Transition Mechanisms BIA, TRT & SOCKS

David Fernández (david@dit.upm.es)
Dpto. Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid

dit UPM

Contents

- Bump in the API (BIA)
- Transport Relay Translator (TRT)
- SOCKS-based IPv6/IPv4 Gateway Mechanism (SOCKS64)

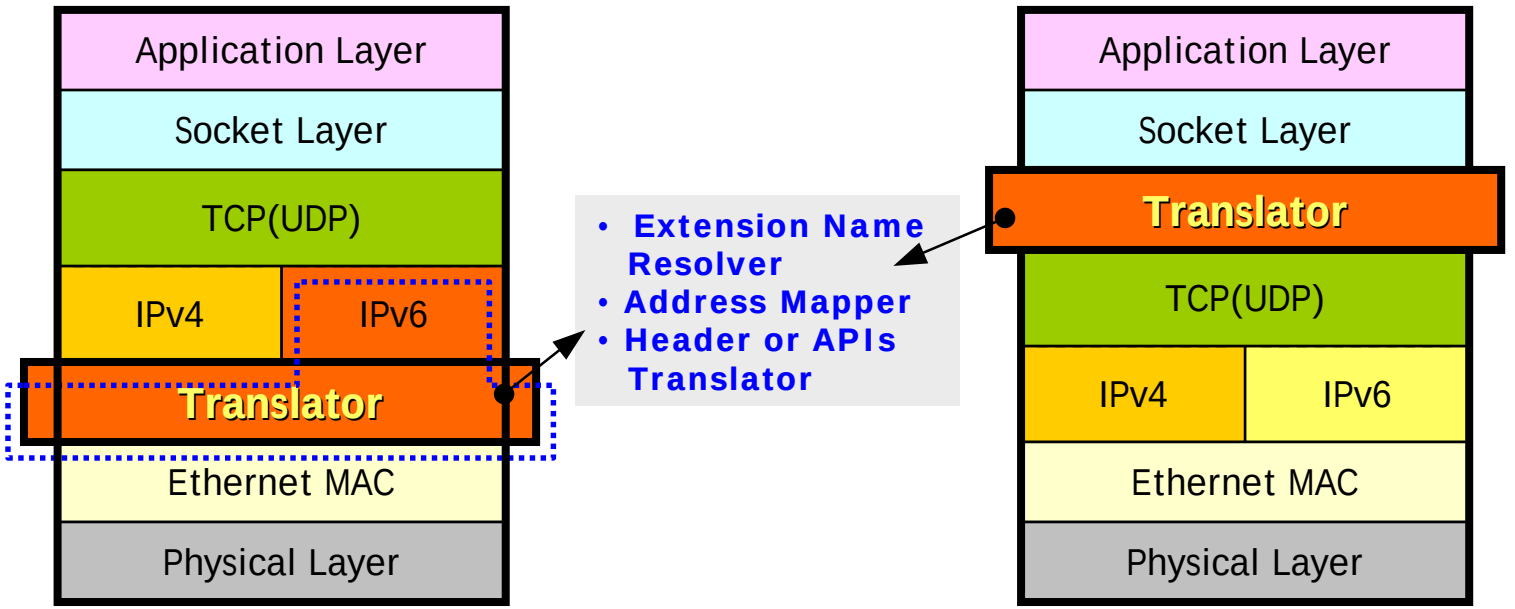
dit
UPM

Bump-in-the-API (BIA)

- Internet Draft: Dual Stack Hosts using "Bump-in-the-API" (BIA). February 2002.
- Defines a HOST transition mechanism to allow existing IPv4 applications in dual stack nodes to communicate using IPv6
- Very similar to BIS, but the translation is made between IPv4 and IPv6 APIs (not header translation involved). Besides:
  - BIS is for systems with no IPv6 stack
  - BIA is for dual stack systems
- Both are specially useful at the first stages of migration, when most applications are not migrated

