IPv6 Training

KENIC-AFRINIC IPv6 Workshop
17th – 20th June 2008

César Olvera (cesar.olvera@consulintel.es)
Jordi Palet (jordi.palet@consulintel.es)
Alvaro Vives (alvaro.vives@consulintel.es)

Agenda

1. Basic Introduction to IPv6
2. Header Formats & Packet Size Issues
3. Addressing
4. ICMPv6, Neighbor Discovery & DHCPv6
5. Applications
6. IPv6 DNS
7. Security/firewalling
8. Transition and Coexistence
9. Mobility
10. Routing

IPv6 Tutorial

8. Transition and Coexistence

Agenda

8.1. Transition concepts

8.2. Dual Stack

8.3. Tunnels

8.4. Tunnel Broker

8.5. 6to4

8.6. Teredo

8.7. Softwires

8.8. Translation

8.9. Security

8.1. Transition concepts

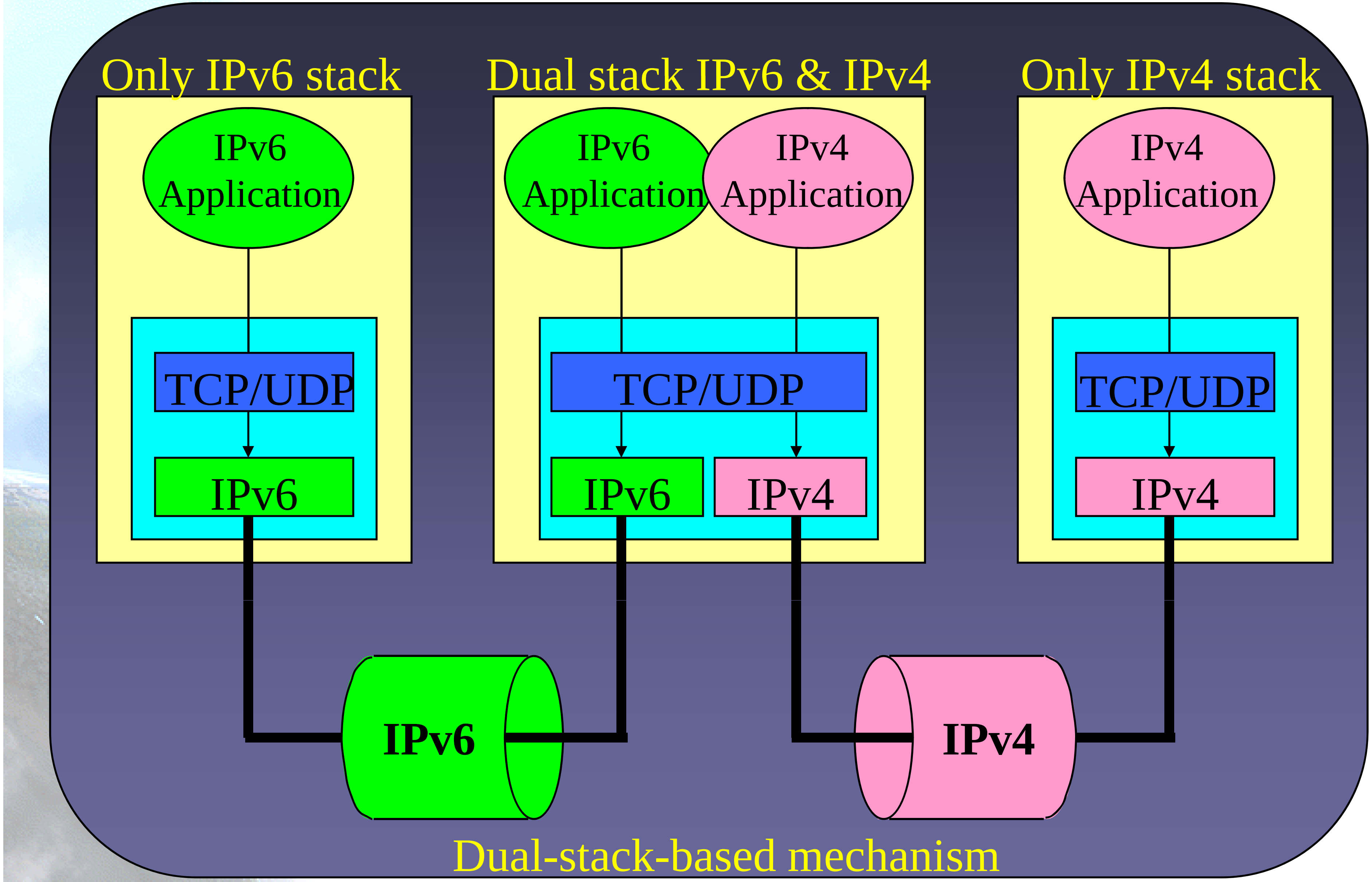
Transition / Co-Existence Techniques

- IPv6 has been designed for easing the transition and coexistence with IPv4
- Several strategies have been designed and implemented for coexisting with IPv4 hosts, grouped in three types:
  - Dual stack: Simultaneous support for both IPv4 and IPv6 stacks
  - Tunnels: IPv6 packets encapsulated in IPv4 ones
    - This is the commonest choice
  - Translation: This should be the last choice because it isn't perfect

