Transition Mechanisms
BIA, TRT & SOCKS

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

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

- 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.
**BIS vs. BIA**

**BIS**

- Application Layer
- Socket Layer
- TCP(UDP)
- IPv4
- IPv6
- Ethernet MAC
- Physical Layer

**BIA**

- Application Layer
- Socket Layer
- TCP(UDP)
- IPv4
- IPv6
- Ethernet MAC
- Physical Layer

**Translator**

- Extension Name Resolver
- Address Mapper
- Header or APIs

Bump-in-the-API Architecture

- **Name Resolver:** intercepts DNS request primitives to ask for AAAA records in addition 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 vice versa.
- **Name Resolver and Address Mapper functions are the same than in BIS.**
Operation example of BIA (I)

Dual Stack with BIA

[Diagram showing the interactions between IPv4 and IPv6 systems using BIA (Dual Stack with BIA)]

IPv4 app.  Socket API (v6/v4)  API Translator  AddressMapper  FunctionMapper  TCP(UDP)/IP (v6/v4)

Resolve an IPv4 address for “host6”

Query ‘A’

Call IPv4 Socket API function

Query ‘A’ and ‘AAAA’ for host6

An IPv4 Socket API function Call

Request one IPv4 address (internal IPv4 address allocation)

Request IPv6 address from table

Socket API translation (v6->v4)

Reply with the IPv4 address

Reply with the IPv6 address

Reply with the ‘A’ record

Query ‘AAAA’

Reply only with ‘AAAA’

An IPv6 Socket API function Call

Request IPv4 address from table

An IPv6 Socket API function Call

An IPv6 Socket API function Call

An IPv6 Socket API function Call

Operation example of BIA(II)

Dual Stack with BIA

IPv4 app. | Socket API (v6/v4) | API Translator | TCP(UDP)/IP (v6/v4) | Host6
---|---|---|---|---

| Name Resolver | Address Mapper | Function Mapper |

**Receive a data from “host6”**
- An IPv6 Socket API function Call
- Request IPv4 address from table
- Reply with the IPv4 address
- An IPv4 Socket API function Call

**Reply an IPv4 data to “host6”**
- An IPv6 Socket API function Call
- Request IPv6 address from table
- Reply with the IPv6 address
- An IPv4 Socket API function Call

Socket API Translation (v4→v6)

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)
**Transport Relay Translator (TRT)**

- 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
TRT Operation

IPv6

Client

TCP/UDP over IPv6

IPv4

TRT

TCP/UDP over IPv4

IPv4

Destination

Application

TCP/UDP

IPv4

Subnet

TCP/UDP

IPv4

Subnet

TCP/UDP

IPv6

Subnet

TCP/UDP

IPv4

Subnet

TCP/UDP

IPv4

Subnet
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.
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
SOCKS

- Defines an IPv6/IPv4 gateway mechanisms based on SOCKS v5 (RFC 1928)
  - Allows IPv4 hosts to communicate with IPv6 hosts (or vice versa)
  - 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)
**SOCKS Operation**

SOCKS Client -> IPv4 -> SOCKS Server -> IPv6 -> Destination

- **Socksified connection** (control & data)
- **Normal data connection**

**Diagram Details:**
- Application
- SOCKS Lib
- Socket DNS
- IPv4
- Subnet
- Gateway
- Socket DNS
- IPv4
- IPv6
- Subnet
- Subnet

© 2002, DIT-UPM
IPv6 Tutorial – Transition Mechanisms: BIA, TRT & SOCKS

14
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.
SOCKS Summary

- **Translation** based mechanism for sites

- **Advantages:**
  - Useful for limited deployment of IPv6 access
  - Easy installation for corporate networks already running SOCKS as firewallsing 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)
Implementations

- **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
References