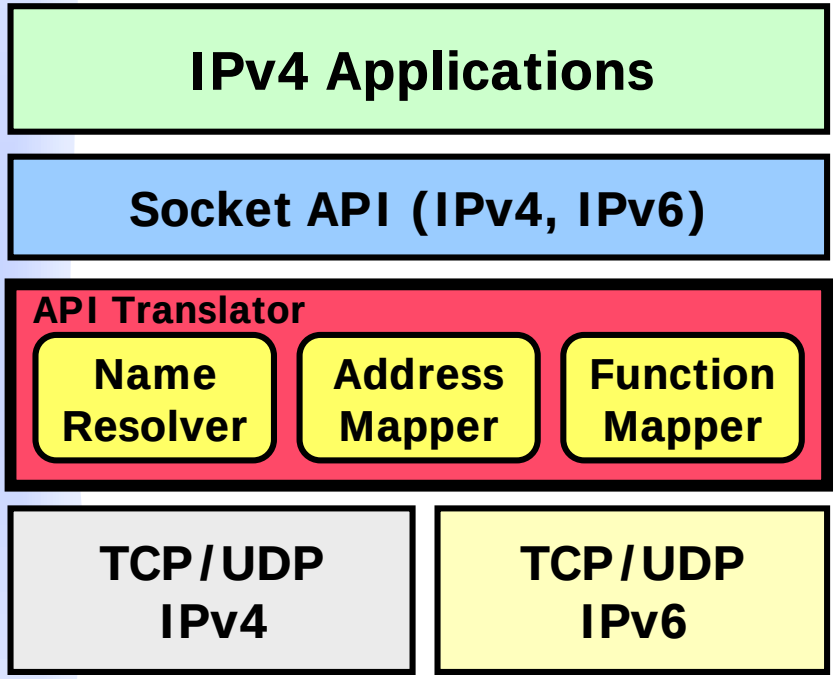
dit UPM

BIS vs. BIA



Source: **Seungyun Lee, ETRI, http://www.krv6.net/bia/**

Bump-in-the-API Architecture

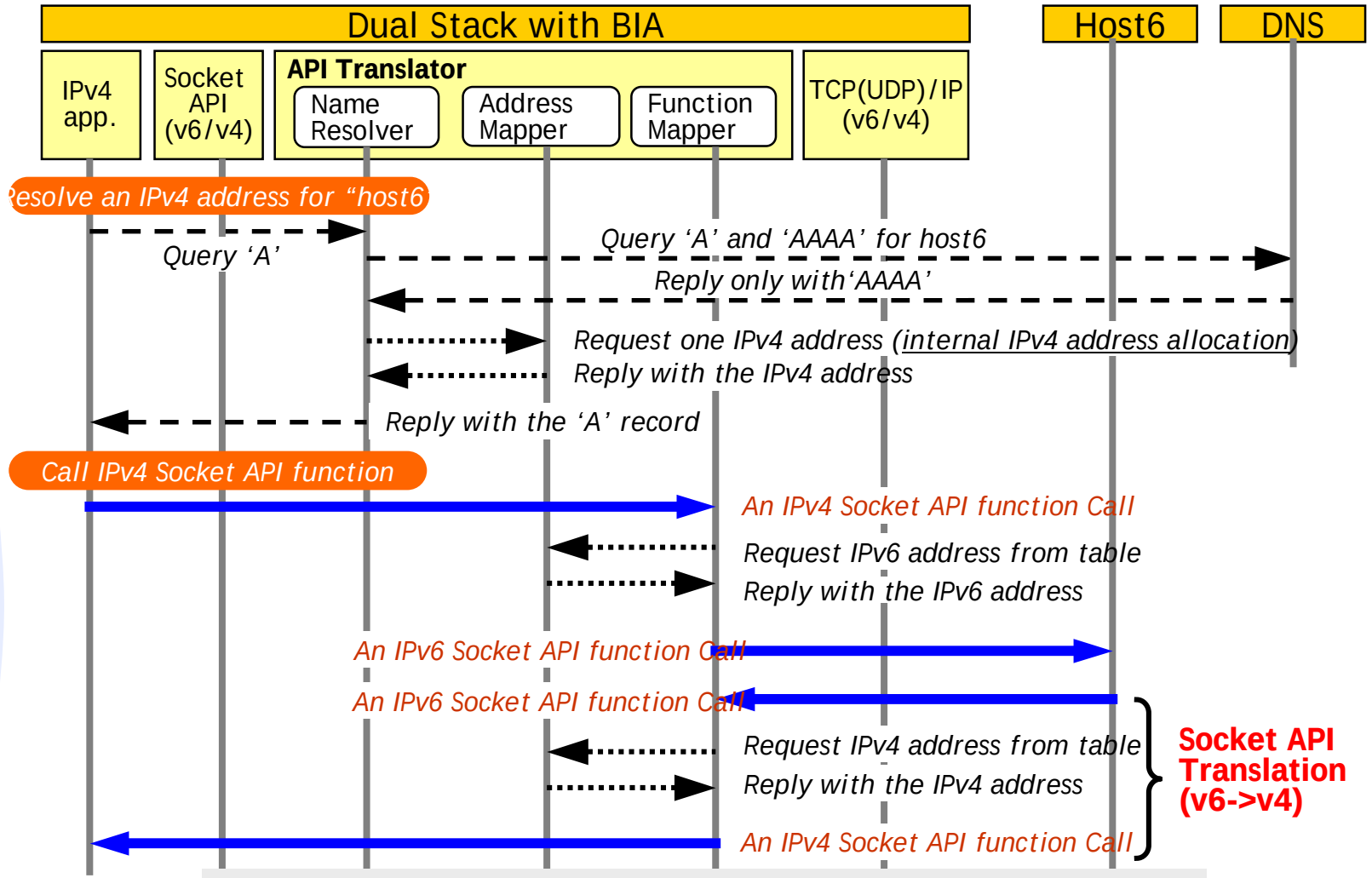


BIA Architecture

- Name Resolver: intercepts DNS request primitives to ask for AAAA records in adition to A records
- Address Mapper: maintains and internal table of mappings between IPv4 and IPv6 addresses
- Function Mapper: translates IPv4 socket API functions into IPv6 socket API functions and viceversa
- Name Resolver and Address Mapper functions are the same than in BIS

