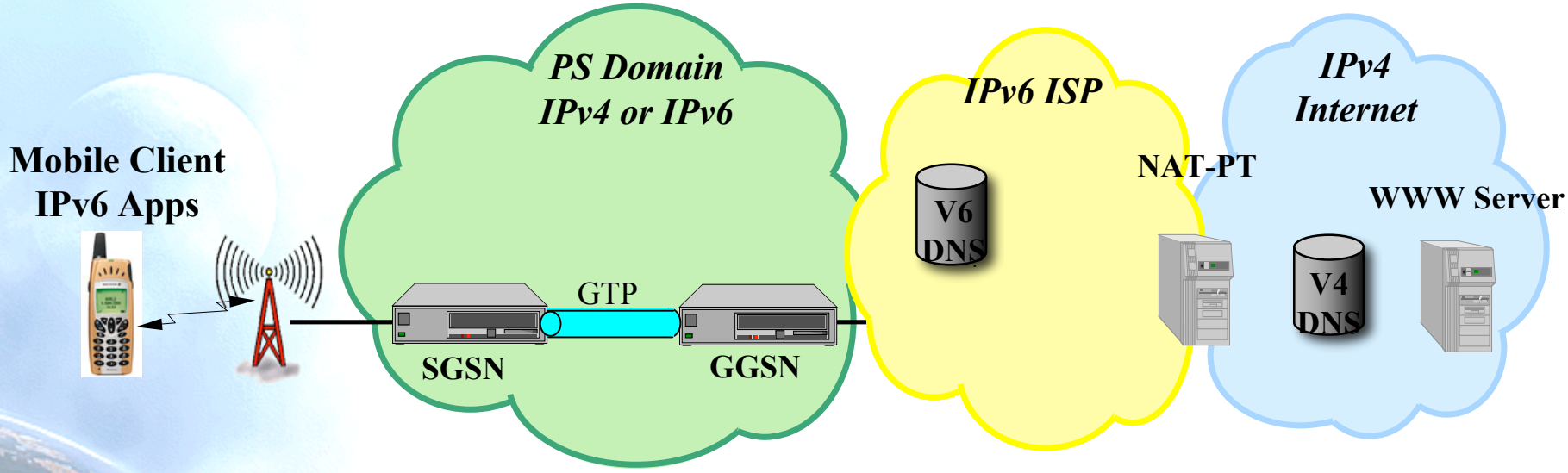
GGSN Allocated IPv6 Address

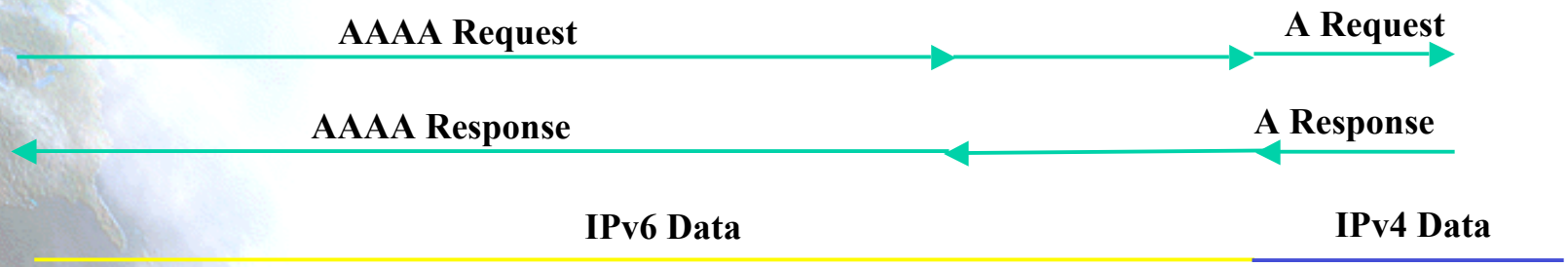
IPv6 Address

IPv6 Data

This IPv6 address could be permanent, best effort QoS for www traffic, etc.