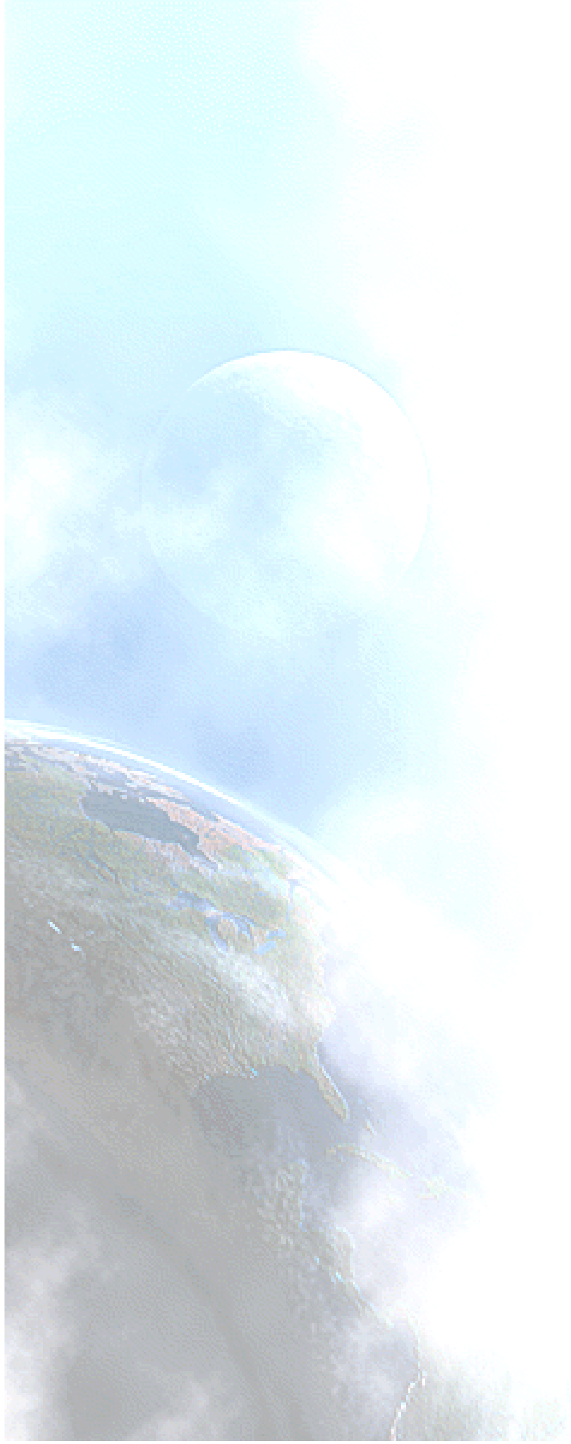
8.2. Dual Stack

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



Dual-stack-based mechanism

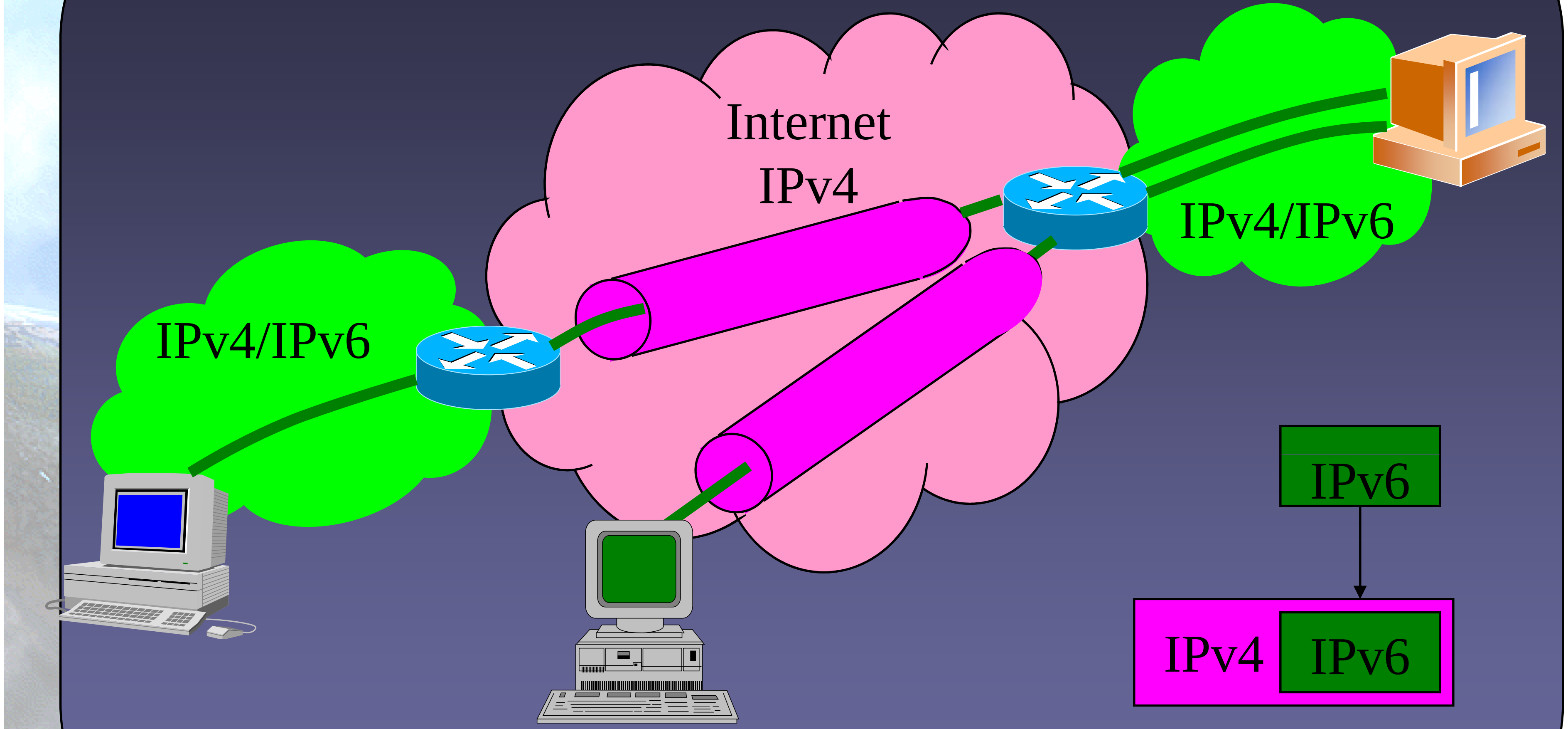


8.3. Tunnels

Tunnels to Get Through IPv6-Ignorant Routers

- Encapsulate IPv6 packets inside IPv4 packets (or MPLS frames) in order to provide IPv6 connectivity through IPv4-only networks
- Many methods exist for establishing tunnels:
  - manual configuration
  - “tunnel brokers” (using web-based service to create a tunnel)
  - “6over4” (intra-domain, using IPv4 multicast as virtual LAN)
  - “6to4” (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)

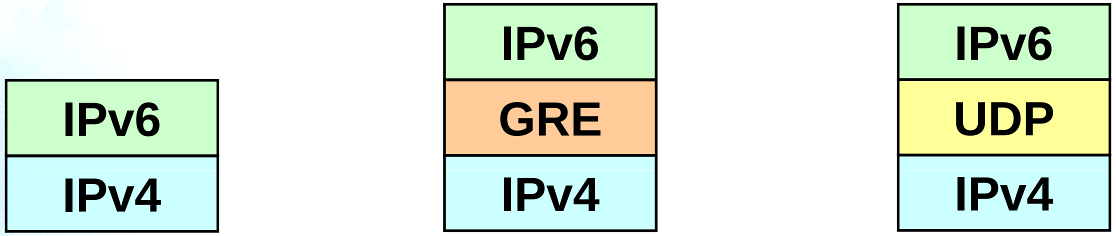
Tunnels IPv6 in IPv4 (1)



Tunnel-based mechanism

Tunnels IPv6 in IPv4 (2)

- There are different ways for encapsulating the IPv6 packets into IPv4 ones



IPv6
IPv4

IPv6
GRE
IPv4

IPv6
UDP
IPv4

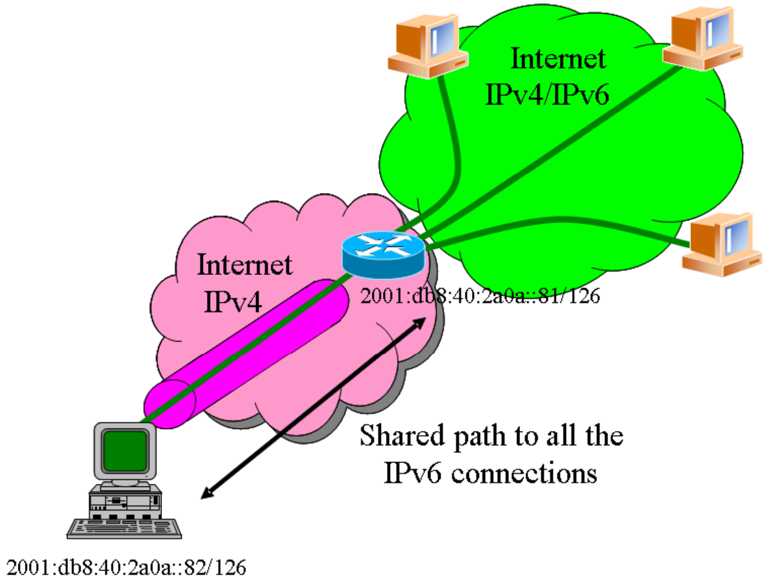
- Same for IPv4 being used in IPv6-only networks

Tunnels IPv6 in IPv4 (3)

- Some transition mechanism based on tunnels:
  - 6in4 (*) [6in4]
  - TB (*) [TB]
  - TSP [TSP]
  - 6to4 (*) [6to4]
  - Teredo (*) [TEREDO], [TEREDOC]
  - Automatic tunnels [TunAut]
  - ISATAP [ISATAP]
  - 6over4 [6over4]
  - AYIYA [AYIYA]
  - Silkroad [SILKROAD]
  - DSTM [DSTM]
  - Softwires (*) [SOFTWIRES]

- (*) Commoner mechanisms and explained in depth in the following slides

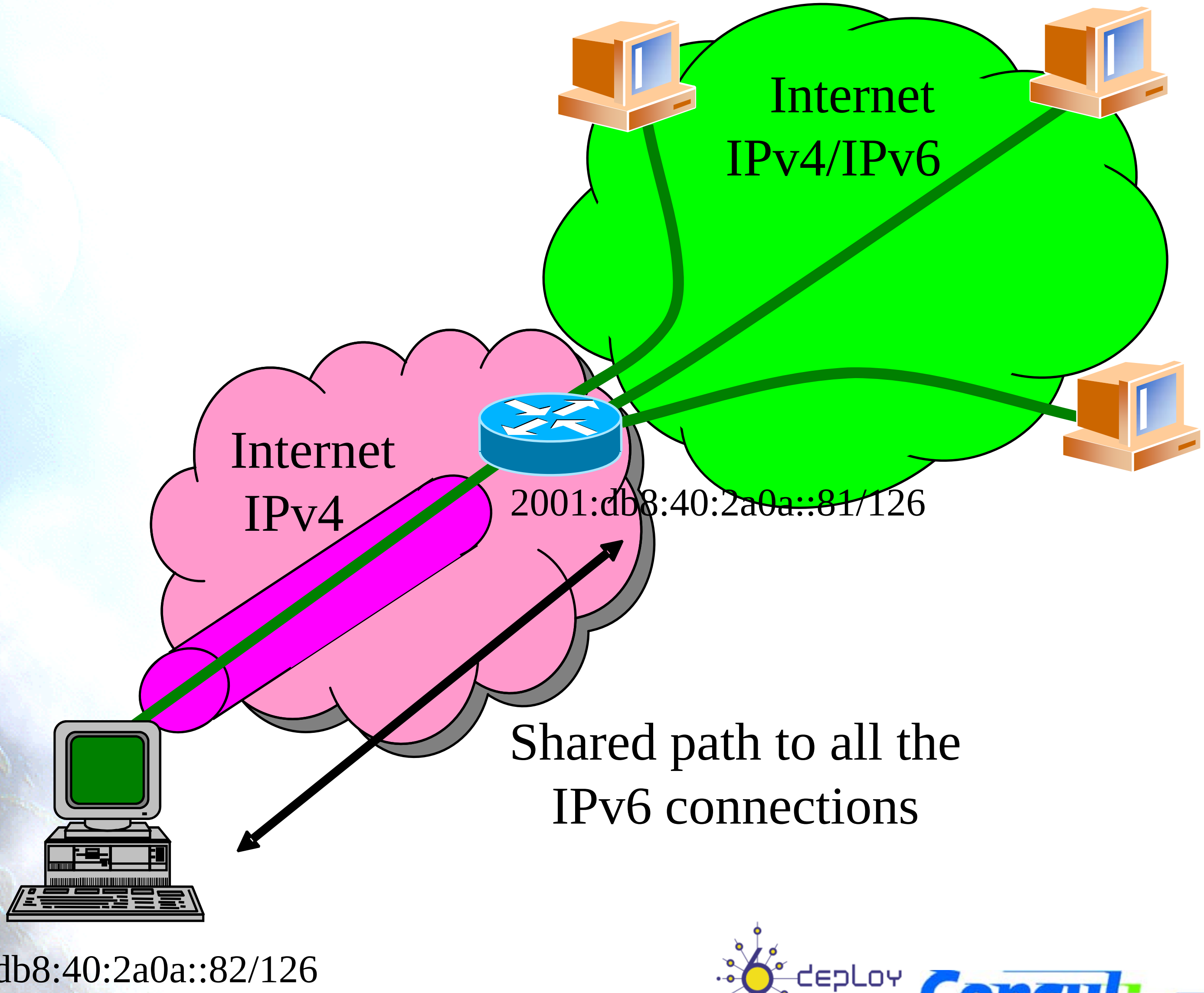
6in4 Tunnels Details



- It encapsulates directly the IPv6 packet into the IPv4 packet
- It is usually used between:
  - end host ==> router
  - router ==> router
- However, it is also possible for
  - end host ==> end host
- From the point of view of IPv6 the tunnel is considered as a point-to-point link
  - Only an IPv6 network-hop although several IPv4-hops exist in the path
- The IPv6 addresses of both tunnel-ends belong to the same prefix
- All the IPv6 connections of the end-host flow always through the router located at the tunnel-end-point
- The 6in4 tunnels can be built from end-hosts located behind a NAT box
  - It is essential that the NAT implementation supports “proto-41 forwarding” [PROTO41] to let the IPv6-encasulated packets traverse the NAT box