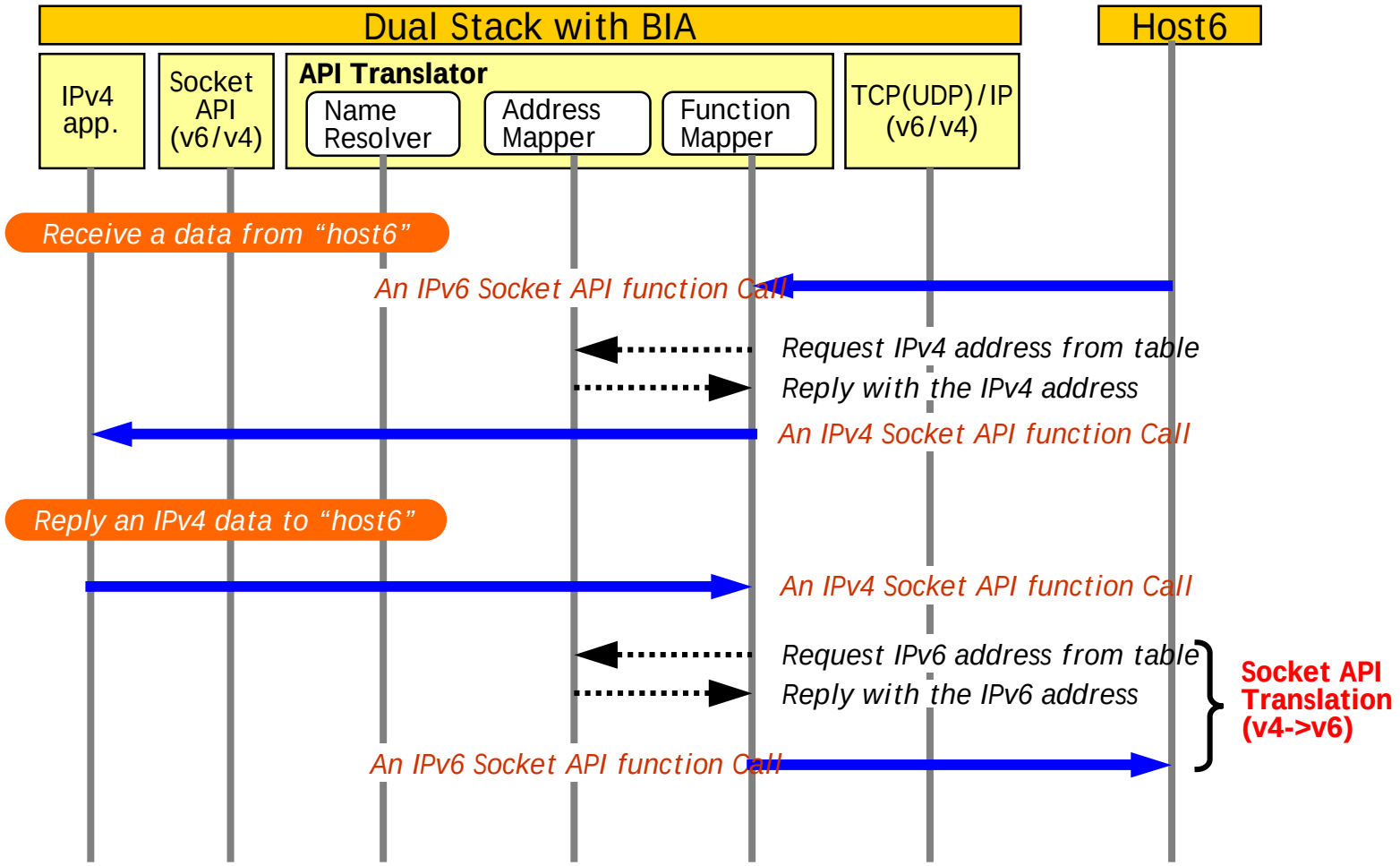
dit UPM

Operation example of BIA (I)



Source: **Seungyun Lee, ETRI, http://www.krv6.net/bia/**

Operation example of BIA(II)



Source: **Seungyun Lee, ETRI, http://www.krv6.net/bia/**

BIA Summary

- Translation based mechanism for hosts
- Advantages:
  - Useful for early adopters with applications not yet migrated or applications without source code available
- Disadvantages:
  - Only basic socket API functionality supported (in general, does not work with IPv4 or IPv6 options)
  - Must incorporate application specific translation algorithms (as other translation mechanisms)