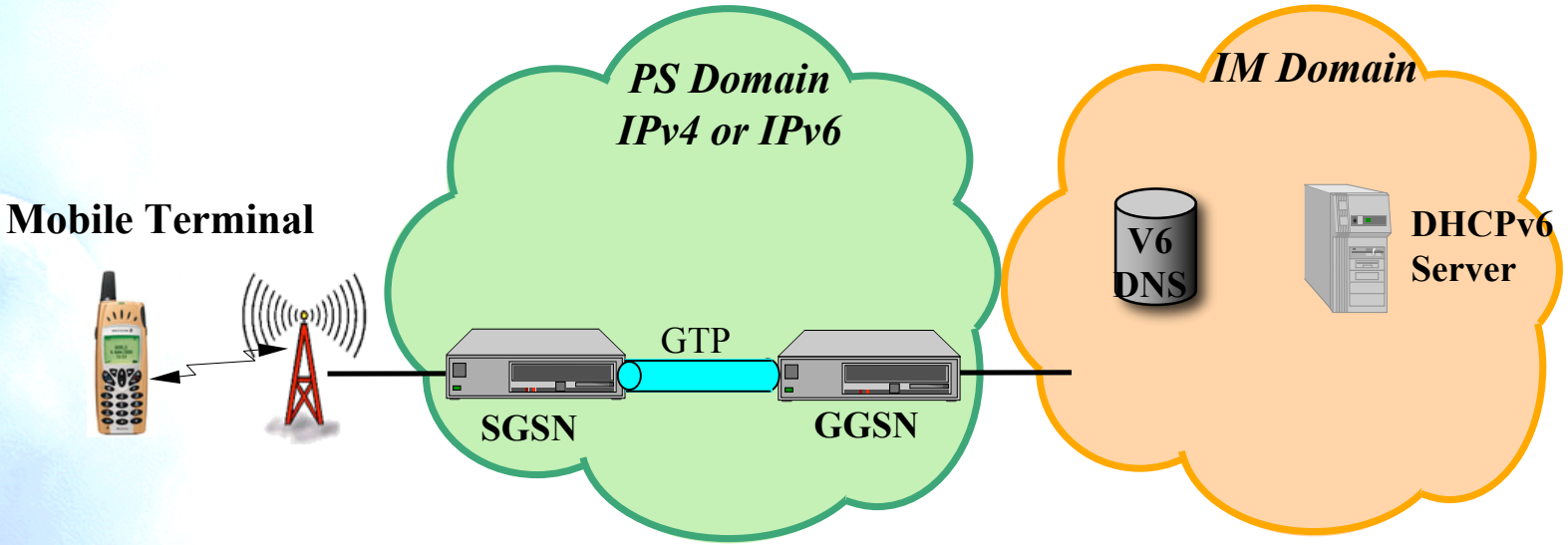
WWW Access via IPv6 ISP with NAT-PT





Also works for incoming calls but:
With NAT-PT loss end to end transparency

IM IPv6 Address Allocation



Access to operators GGSN

IPv6 Address allocation