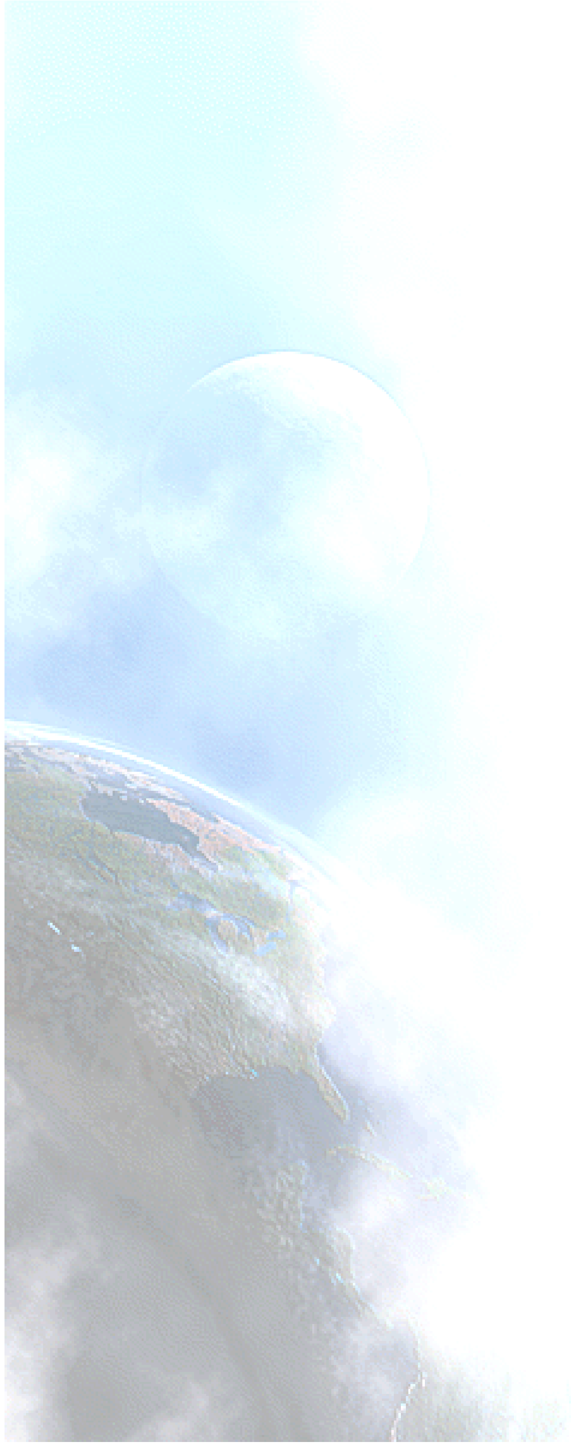
8.4. Tunnel Broker

Tunnel Broker



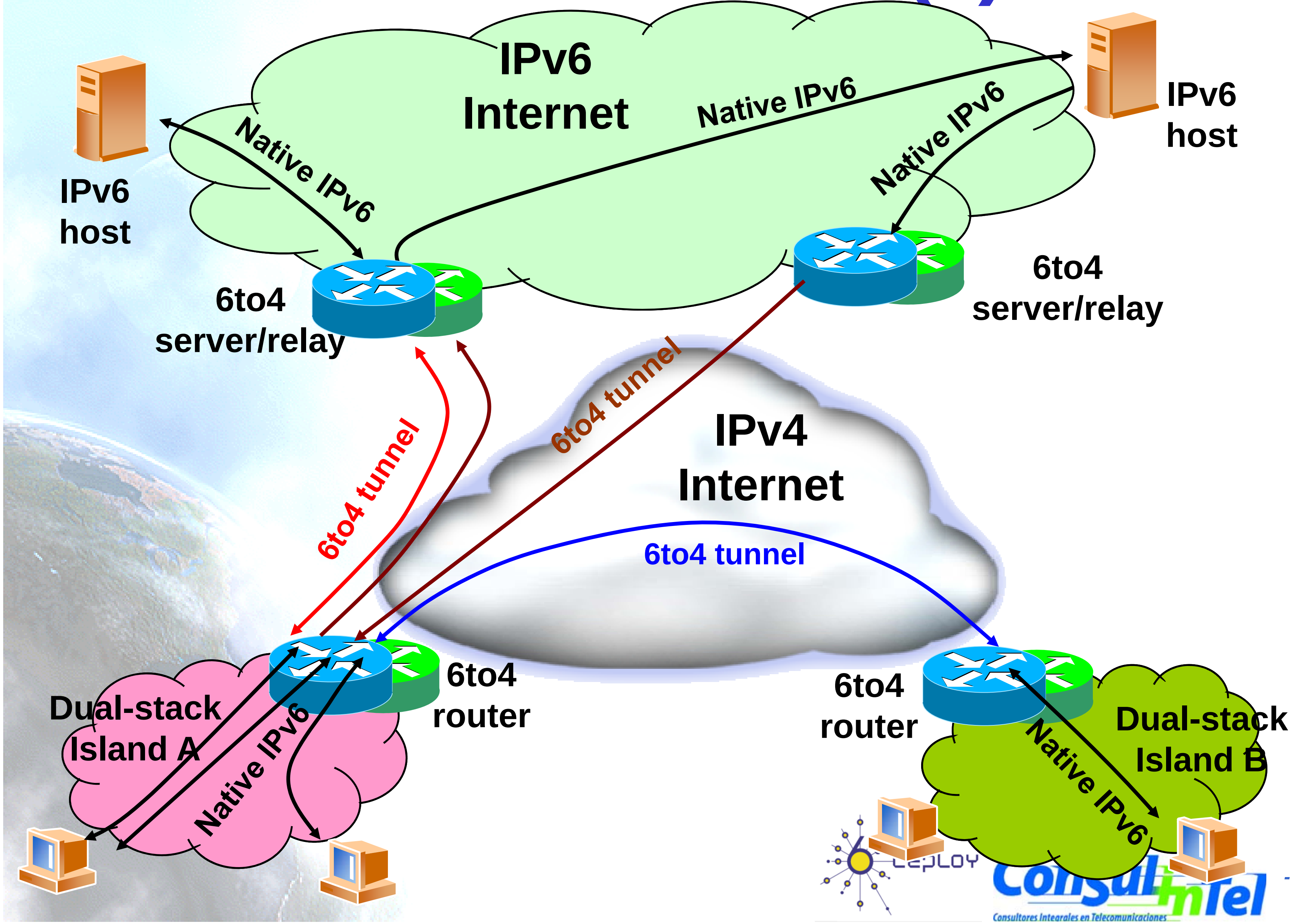
Tunnel Broker [RFC3053]

- The 6in4 tunnels require the manual configuration of the devices involved in the tunnel creation
- To easy the address assignment and the IPv6 tunnel creation, the Tunnel Broker (TB) concept has been developed
  - It is a intermediate host which the end user is connected, usually by using a web browser
- The user asks to the TB the creation of an IPv6 tunnel. The TB assigns to the user an IPv6 address and gives to the user instructions for building the tunnel in the user's side
- The TB also configures the router, which is the TEP for the end user
- In http://www.ipv6tf.org/using/connectivity/test.php exists a list of available TBs
- TSP [TSP] is a special case of TB because it is based on an application installed in the user's host which contacts to the TSP server to built the IPv6 tunnel. However, the concept is similar to the one previously enounced



8.5. 6to4

6to4 Tunnels (1)

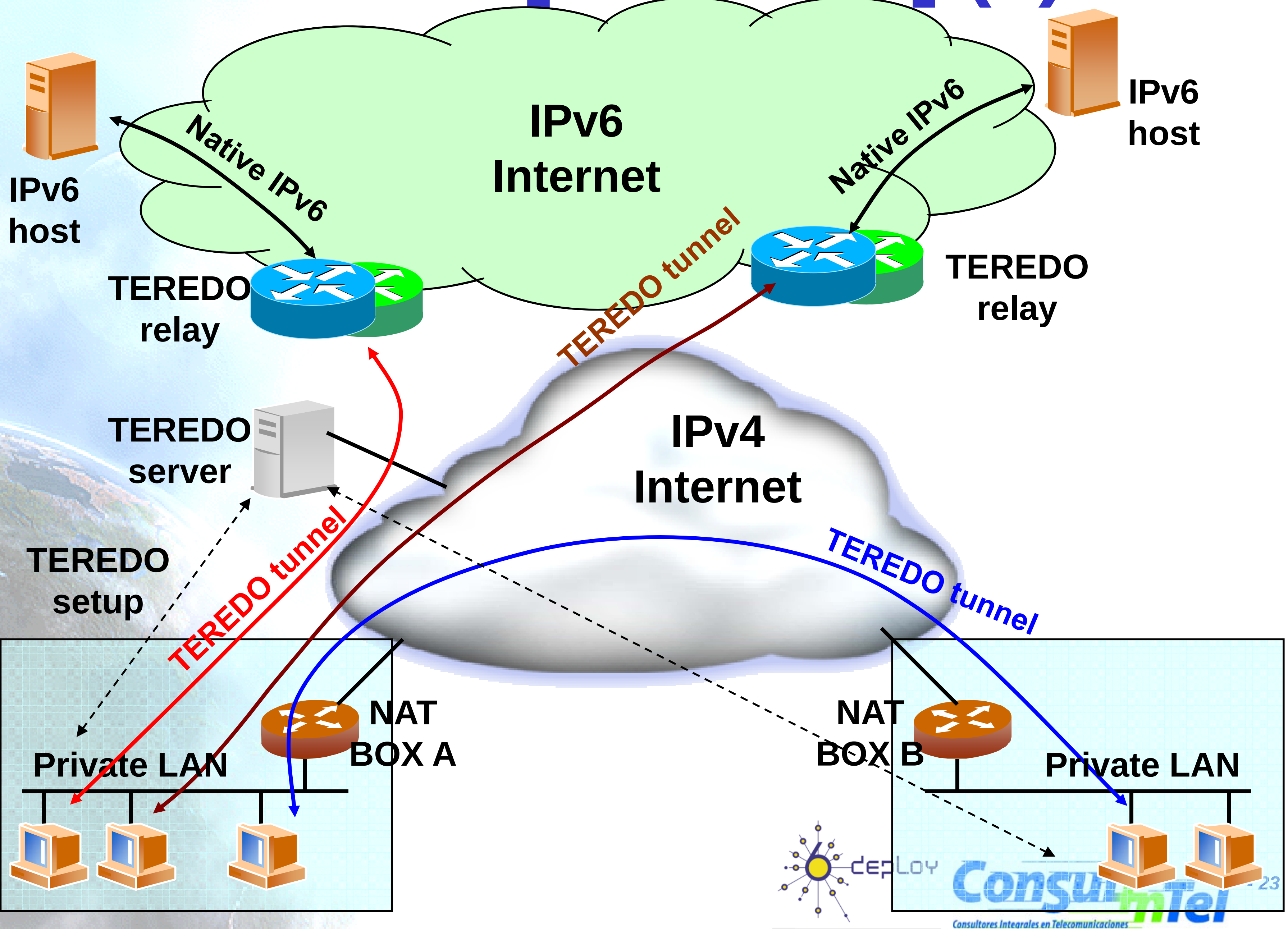


6to4 Tunnels (2)