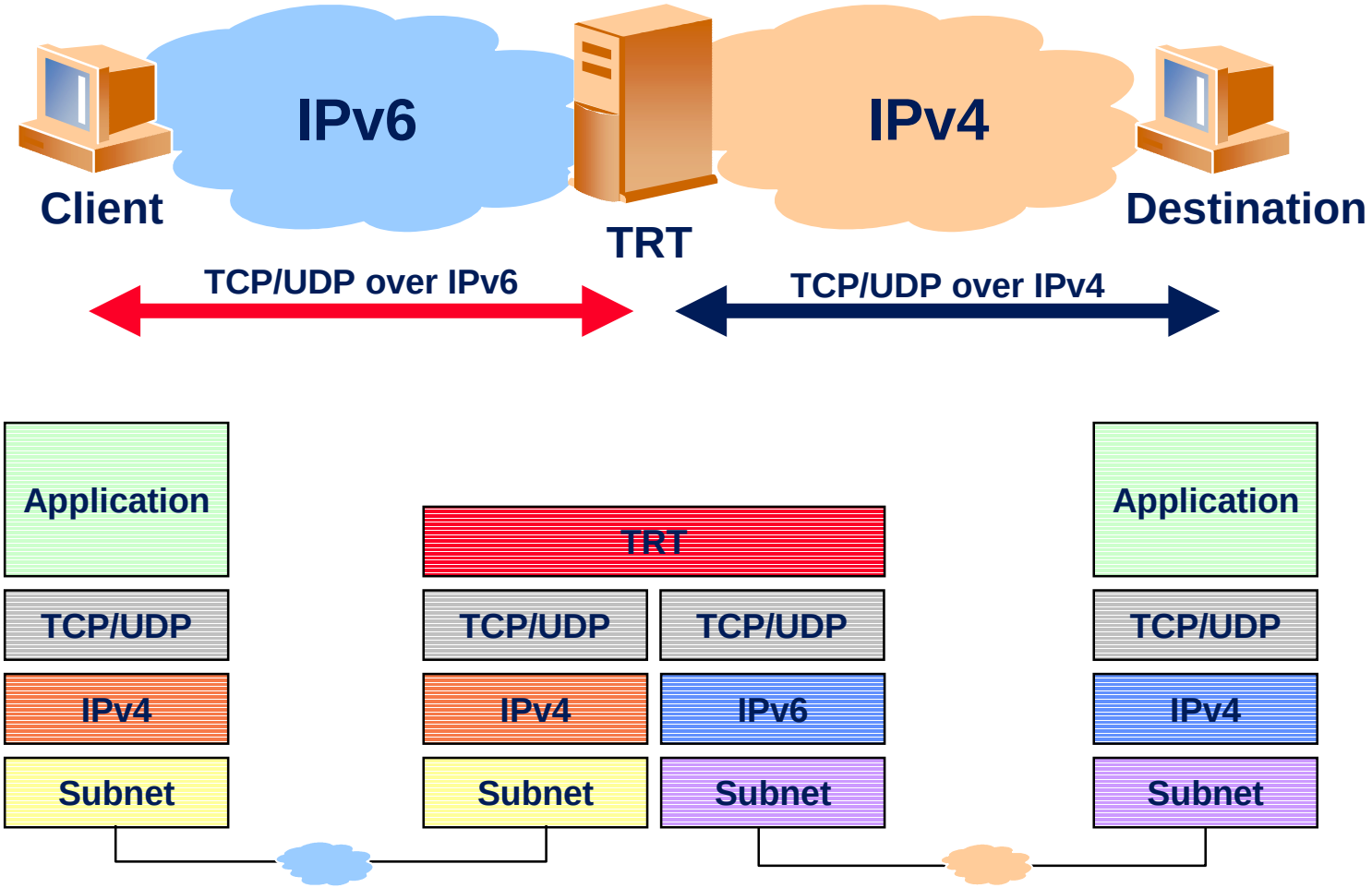
dit UPM

Transport Relay Translator (TRT)

- RFC 3142: An IPv6-to-IPv4 Transport Relay Translator. June 2001.
- Allows IPv6-only hosts to exchange (TCP, UDP) traffic with IPv4-only hosts.
- Characteristics:
  - Conversion between IPv6 and IPv4 made at transport layer in TRT system
    - TRT "captures" transparently TCP segments and UDP datagrams
  - IPv4 addresses embedded in IPv6 addresses
    - IPv6 unicast prefix reserved for TRT operation
  - TRT is a stateful system

dit UPM

TRT Operation



dit UPM

TRT Detailed Operation

- An IPv6 prefix, for example fec0:0:0:1::/64, is reserved for address mapping
- When an IPv6 host wants to make a TCP connection with an IPv4 host, for example 10.0.0.1, tries an IPv6 connection to fec0:0:0:1::10.1.1.1
- Packets to fec0:0:0:1::/64 prefix are routed through TRT system
- TRT captures the packets, ends the IPv6 connection and opens a new IPv4 connection to the address contained in the lower 32 bits of the original IPv6 address
- Similar procedure is used for UDP communications

dit
UPM

TRT Summary

- Translation based mechanism for sites
- Advantages:
  - No modifications to IPv6-only or IPv4-only nodes needed
  - No fragmentation or path MTU issues
- Disadvantages:
  - Unidirectional (from IPv6 to IPv4)
    - Address mapping from IPv4 to IPv6 difficult
  - TRT is a stateful system
    - A transport layer connection must go through the same TRT (single point of failure)
  - Special code needed to relay NAT unfriendly protocols