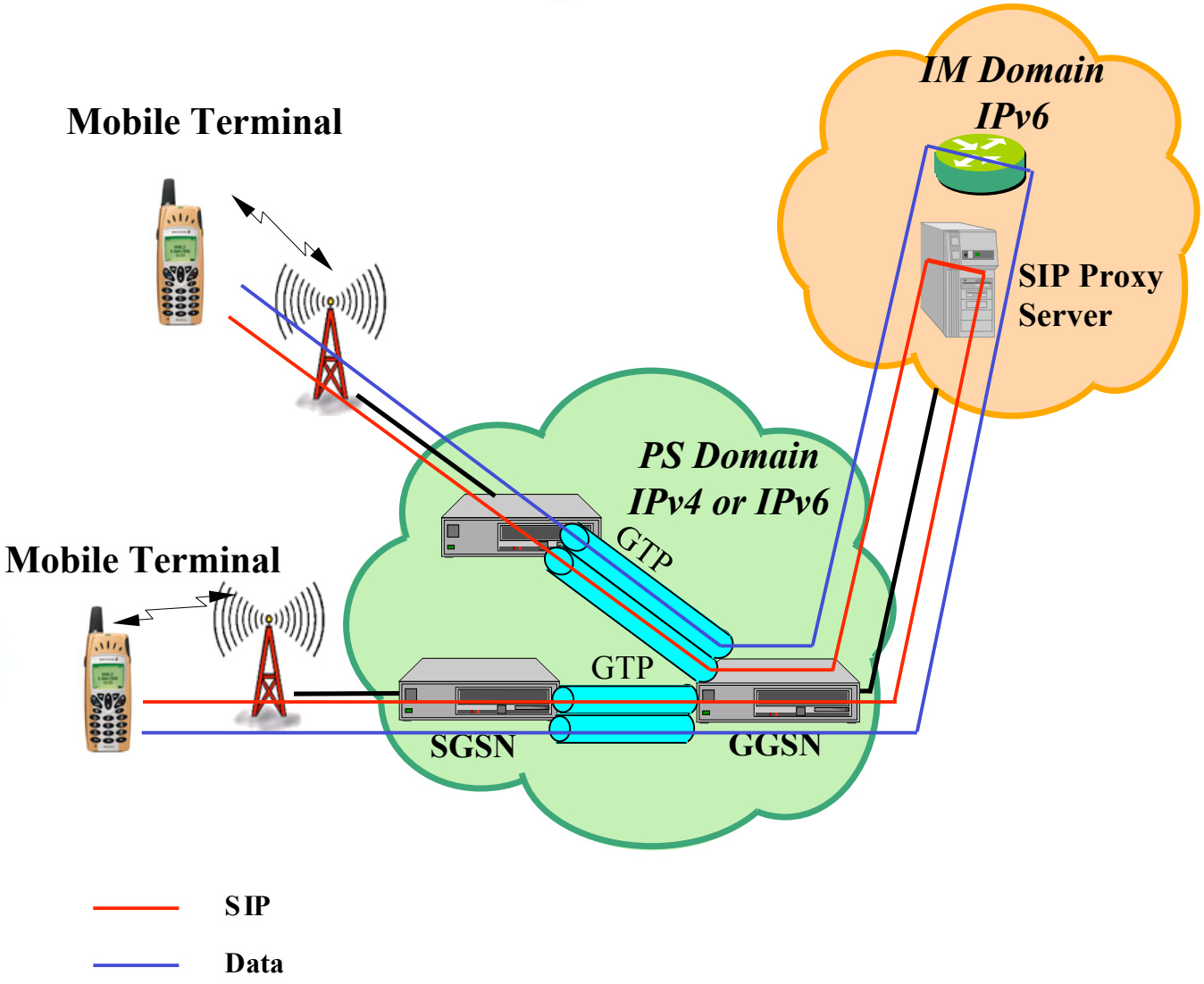
Dynamic DNS update

GGSN Allocated IPv6 Address

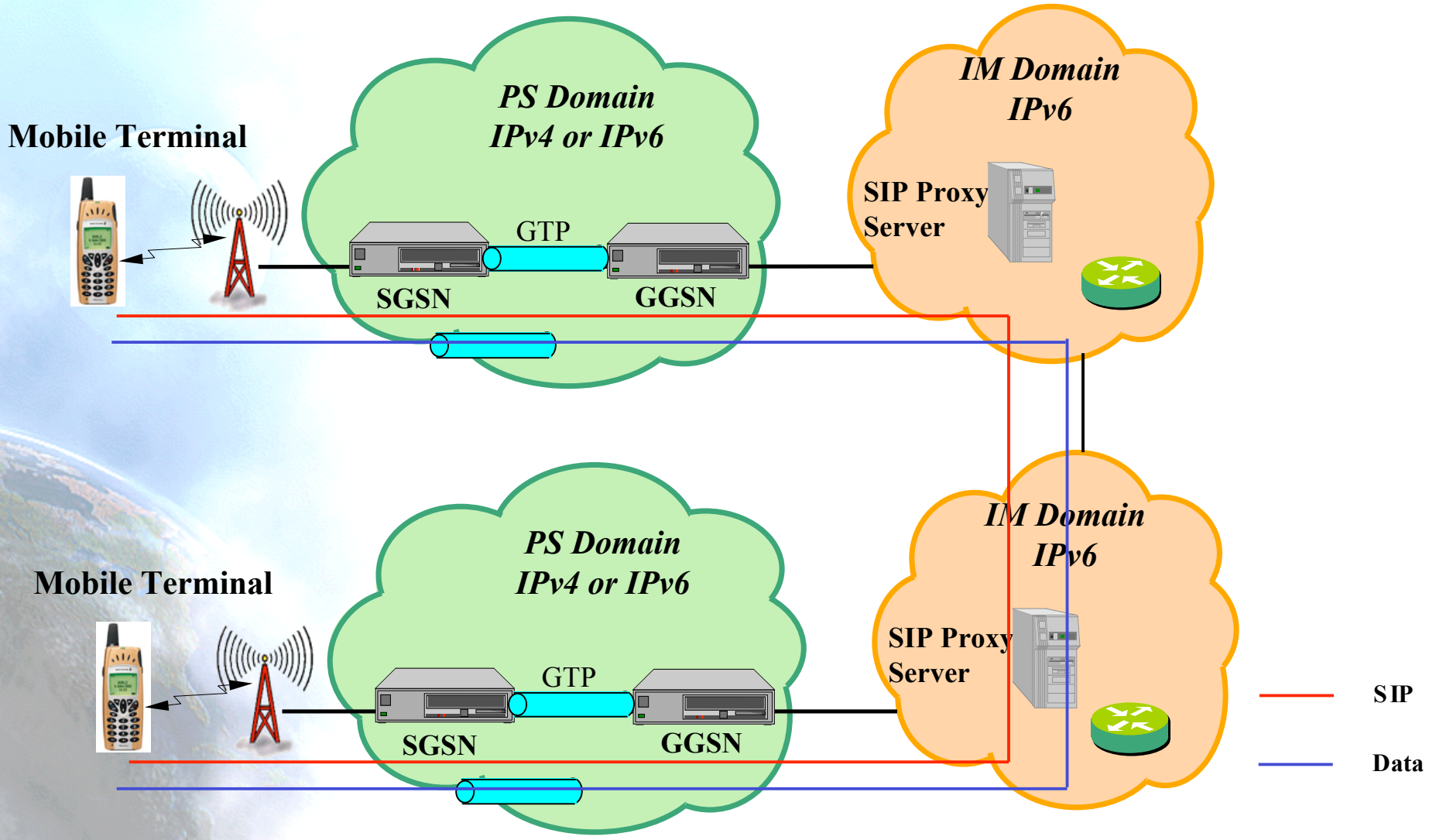
IPv6 Address

This IPv6 address is permanent, GTP tunnel has QoS appropriate to SIP/others signaling