- Defined on [RFC3056]
- IPv6 packets are encapsulated into IPv4 ones, in a similar way than the 6in4 tunnels
- Differences:
  - The user’s IPv6 address does not depend on the router used to get IPv6 connected but on the public IPv4 used by the user
    - Prefix 2002::/16
  - All the user’s outgoing IPv6 packets are always sent to the same “6to4 relay”. However the user’s incoming IPv6 packets could come from different “6to4 relays”
- IPv4 anycast prefix:
  - 192.88.99.1 [RFC3068]

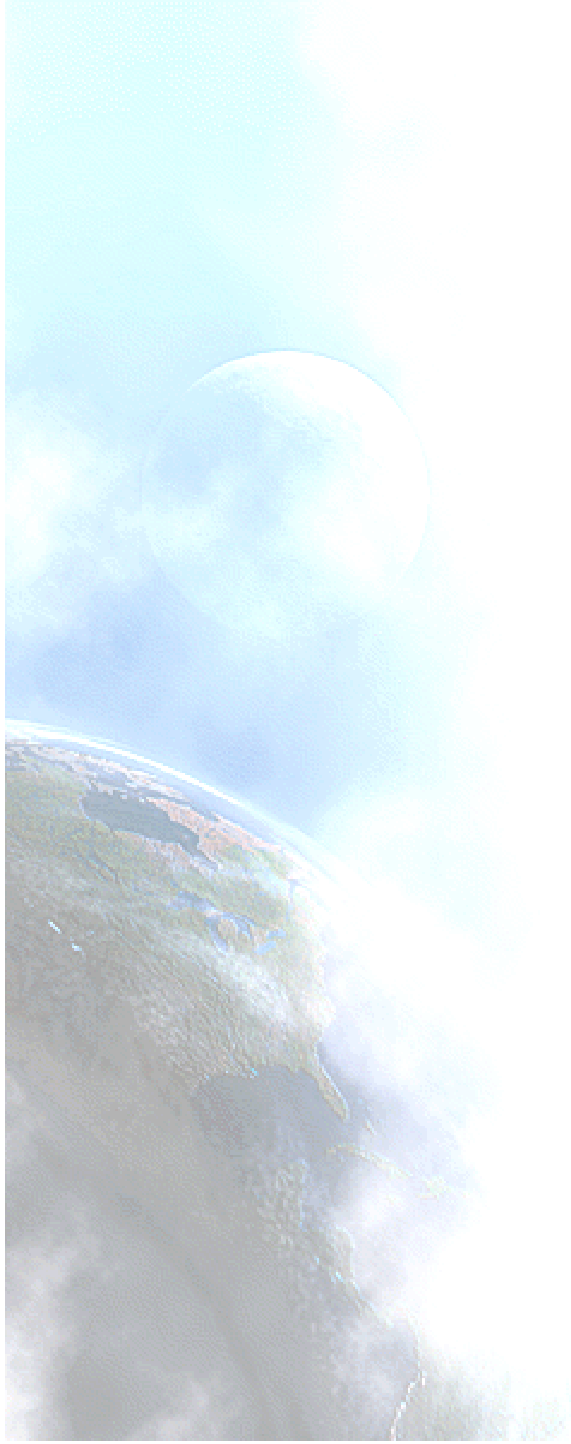
8.6. Teredo

Teredo [RFC4380] (1)



Teredo [RFC4380] (2)

- Teredo [TEREDO] [TEREDOC] is thought for providing IPv6 to hosts that are located behind a NAT box that is not “proto-41 forwarding”
  - It encapsulates the IPv6 packets into UDP/IPv4 packets
- It only works in the following NAT types:
  - Full Cone
  - Restricted Cone
- It does not work in the following NAT type:
  - Symmetric (Solved in Windows Vista)
- Teredo uses different agents to work:
  - Teredo Server
  - Teredo Relay
  - Teredo Client
- The user configures in its host a Teredo Server which provides an IPv6 address from the 2001:0000::/32 prefix and such an address is based on the user’s public IPv4 address and used UDP port
  - If the Teredo Server is also a Teredo Relay, the user has also IPv6 connectivity with any IPv6 hosts
  - Otherwise, the user only has IPv6 connectivity with other Teredo users
- Microsoft currently provides public Teredo Servers for free, but not Teredo Relays



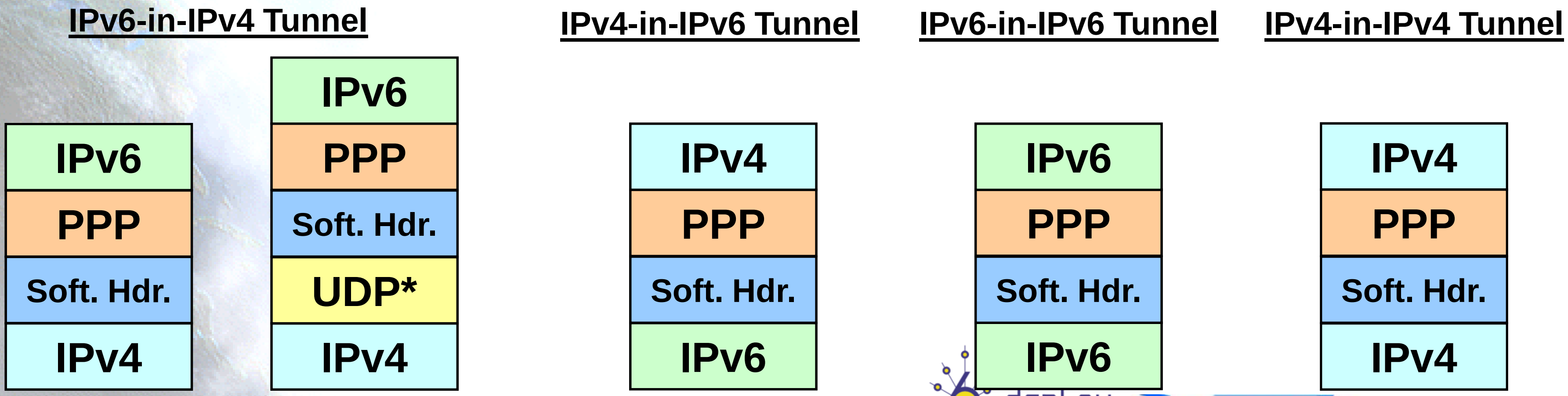
8.7. Softwires

Softwires

- Protocol being discussed within IETF's Softwire WG. Characteristics:
  - "Universal" transition mechanism based on tunnels
    - IPv6-in-IPv4, IPv6-in-IPv6, IPv4-in-IPv6, IPv4-in-IPv4
    - NAT traversal on access networks
    - Provides IPv6 prefix delegation (/48, /64, etc.)
    - User authentication for tunnel creation using AAA infrastructure
    - Possibility of secure tunnels
    - Low overhead of IPv6 packets over the tunnels
    - Supports portable devices with scarce hardware resources
  - Will enable provision of IPv6 connectivity to devices like ADSL routers, mobile phones, PDAs, etc. when no native IPv6 connectivity exists
  - Could provide IPv4 connectivity to devices with IPv6 only connectivity
- Softwires is not a new protocol but the definition of how to use existing protocols in order to provide IPv6 connectivity on IPv4 only networks and vice versa
- It is based on L2TPv2 (RFC2661) and L2TPv3 (RFC3991)

Softwires Encapsulating based on L2TPv2

- Described on draft-ietf-softwire-hs-framework-l2tpv2
- There are two entities:
  - Softwires Initiator (SI): agent who solicits the tunnel
  - Softwires Concentrator (SC): agent who creates the tunnel (tunnel end-point)
- PPP is used to transport IPvx (x=4 or 6) in IPvx (x=4 or 6) packets
  - Optionally PPP packets can be encapsulated on UDP for NAT traversal