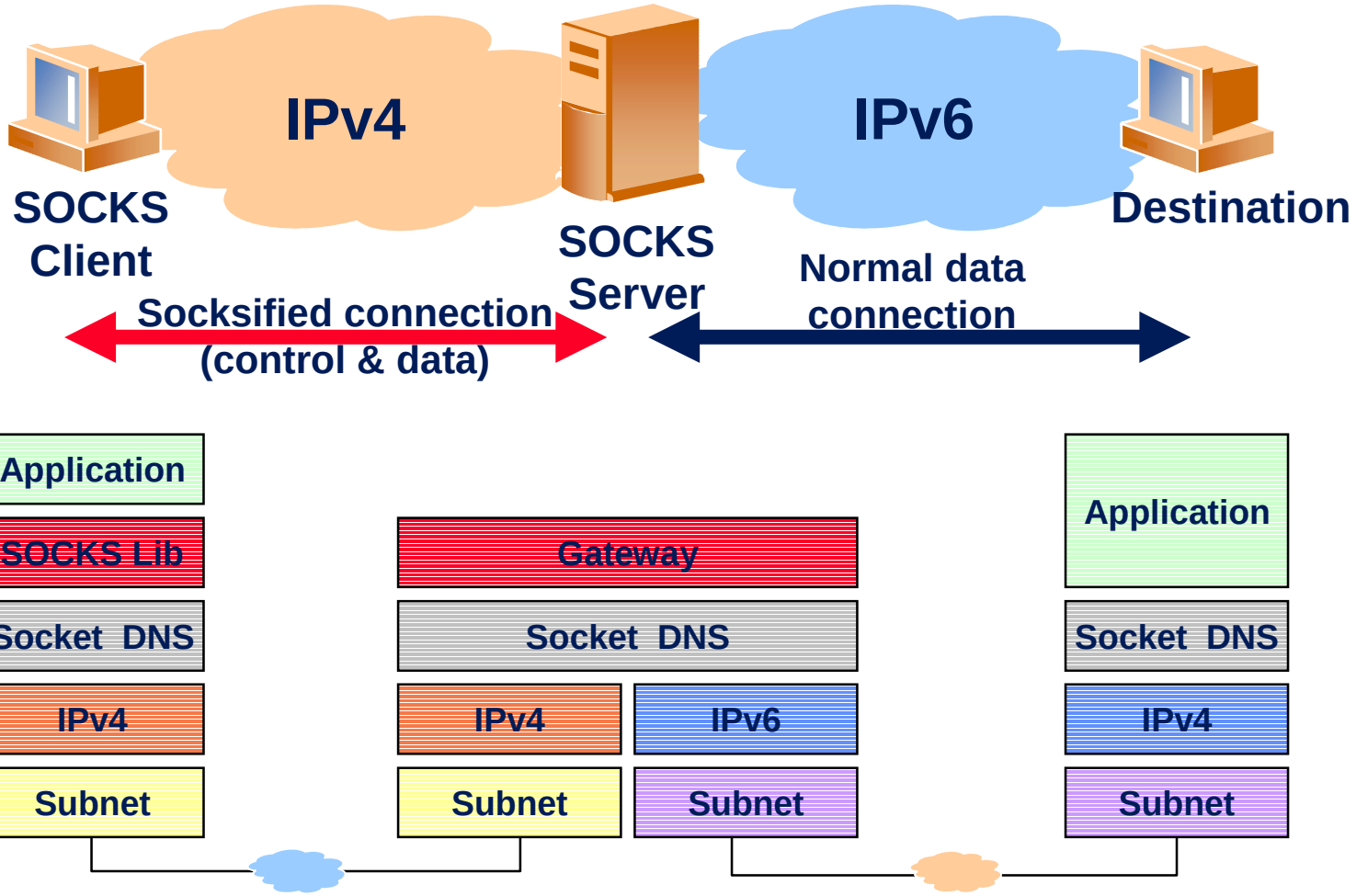
dit UPM

SOCKS

- RFC 3089: A SOCKS-based IPv6/IPv4 Gateway Mechanism. April 2001.
- Defines an IPv6/IPv4 gateway mechanisms based on SOCKS v5 (RFC 1928)
  - Allows IPv4 hosts to communicate with IPv6 hosts (or viceversa)
  - It makes use of a SOCKS server to relay two terminated IPv4 and IPv6 connections at the application layer
  - Clients must be “socksified”, i.e., a SOCKs library must be installed (although applications need no modifications)

dit UPM

SOCKS Operation



dit UPM

SOCKS Detailed Operation

- Client application initiates a connection to an external node using a name (FQDN)
- Client Socks library intercepts name resolution request and initiates an authenticated TCP connection to SOCKS server (port 1080)
- SOCKS server returns to client a “fake IPv4 address”
- SOCKS server initiates a connection with the remote node and works as a relay between client and remote node. Besides it makes the IPv6/IPv4 translation based on SIIT.
- Between client and server, packets are sent over the socksified connection.

dit UPM

SOCKS Summary