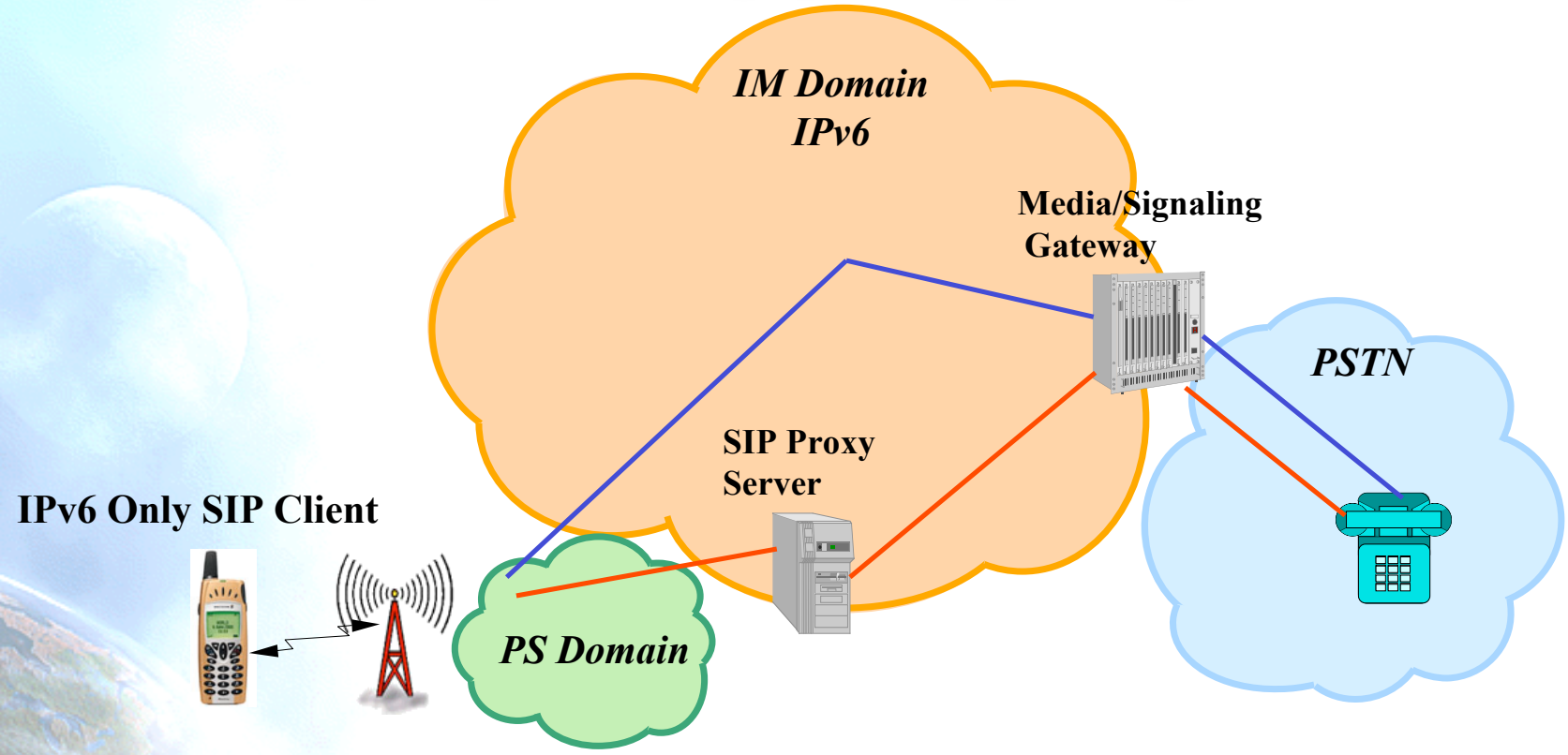
IM Call one Operator



IM Call between Operator



Mobile IM Call to PSTN

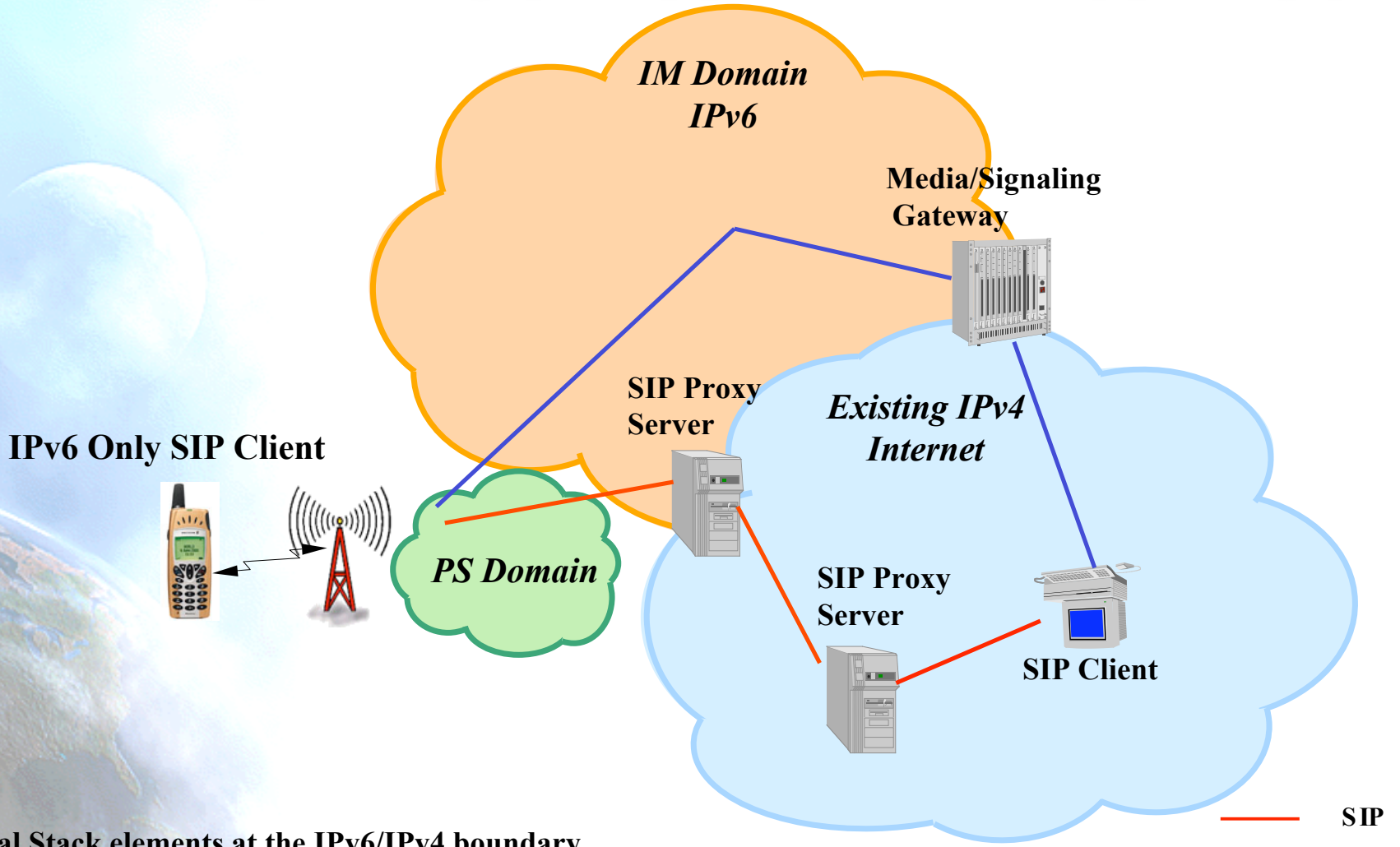


Interworking done at edge of IM Domain

Signaling

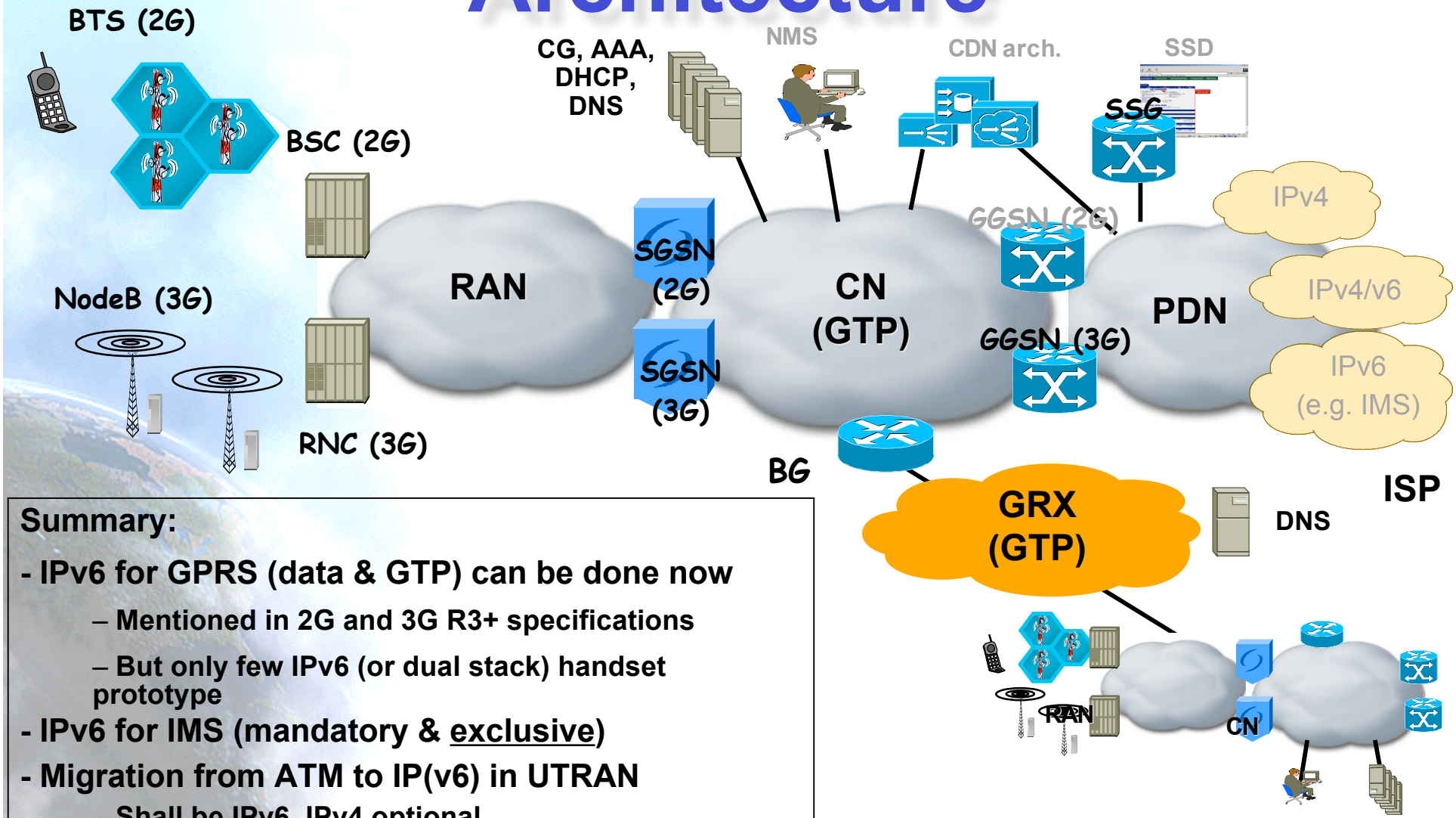
Data

Mobile IM Call to IPv4 Internet



Dual Stack elements at the IPv6/IPv4 boundary

3GPP Mobile Wireless Network Architecture



Summary:

- IPv6 for GPRS (data & GTP) can be done now
  - Mentioned in 2G and 3G R3+ specifications
  - But only few IPv6 (or dual stack) handset prototype
- IPv6 for IMS (mandatory & exclusive)
- Migration from ATM to IP(v6) in UTRAN
  - Shall be IPv6, IPv4 optional
  - dual-stack recommended
- IP (v4 or v6) for user applications

IPv6 WG: IPv6 for 2/3G Cellular

- draft-ietf-ipv6-cellular-host-03.txt
- Informational.
- Audience: Implementers of cellular hosts that will be used with GPRS, 3GPP UMTS Release 99, Release 4, Release 5, or future releases of UMTS.
- Goal:
  - Guidance on which parts of IPv6 to implement in such cellular hosts.
  - Issues relating to the use of these components when operating in these networks.
- May also apply to other cellular link types (no detailed analysis): Is a topic of future work.
- Not a definitive list of IPv6 functionality for cellular links other than those listed above.
- Future changes in 3GPP that require changes in host implementations, may result in updates to this document.

IPv6 WG: Node Requirements

- draft-ietf-ipv6-node-requirements-02.txt
- Goal:
  - Minimal set of functionality required for IPv6 nodes.
  - To ensure interoperability.
- Assumes that all IPv6 nodes meet the minimum requirements specified.
- Many IPv6 nodes will implement optional or additional features.

Draft Contents

- Sub-IP Layer
  - Follow link-layer specs (Ethernet, PPP, ATM PVC, …)
- IP Layer
- Transport and DNS
- IPv4 Support & Transition
  - Is interoperability with IPv4 a requirement? If so, use native addresses and RFC2893 - Transition Mechanisms for IPv6 Hosts and Routers.
- Mobility
  - What should nodes implement for MIPv6? Possible Mobil Node and Home Agent.
- Security
- Router Functionality (?)
  - If yes, then RFC2711 - IPv6 Router Alert Option and RFC2461 - Neighbor Discovery for IPv6)
- Network Management (?)