IPv6-in-IPv4 Tunnel

IPv6
PPP
Soft. Hdr.
IPv4

IPv6
PPP
Soft. Hdr.
UDP*
IPv4

IPv4-in-IPv6 Tunnel

IPv4
PPP
Soft. Hdr.
IPv6

IPv6-in-IPv6 Tunnel

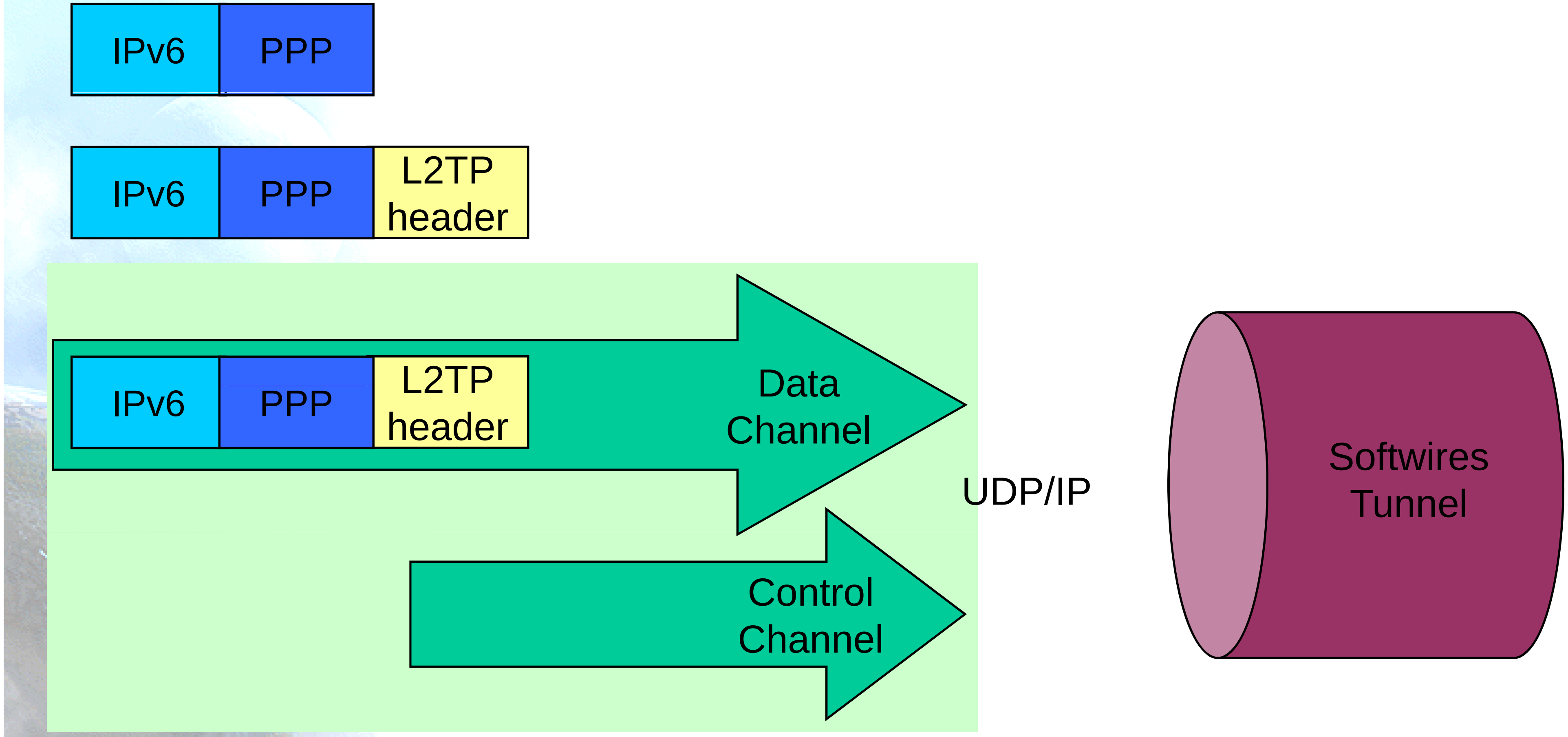
IPv6
PPP
Soft. Hdr.
IPv6

IPv4-in-IPv4 Tunnel

IPv4
PPP
Soft. Hdr.
IPv4

* Optional

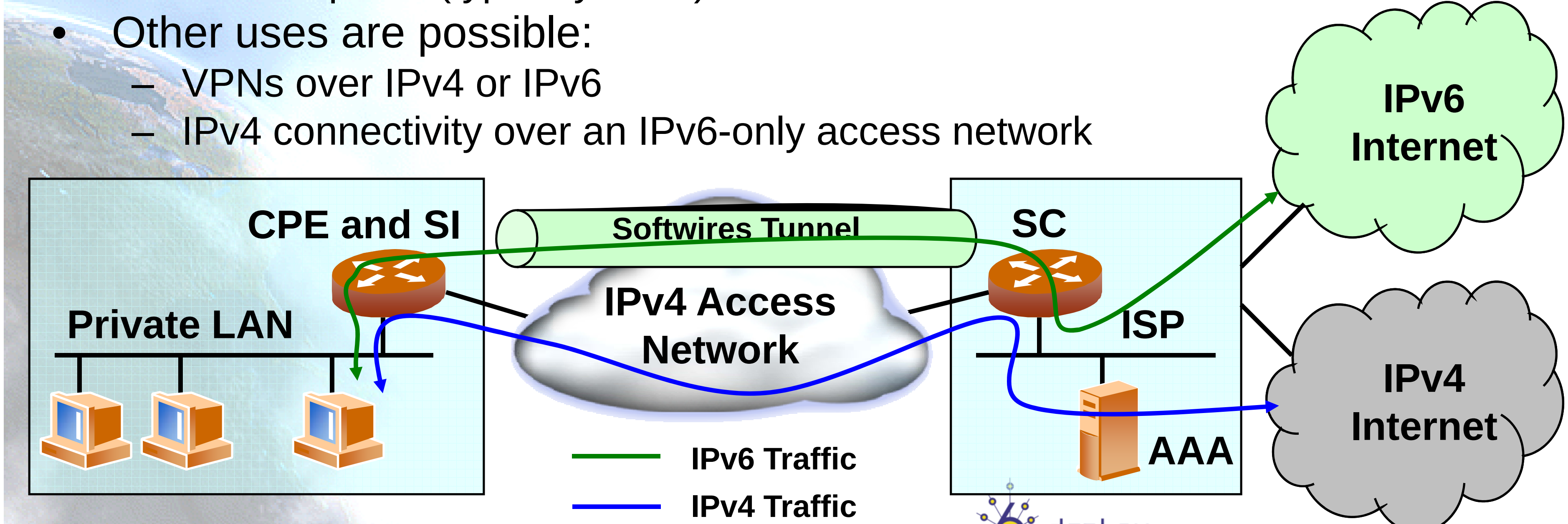
Softwires Based on L2TPv2



- There are a Control and a Data Plane
- PPP is used as an encapsulating protocol

Example of Use

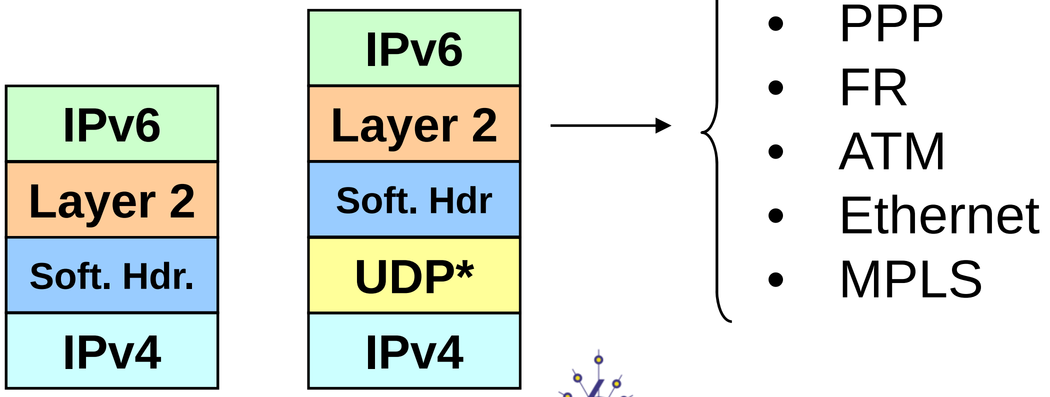
- An expected use of Softwires is for providing IPv6 connectivity to domestic users through an IPv6-only access network
  - The SC is on ISP's network (DSLAM, Aggregation Router, or other device)
  - The SI is on user's network (the CPE or other device)
  - The SC provides IPv6 connectivity to the SI and the SI act as IPv6 router for user networks
  - Prefix delegation (DHCP-PD) is used between the SC and the SI to provide an IPv6 prefix (typically a /48)
- Other uses are possible:
  - VPNs over IPv4 or IPv6
  - IPv4 connectivity over an IPv6-only access network



Softwires Encapsulating Based on L2TPv3

- Same philosophy as with L2TPv2 but with L2TPv3 particularities:
  - Transport over IP/UDP of other layer two protocols different than PPP: HDLC, FR, ATM, Ethernet or MPLS
  - Enhanced header format for better performance in the SC (speeds equal to T1/E1, T3/E3, OC48)
  - Minimum overhead on encapsulated packets (only 4 to 12 extra bytes)
  - Adds EAP as authentication mechanism to CHAP and PAP used in L2TPv2

IPv6-in-IPv4 Tunnel



IPv6	IPv6
Layer 2	Layer 2
Soft. Hdr.	Soft. Hdr
IPv4	UDP*
	IPv4

- HDLC
- PPP
- FR
- ATM
- Ethernet
- MPLS

*** Optional**

8.8. Translation

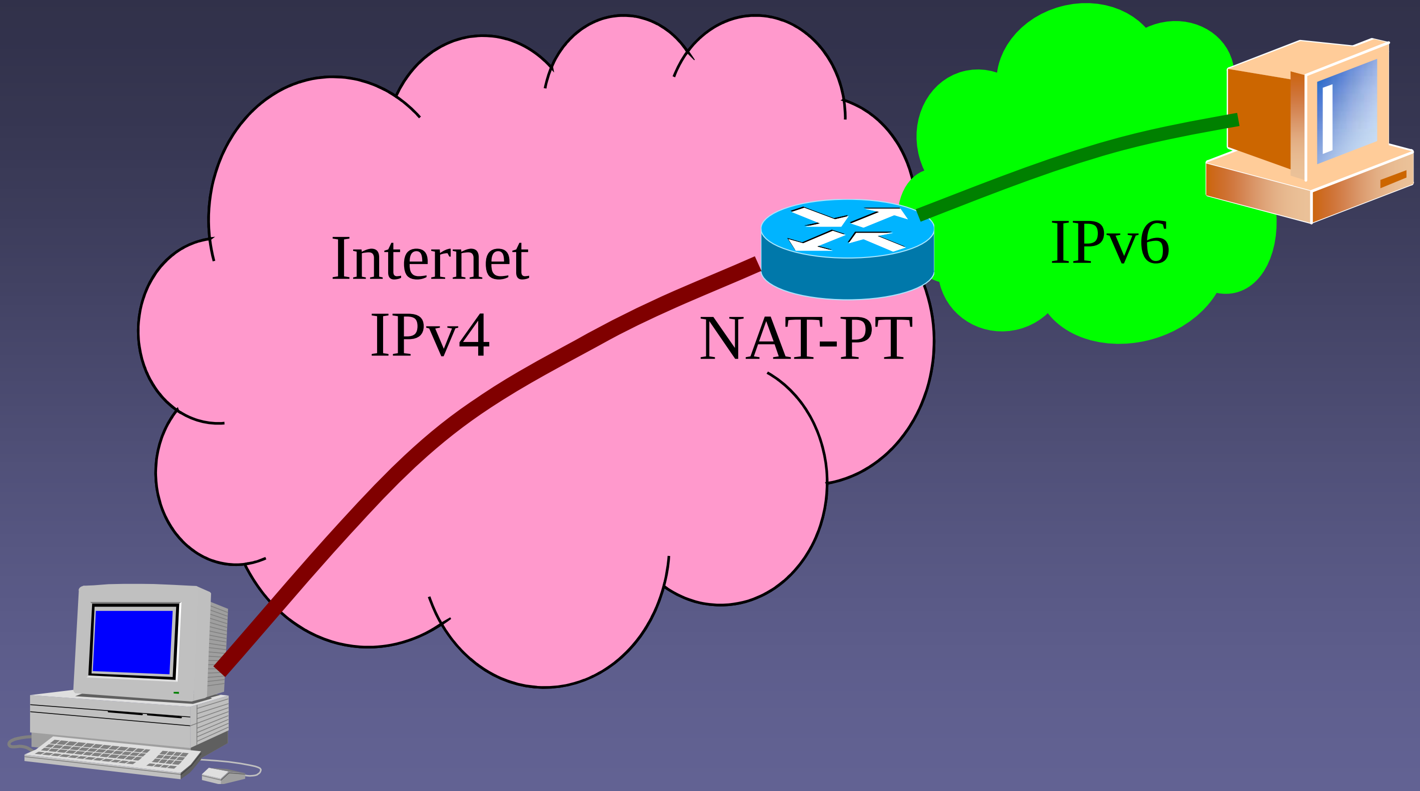
Translation IPv4/IPv6 (1)