- Translation based mechanism for sites
- Advantages:
  - Useful for limited deployment of IPv6 access
  - Easy installation for corporate networks already running SOCKS as firewalling mechanism (no modifications to SOCKSv5 needed)
  - Provides security (AAA in the access to service)
- Disadvantages:
  - Installation of SOCKSv5 library in clients
  - Supports only client initiated connections
  - Must incorporate application specific translation algorithms (as other translation mechanisms)

dit UPM

Implementations

- BIA:
  - ETRI & i2Soft. http://www.krv6.net/bia/
- TRT
  - KAME. http://www.kame.net
  - Portable Transport Relay Translator Daemon (pTRTd). http://v6web.litech.org/ptrtd/
- SOCKS:
  - NEC. http://www.socks.nec.com
  - Fujitsu. ftp://ftp.kame.net/pub/kame/misc

dit UPM

References

- S. Lee et al. “Dual Stack Hosts using "Bump-in-the-API" (BIA)”. Internet Draft. February 2002. (draft-ietf-ngtrans-bia-03.txt)
- J. Hagino, K. Yamamoto. “An IPv6-to-IPv4 Transport Relay Translator”. RFC 3142. June 2001.
- H. Kitamura. “A SOCKS-based IPv6/IPv4 Gateway Mechanism”. RFC 3089. April 2001.

dit UPM