IP Layer

- General (RFC2460)
- Neighbor Discovery (RFC2461)
- Path MTU Discovery & Packet Size (RFC1981, 2675)
- ICMPv6 (RFC2463)
- Addressing
  - RFC2373 - IP Version 6 Addressing Architecture
  - RFC2462 - IPv6 Stateless Address Autoconfiguration
  - RFC3041 - Privacy Extensions for Address Configuration in IPv6
  - Default Address Selection for IPv6
  - Stateful Address Autoconfiguration
- Other
  - RFC2473 - Generic Packet Tunneling in IPv6 Specification
  - RFC2710 - Multicast Listener Discovery (MLD) for IPv6

Transport and DNS

- Transport Layer
  - RFC2147 - TCP and UDP over IPv6 Jumbograms
- DNS
  - What is the level of support for DNS in a node? Is it optional? Not all nodes will need to resolve addresses
  - RFC2874 - DNS Extensions to Support IPv6 Address Aggregation and Renumbering
  - RFC2732 - Format for Literal IPv6 Addresses in URL's
- Other
  - Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

Security

- Basic Architecture (RFC2401, 2402, 2406)
- Security Protocols (ESP, AH)
- Transforms and Algorithms (RFC2405, 2451, 2410, 2404, 2403, 2104)
- Key Management Method (RFC2407, 2408, 2409)

v6ops-3GPP Design Team: Scope & Goal

- Identify relevant transition scenarios
- Map relevant transition mechanisms to the scenarios
  - Identify relevant transition mechanisms
  - Perform “Gap Analysis” (identify missing transition tools)
- Make analysis for usage of transition tools
- Document the results
  - Scenarios
  - Solutions
  - Gaps
  - Recommendations
- Discuss these in the WG
- A Non-Goals
  - Specify new transition mechanisms
  - Change 3GPP specs

Status of the documents

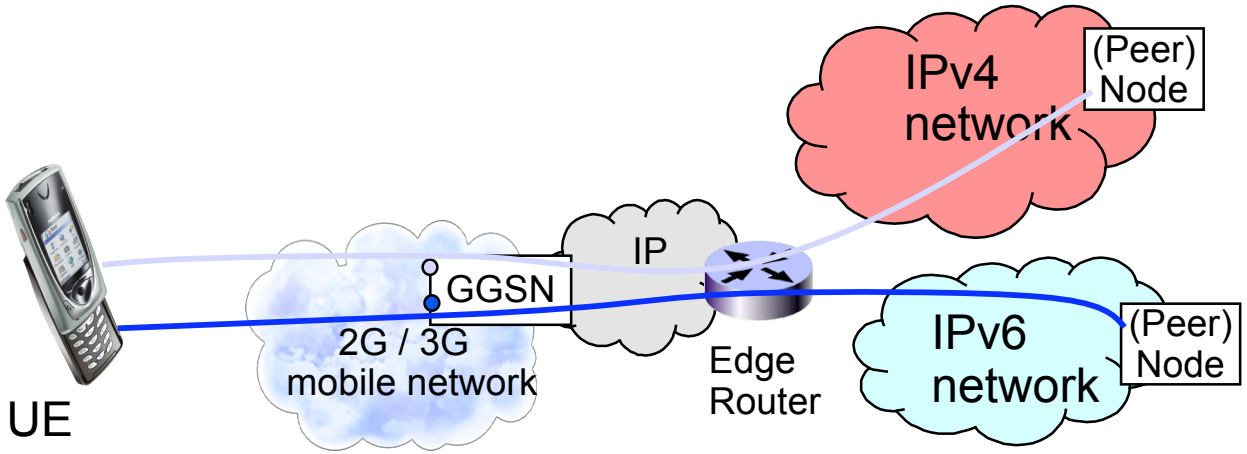
- Scenarios document
  - RFC3574
- Analysis
  - draft-ietf-v6ops-3gpp-analysis-10.txt

Scenarios

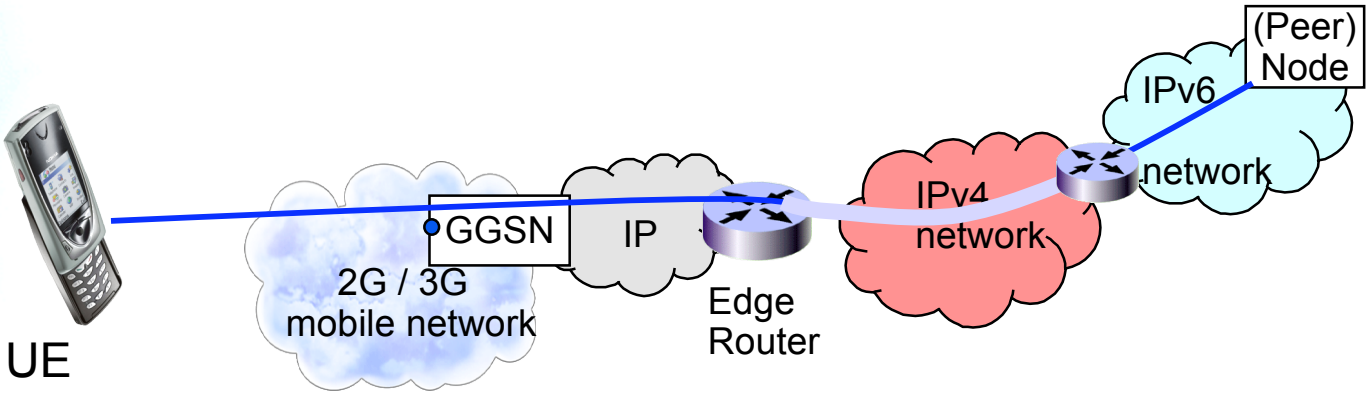
1. GPRS Scenarios
    1. Dual Stack UE connecting to IPv4 and IPv6 nodes
    2. IPv6 UE connecting to an IPv6 node through an IPv4 network
    3. IPv4 UE connecting to an IPv4 node through an IPv6 network
    4. IPv6 UE connecting to an IPv4 node
    5. IPv4 UE connecting to an IPv6 node
2. Transition scenarios with IMS
    1. UE connecting to a node in an IPv4 network through IMS
    2. Two IPv6 IMS islands connected via an IPv4 network