- 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

Translation IPv4/IPv6 (Obsolete) (2)

- There are several solutions, but all of them try to translate IPv4 packets into IPv6 and vice-versa
  - [SIT], [BIS], [TRT], [SOCKSv64]
- The commonest is NAT-PT [NATPT], [NATPTIMPL]
  - An intermediate node (router) modifies the IPv4 headers to convert them into IPv6 headers
  - The treatment of the packets is complex
- It is the worst solution because the translation is not perfect and it requires ALGs support, in the same way that IPv4-NATs
  - DNS, FTP, VoIP, etc.

Translation IPv4/IPv6 (Obsolete) (3)



Translation-based mechanism

8.9. Security

Security on Transition Mechanisms

- Security on communications is an objective in actual hostile Internet
- Each new protocol/mechanism introduce new threats and/or opportunities that could be used by malicious nodes
- Transition mechanisms are not an exception and analysis of possible threats and recommendations exist for the most common ones:
  - –6in4 tunnels
  - –6to4 tunnels
  - –Teredo

Security in 6in4 Tunnels (RFC4891) (1)

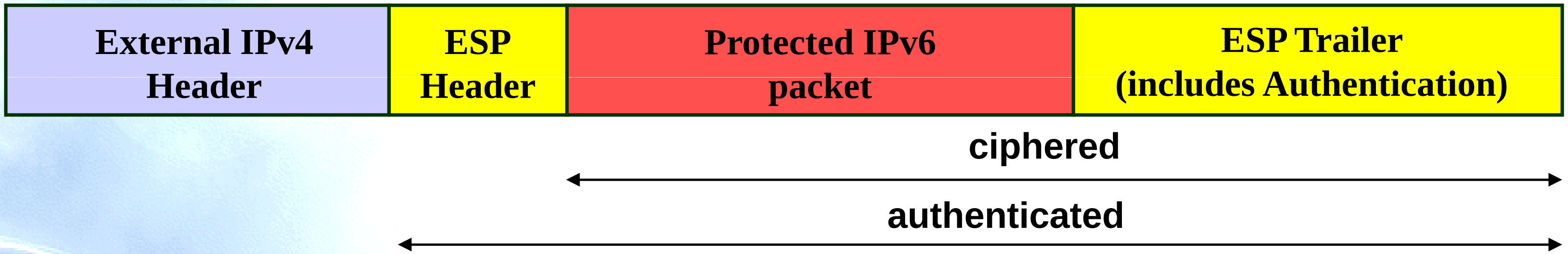
- There are two main threats for 6in4 tunnels:
  - The IPv4 source address of the encapsulating ("outer") packet can be spoofed. Can be minimized by means of:
    - Universal deployment of IPv4 ingress filtering
    - The decapsulator verifies that the IPv4 source address of the packet is the same as the address of the configured tunnel endpoint. The decapsulator may also implement IPv4 ingress filtering, i.e., check whether the packet is received on a legitimate interface.
  - The IPv6 source address of the encapsulated ("inner") packet can be spoofed. Can be minimized by means of:
    - The decapsulator verifies whether the inner IPv6 address is a valid IPv6 address and also applies IPv6 ingress filtering before accepting the IPv6 packet.
- In practice it is necessary to use other method to mitigate these threats, because the measures seen above are not enough and not universally deployed.
  - RFC4891 proposes using IPsec for providing stronger security in preventing these threats and additionally providing integrity, confidentiality, replay protection, and origin protection between tunnel endpoints.
- The tunnel protection must be applied to the three IPv6 traffic types:
  - Global Unicast/Anycast IPv6 Traffic
  - Link-local IPv6 Traffic
  - Multicast IPv6 Traffic

Security in 6in4 Tunnels (RFC4891) (2)

- IPsec can be used in two ways to protect 6in4 tunnels:
  - –Transport Mode (recommended)
  - –Tunnel Mode
- IPsec provides between tunnel endpoints:
  - – Integrity
  - – Confidentiality
  - – Replay protection
  - – Origin protection.
- To use IPsec in 6in4 tunnels it is necessary:
  - – Use an IPsec implementation that follows RFC4301
    - It updates RFC2401with new functionalities needed for 6in4 tunnels
  - – If IKE is used for key management and SA negotiation, it is recommended to use IKEv2 (RFC4306)
- AH can be used as an alternative to ESP
  - – The difference is that AH is able to provide integrity protection for certain fields in the outer IPv4 header and IPv4 options.
  - – As the outer IPv4 header will be discarded at the tunnel end point there is no particular reason to use AH.

IPsec in Transport Mode with 6in4 Tunnels

Transport Mode Example with ESP



External IPv4 Header

ESP Header

Protected IPv6 packet

ESP Trailer (includes Authentication)

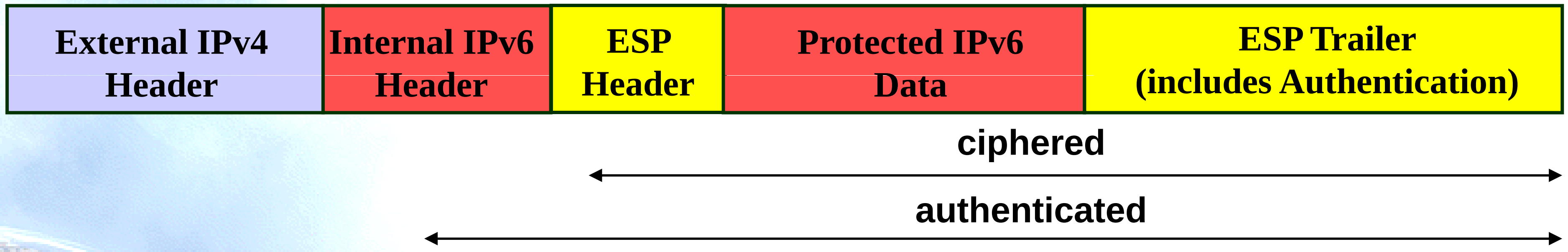
ciphered

authenticated

- Transport Mode
  - – AH or ESP are used depending on the desired security level
  - – The SA is defined by means of (among others):
    - Source IPv4 address
    - Destination IPv4 address
    - Traffic type: IPv6 (protocol 41)
  - – The receiver verifies that the packet came from the right IPv4 endpoint
  - – IPsec in transport mode does not protect against IPv6 source address spoofing of the inner packet. Could be solved by means of ingress filtering on the tunnel endpoint interface.

IPsec in Tunnel Mode with 6in4 Tunnels

Tunnel Mode Example with ESP



External IPv4 Header

Internal IPv6 Header

ESP Header

Protected IPv6 Data

ESP Trailer (includes Authentication)

ciphered

authenticated

- Tunnel Mode:
  - AH or ESP are used depending on the desired security level
  - The SA is defined by means of (among others):
    - Source IPv4 address
    - Destination IPv4 address
  - The receiver verifies that the packet came from the right IPv6 endpoint
  - IPsec in tunnel mode does not protect against IPv4 source address spoofing of the outer packet. This is not a problem because the IPsec SA guarantees that the packet comes from the correct node

Security in 6in4 Tunnels (RFC4891) (3)

- Transport Mode is recommended because there are some drawbacks on using tunnel mode:
  - The majority of IPsec implementations DON'T model the SA in tunnel mode as a network interface
    - It is necessary to indentify all link-local and multicast traffic resulting in a too-long SAs list
    - It is not possible to implement it on 6in4 tunnels between tunnels because the transported traffic is not from a specific prefix but potentially from all the IPv6 Internet
      - There could be in the tunnel packets directed to/from any IPv6 node
- To protect 6in4 tunnels it is recommended to:
  - Manually configure the tunnel
  - Use IPsec in transport mode with ESP (RFC4301)
  - Configure IPv6 ingress filtering on the tunnel interface
  - Used IKEv2 if automatic key management is used

Security in 6to4 Tunnels (RFC3964) (1)

- Threats identified for 6to4 tunnels are based on the specific behavior of 6to4 nodes:
  - – Any 6to4 router should accept 6to4 packets from any other 6to4 router or relay
  - – Any 6to4 router should accept packets from any other native IPv6 router
- There are three general types of threats:
  - – **Denial-of-Service** (DoS) attacks: a malicious node prevents 6to4 communication between the node under attack and other nodes.
  - – **Reflection Denial-of-Service** (DoS) attacks: a malicious node reflects the traffic off unsuspecting nodes to a particular node (node under attack) in order to prevent communication between the node under attack and other nodes using 6to4.
  - – **Service theft**, in which a malicious node/site/operator may make unauthorized use of 6to4 service.
- The attacks that exploit that threats are:
  - – Attacks with Neighbor Discovery (ND) Messages
  - – Spoofing traffic to 6to4 nodes
  - – Reflecting traffic from 6to4 nodes
  - – Local IPv4 broadcast attack
  - – 6to4 Service Theft

Security in 6to4 Tunnels (RFC3964) (2)

