Digital IPv6 Implementation Experience

Jim Bound
bound@zk3.dec.com
Digital Equipment Corporation

TCP/IP EXPO
San Jose Convention Center
Aug 8, 1995

Topics

> IPv6 Advanced Development Prototype Goals

> IPv6 Architecture and Principles Used

> IPv6 Implementation Overview

> IPv6 Industry Implementations

Goals and Objectives

> Test the IPv6 IETF proposed specifications with running code.

> Interoperability testing with other industry implementations.

> Understand the changes to the existing TCP/IP software.

> Add new IPv6 features into the TCP/IP software paradigm.

> Test the market for future product directions.

> Remain compatible with BSD UNIX 4.4 TCP/IP Stack.

> Performance greater than IPv4, but never less.

> Cannot break any IPv4 applications.

IPng Prototype History

> SIP 8 Octet proposal on Digital UNIX and OpenVMS, based on INRIA SIP 8 Octet public domain code base.

> TUBA (TCP/UDP over CLNP) on Digital UNIX.

> SIPP 16 Octet proposal on Digital UNIX.

> IPv6 on Digital UNIX and in process on OpenVMS.

Why is there a need for an IPv6 solution?

> Exhaustion of the IPv4 Address Space

> Routing tables are growing too fast.

> Utility of current address structure is not robust enough.

> Emerging Technology requires new features.

IPv6 Architecture and Principles

> IPv6 APIs must not break IPv4 APIs

> IPv6-IPv4 Address Families tightly coupled.

> IPv6-IPv4 Socket Layer loosely coupled.

> IPv6-IPv4 Transport Layer tightly coupled.

> IPv6-IPv4 Network Layer loosely coupled.

> IPv6 Neighbor Discovery processing at the Network Layer.

> IPv6 Autoconfiguration permeates all Implementation Layers.

> IPv6 Security premeates all Implemenation Layers.

> IPv6 Autogegistration tightly coupled to Autoconfiguration.

> IPv6 use of Multicast supercedes Broadcast.

> IPv6-IPv4 Transition must provide wide range of solutions.

IPv6 Base Specifications to Test

> Base Protocol Specification

> Addressing Architecture

> Internet Control Message Protocol v6 (ICMPv6)