GPRS scenarios 1 and 2

1. Dual stack UE connecting to IPv4 and IPv6 nodes

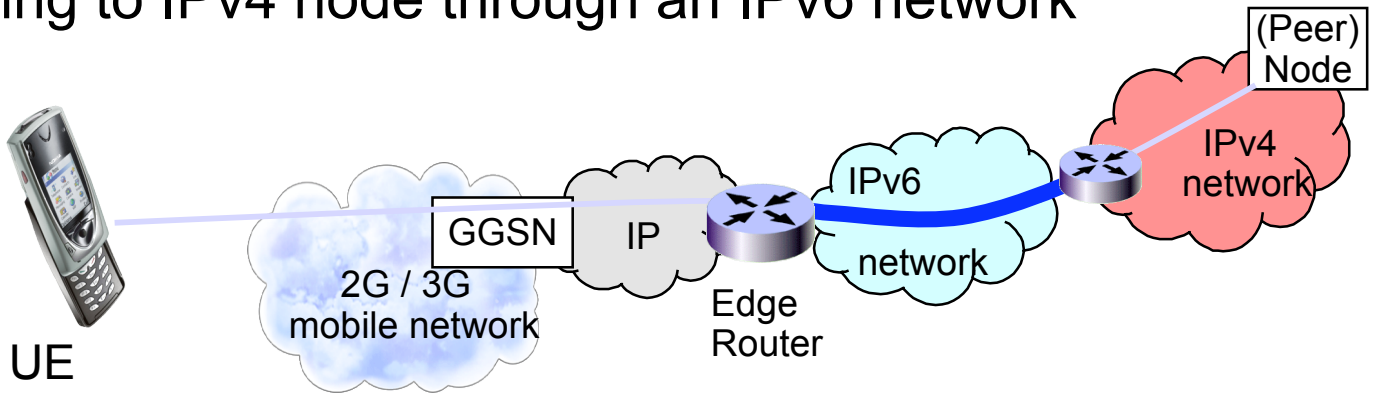


2. IPv6 UE connecting to IPv6 node through an IPv4 network

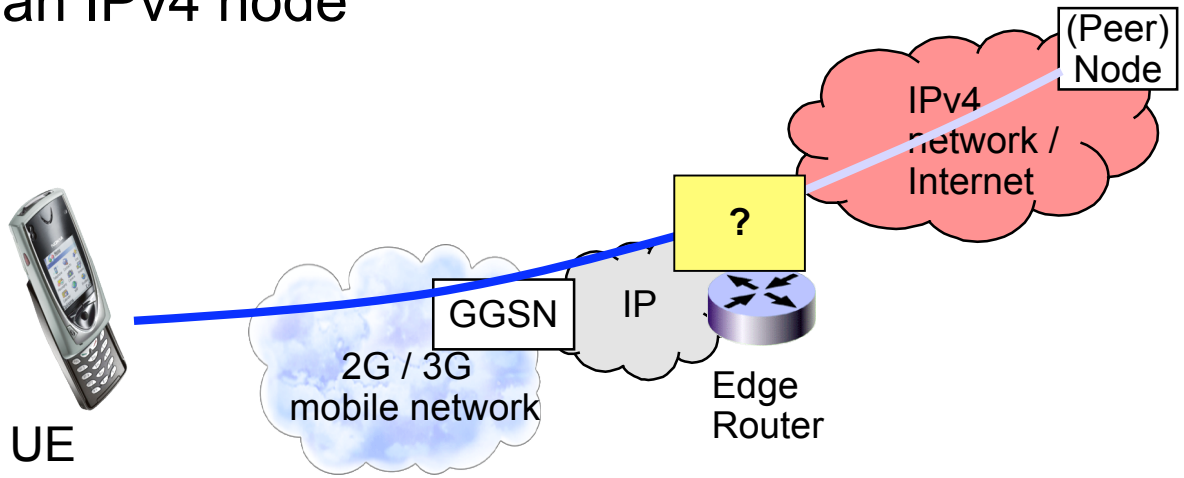


GPRS scenarios 3 and 4

3. IPv4 UE connecting to IPv4 node through an IPv6 network

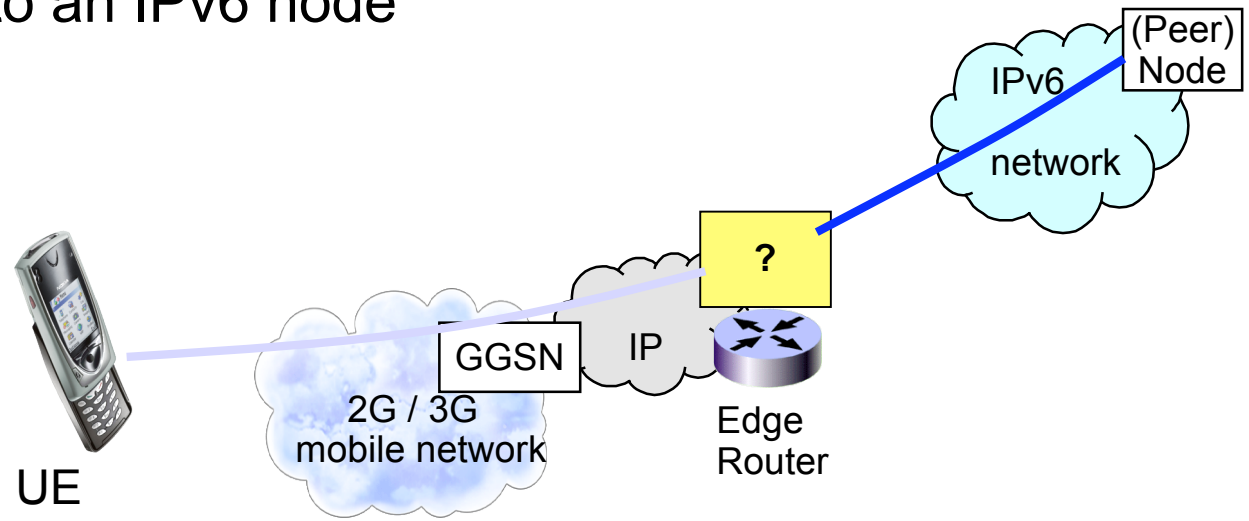


4. IPv6 UE connecting to an IPv4 node



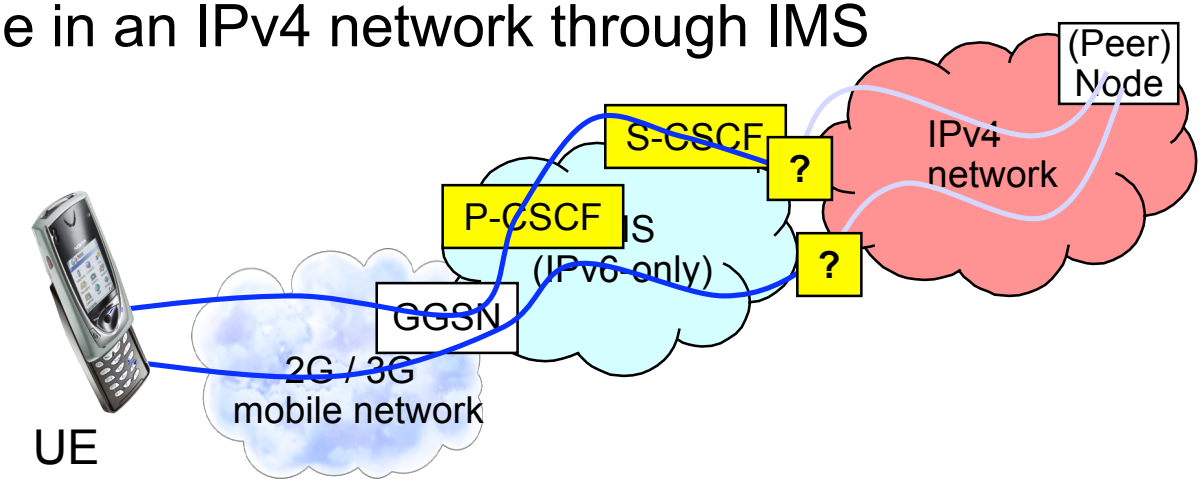
GPRS scenario 5