- Attacks could be directed to:
  - –6to4 networks.
  - –IPv6 networks.
  - –IPv4 networks.
- Mitigation Methods:
  - – ND messages not allowed on 6to4 interfaces
  - – Ingress filtering on IPv4 and IPv6 networks
  - – Egress filtering of IPv6 6to4 packets if no 6to4 router/relay exist on the network
  - – 6to4 relays should drop packets with 6to4 IPv6 source address that come into a native IPv6 interface
  - – 6to4 relays should drop packets that get into a 6to4 interface with a source address is not 6to4 and/or source IPv4 address does not match with the IPv4 address embedded in the IPv6 address
  - – Limit bandwidth in 6to4 relays
  - – Filter on 6to4 relays all the IPv6 packets which destination address is not 192.88.99.1

TEREDO Security

- Teredo is a special type of tunnel that encapsulates packets in IPv4/UDP in order to traverse NATs.
- As a consequence this mechanism opens a door on the perimeter security devices (firewalls) to some traffic:
  - Legitimate IPv6 Traffic
  - Malicious IPv6 Traffic
- Because of this some traffic is traversing the perimeter security without control. The network/security administrator can't know what kind of IPv6 traffic is going through the network
  - Nowadays, there is no device on the market, able to inspect Teredo traffic in order to apply security policies to the Teredo encapsulated traffic.
- As a result, if Teredo is allowed on the end nodes of a network, it is recommended that:
  - The final node must be protected: last software updates applied, protection tools updated and running (anti-virus, etc.)
  - The network/Security administrator must be warned about the possible vulnerabilities introduced by Teredo (draft-ietf-v6ops-teredo-security-concerns)

Transition References (1)

- [6in4] RFC1933, RFC4213
- [6to4] RFC3056
- [6over4] RFC2529
- [AYIYA] draft-massar-v6ops-ayiya-02
- [BIS] RFC2767
- [DSTM] draft-ietf-ngtrans-dstm-10
- [ISATAP] draft-ietf-ngtrans-isatap-24
- [NATPT] RFC2767
- [NATPTIMPL]
  - –http://www.ipv6.or.kr/english/download.htm ==> Linux 2.4.0
  - –http://www.ispras.ru/~ipv6/index_en.html ==> Linux y FreeBSD
  - –http://research.microsoft.com/msripv6/napt.htm Microsoft
  - –ftp://ftp.kame.net/pub/kame/snap/kame-20020722-freebsd46-snap.tgz ==> KAME snapshot (22.7.2002)
  - –http://ultima.ipv6.bt.com/
- [PRIVACY] RFC3041
- [PROTO41] draft-palet-v6ops-proto41-nat
- [SIIT] RFC2765
- [SILKROAD] draft-liumin-v6ops-silkroad-02

Transition References (2)

- [SOCKSv64] RFC3089
- [SOFTWIRES] draft-ietf-softwire-hs-framework-l2tpv2
- [STATELESS] RFC2462
- [STATEFUL] RFC3315
- [STUN] RFC3489
- [TB] RFC3053
- [TEREDO] RFC4380
- [TEREDOC] http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/teredo.mspx
- [TRT] RFC3142
- [TSP] draft-vg-ngtrans-tsp-01, http://www.hexago.com/index.php?pgID=step1
- [TunAut] RFC1933
- Windows IPv6
  - –http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_ip_v6_add_utils.mspx
  - –http://www.microsoft.com/technet/community/columns/cableguy/cg0902.mspx

IPv6 Tutorial

9. Mobility

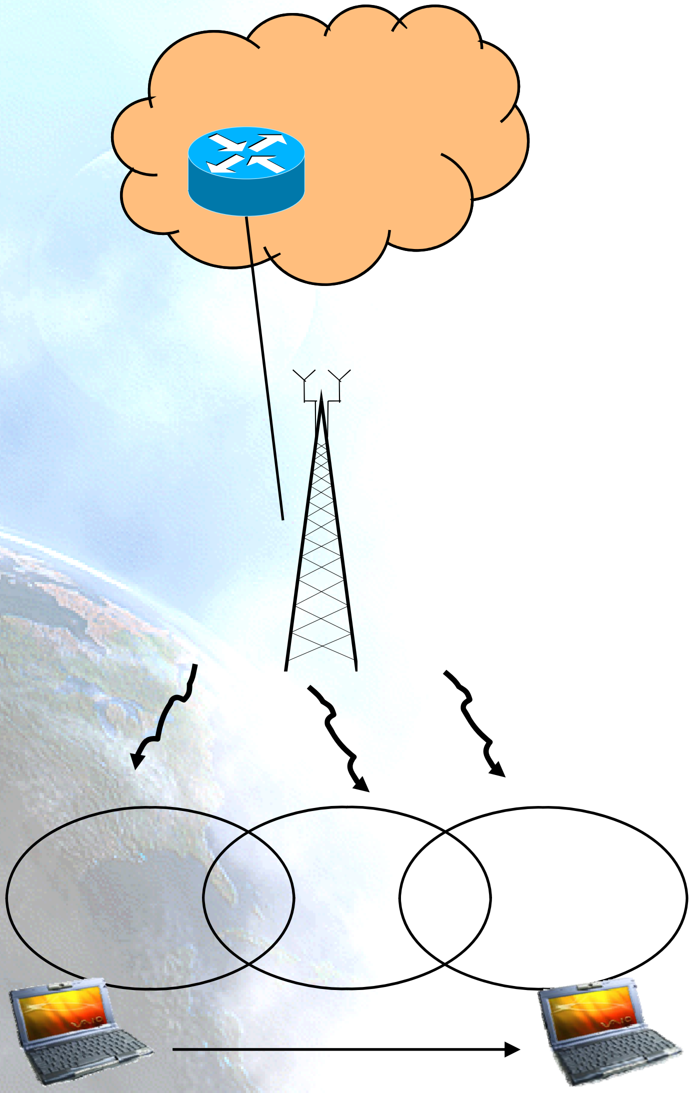
Agenda

9.1. Mobility concepts

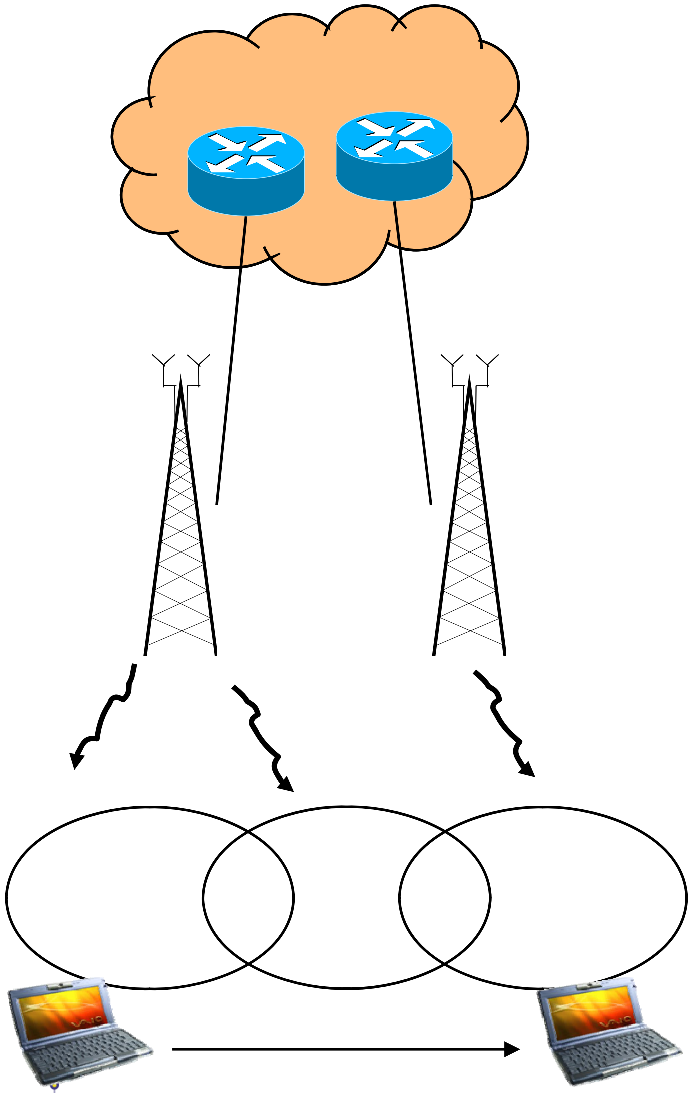
9.2. Mobility in IPv6

9.1. Mobility concepts

Different mobility visions



Mobility layer 2



Mobility layer 3

Mobility in IP layer

- Implications
  - – Communication = f(IP_source, Prt_source, IP_dest., Prt_dest)
  - – If IP address changes, communication is not longer feasible
- Requirements
  - – Compatibility with current applications and systems
  - – No changes in routers
  - – Transparency for applications
  - – ……