5. IPv4 UE connecting to an IPv6 node

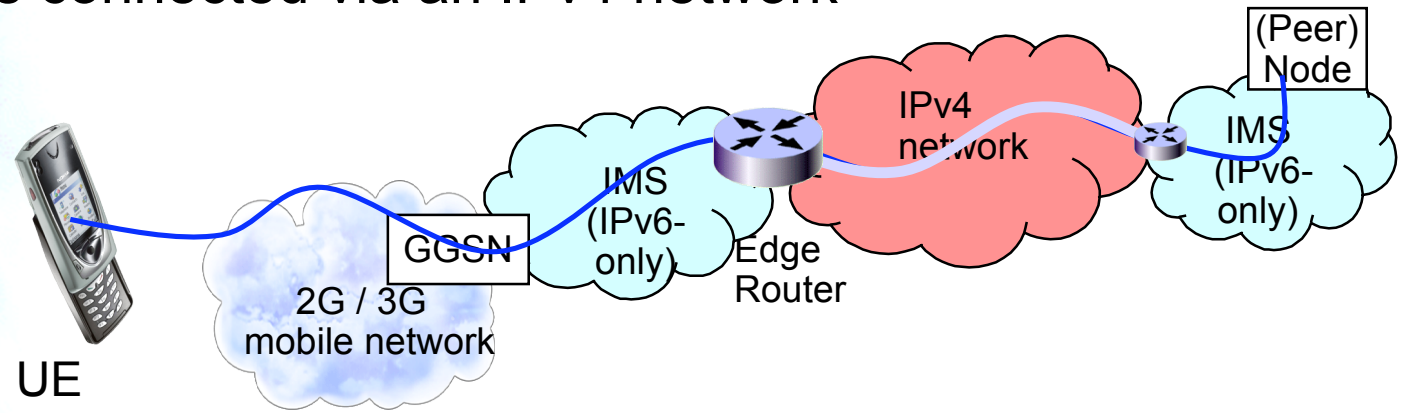


IMS scenarios 1 and 2

1. UE connecting to a node in an IPv4 network through IMS



2. Two IMS islands connected via an IPv4 network



Thanks !

Contact:

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

- Some Slides provided by:
  - Patrick Grossetete, John Loughney, Jonne Soininen

Madrid 2005 IPv6 Summit, soon available at:

www.ipv6-es.com