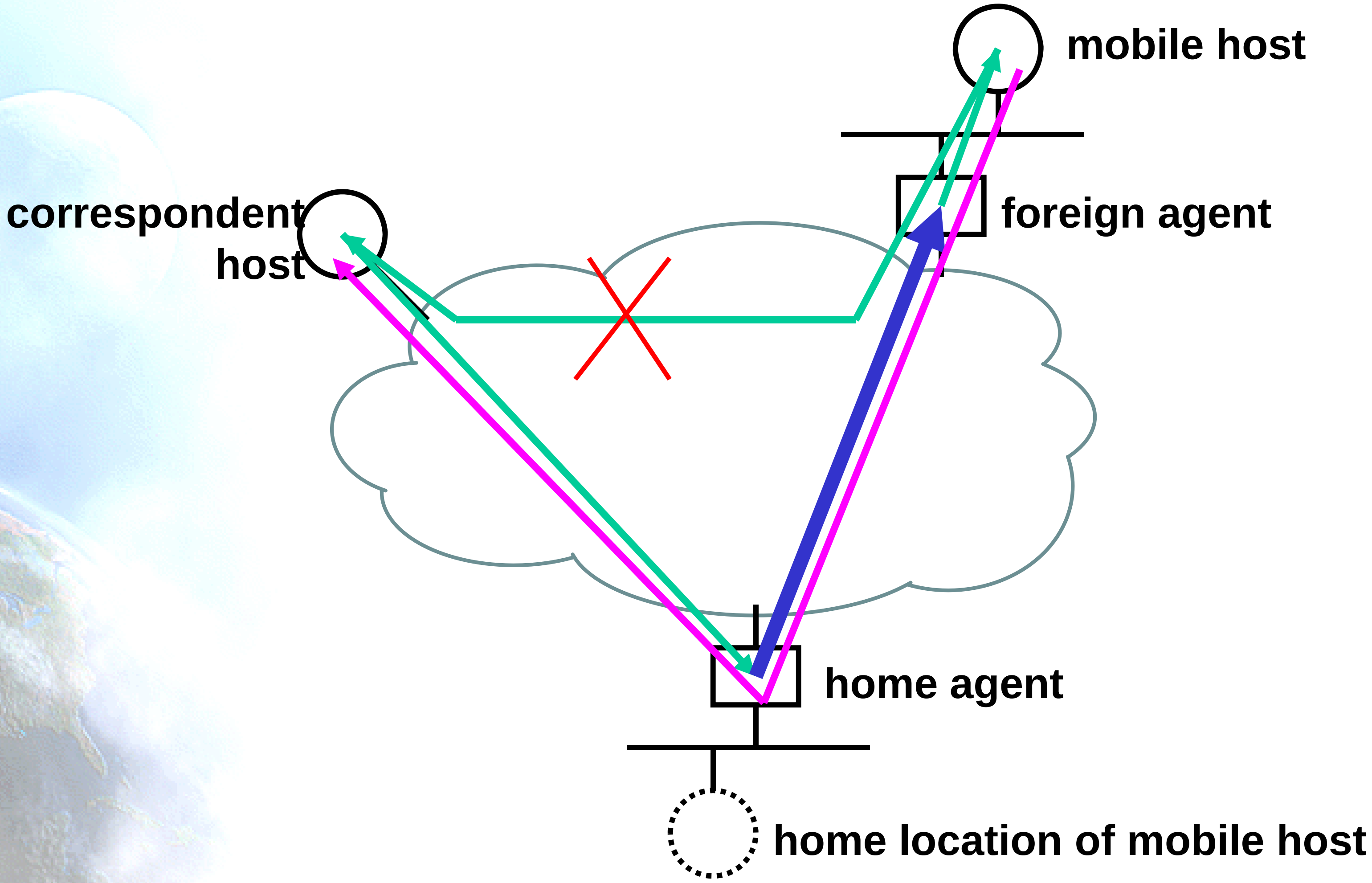
Mobility in IPv4 (1)

- Concepts
  - **Home Agent**: Server in “Home Network” (HN)
  - **Foreign Agent**: Server in foreign network
  - **Mobile Node**: Node which is away from HN
  - **Correspondent Node**: Node communicating to MN
  - **Home Address**: Address from the HN (HoA)
  - **Care of Address**: MN’s address obtained in the foreign network. It’s an address located into the FA, in a virtual interface (CoA)

Mobility in IPv4 (2)

- A MN has one or more HoA
  - they are stable and can be associated to the host name through the DNS
- When MN is in a foreign network, it acquires another IP address
- It registers its new CoA with its HA
- Packets sent to the MN's HoA are intercepted by the HA and then forwarded to the FA by using tunneling
- Packets sent by the MN are delivered in two ways:
  - They are sent to the FA and this forwards them by using the HoA
    - This is an issue if ingress-filtering is implemented into the ISP
  - A tunnel with the HA is created and packets are sent through it

Mobility in IPv4 (3)



Mobility in IPv4 (4)

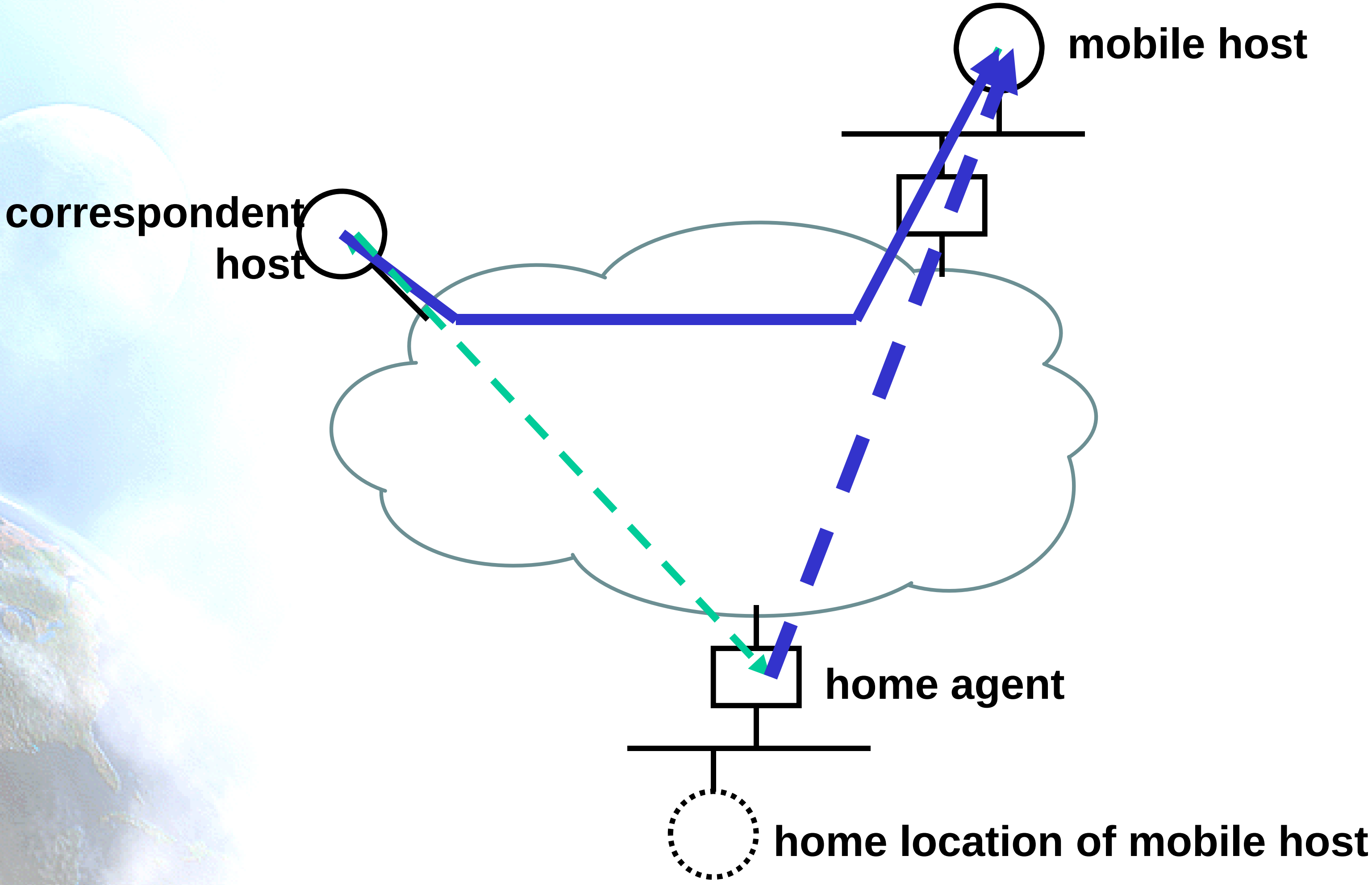
- Security
  - Authentication required
    - FA → HA
    - MN → FA
  - AAA infrastructure is usually used
- Issues with IPv4
  - Scarcity of public IPv4 addresses
    - FAs uses to be located behind routers implementing NAT, so IPv4 packets are modified in transit
  - Complexity and not broadly deployed AAA infrastructures
- Consequence
  - MIPv4 not operating

9.2. Mobility in IPv6

Mobility in IPv6 (1)

- IPv6 has two main features that much help to design a mobility solution
  - Neighbor discovery
  - Autoconfiguration
  - Both of them are used for:
    - Mobile Prefix Discovery: Similar to RS and RA
    - Dynamic HA Address Discovery. More than one HA are possible
- There are many differences to MIPv4, the most remarkable:
  - CoA is setup in the MN rather than the FA
  - There exists no FA
  - Authentication relations are different
    - MN → HA
    - MN → CN
  - ESP is used, so there is no need for AAA
  - Route optimization

Mobility in IPv6 (2)



Mobility in IPv6 (3)

- Route optimization is one of the most remarkable features:
  - At the beginning: CN → HA → MN
  - MN → CN (including the Header Option with its HoA)
    - Alternatively MN → HA → CN by using a tunnel
  - Once communication between CN and MN is setup: CN → MN
- This prevents the fact that HA is one single point-of-failure
- Also, unnecessary delays are prevented when the distance between CN → MN is lower than CN → HA → MN
- Authentication between CN → MN is required

Deploying IPv6 Mobility

- MIPv6 has been standardized 2004
  - It works for manual configurations ➔ Not scalable
- Deploying MIPv6 as network service has several implications
  - To define a scalable mechanism to provide the required parameters for MIPv6 to work without user's manual intervention
    - Bootstrapping: provide HoA, user's cryptographic credentials and HA address
  - To solve network issues to let the MIPv6 service work anywhere
    - HA load balancing
    - IPv4 MIPv6 interworking
    - Firewall traversal
- Most of these issues are being evaluated in the IETF WGs
  - http://www.ietf.org/html.charters/mip6-charter.html (Closed)
  - http://www.ietf.org/html.charters/mext-charter.html
- Also other R&D projects deal with them
  - http://www.ist-enable.eu
  - http://www.nautilus6.org

MIPv6 Standardization

- Basic Mobility Support in IPv6
  - RFC3775 – June 2004
- Use of IPsec to protect MIPv6 signaling between mobile nodes and home agents
  - RFC3776 – June 2004
  - RFC4877 – April 2007 (updates RFC3776)
- Others:
  - RFC4823
  - RFC4225
  - RFC4285
  - RFC4295
  - RFC4887
  - RFC4449
  - RFC4584
  - RFC4640
  - RFC4882

Thanks !

Contact:

- Jordi Palet Martínez (Consulintel): jordi.palet@consulintel.es
- Alvaro Vives Martínez (Consulintel): alvaro.vives@consulintel.es

6DEPLOY Project

http://www.6deploy.org

The IPv6 Portal:

http://www.ipv6tf.org