> Neighbor Discovery

> Stateless and Stateful Address Configuration

> Transition Mechanisms

> Security Architecture and Payloads

> DNS IPv6 Records

> DNS Autoregistration

> BSD Sockets APIs

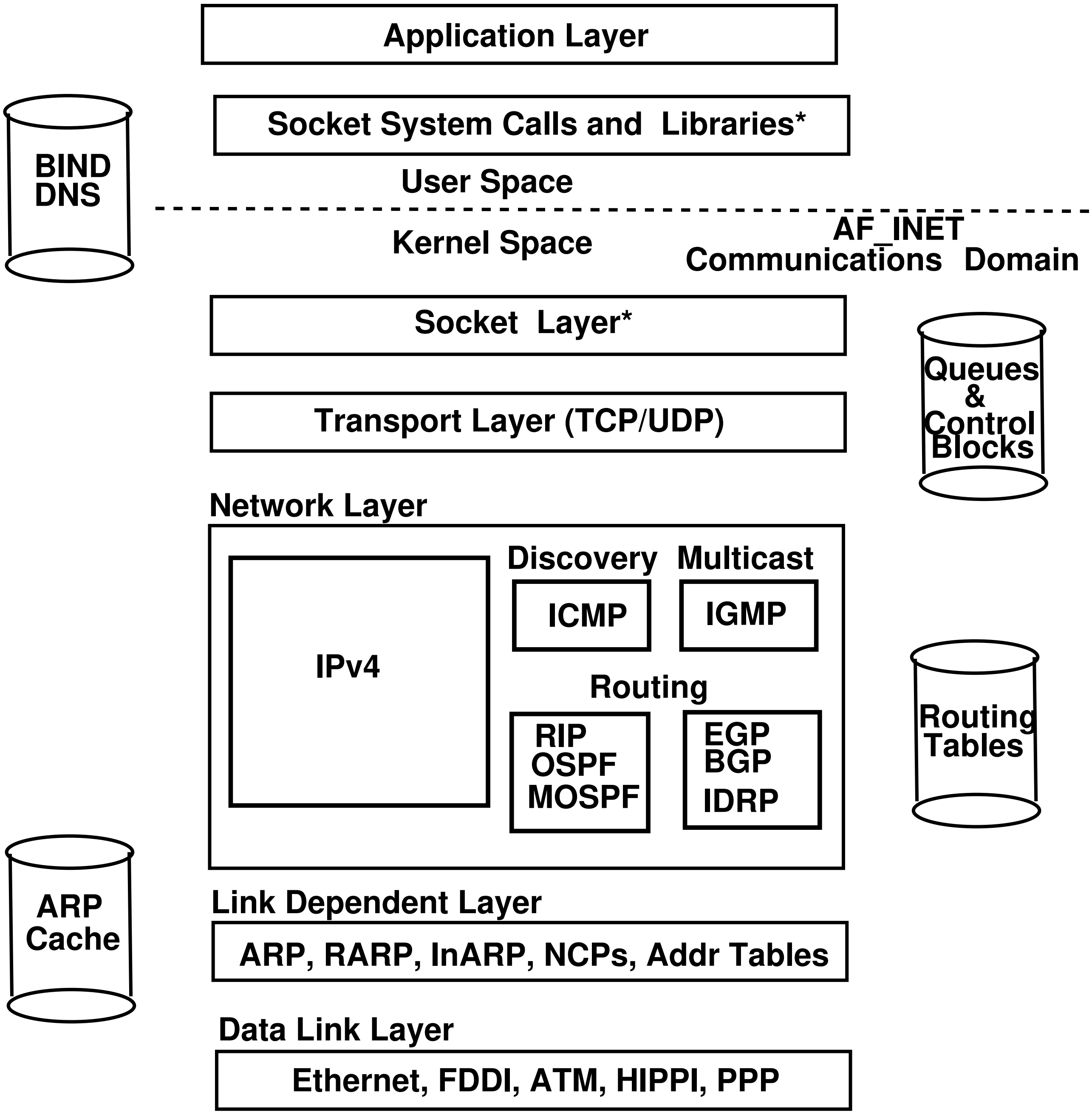
> Path MTU Discovery

> OSPF Routing

> Mobility

A BSD UNIX Host TCP/IP Architecture

Most all parts of the Host model here will require alterations to support IPv6.



* Could be Streams Architecture too

General IPv6 Implementation Philosophy

> Do not create new header files or modules unless they are IPv6 specific.

> IPv6 application servers should support both IPv6 and IPv4 clients, if the node has an IPv4-IPv6 compatible address.

> Use new features in BSD 4.4 (e.g. Radix Tree).

> Increased performance for IPv6.

> If you can fix something in IPv4 now is the time to do it.

> Debug crash dumps when they occur immediately.

General Philosophy IPv6 Transport

> Use 16 octet addresses internally.

> API can be either IPv6 or IPv4

> All hard coded references to combined transport/network layer headers have been removed (e.g. tcpiphdr, udpiphdr), and replaced with a transport layer header.

> Information that was in IPv4 header obtain by other means (e.g. PCBs, Mbuf, Parameter Passing).

> IPv6 checksum must be fast even with large amount of optional data in the network layer header.

API/Socket/Transport Layer

> Socket PCBs for IPv4 and IPv6.

> TCP/UDP PCBs use 16 octet IPv6 address types.

> Check Address Family of API used to determine if IPv4 or IPv6.

> Check if IPv4-IPv6 Compatible or IPv4-Mapped Address.

> Check if packet should be tunneled.

> Network Layer (IPv4 or IPv6) determined by the API and address.

> Cannot call TCP or UDP from socket layer directly.

> Checksum done using transport header and transport data.

> Routines that were very IPv4 specific:

- tcp_subr.c
- tcp_input.c
- tcp_ouput.c
- udp_usrreq.c

> Pseudo header and incremental checksum added.

> Use IP_ADDRFORM in IETF IPv6 BSD API Specification to downgrade or upgrade addresses.

> Applications can support both IPv4 and IPv6 when using IPv6 API.

General Philosophy of Network Layer

> Internetwork Layer modules for IPv4 and IPv6 must be separate.

> Use BSD 4.4 Radix Tree and Interfaces for routing tables and neighbor discovery caches (IPv4 ARP cache), and for the transport layer.

> Use tunnel support as virtual interfaces in the network layer for IPv6-IPv4 encapsulation of packets.

> Look for performance gains because of the way IPv6 has specified network layer options in the network layer header.

> Move neighbor discovery to the Internetwork Layer module.

Network Layer

> IPv4 ipintr() modified to send packets to ip6_input.c when received from the interface.

> If protocol ID is 41 then IPv4 packet with IPv6 encapsulated so send to VIF tunnel and to ip6_input.c.

> Processing is mostly the same as IPv4 except for the changes in the IPv6 protocol (e.g. header alignment, valid options, addresses are IPv6 configured).

> Processing for ip6_output.c is also similar as IPv4 except:

- Check if interface is for tunnel based on route entry passed from transport layer.
- Neighbor Discovery is called if route entry is not present or if link layer address is not available or timed-out.

IPv6 To Study List

> Fragmentation in IPv6 is end-to-end only, need to determine most efficient method for PMTU Discovery for UDP.

> Determine most efficient means to alter connection state based on invalidation and deprecation timers at TCP and UDP for autoconfiguration.

> Source Routes at the API, Transport, and Network Layer.

> Use of Flow-ID.

> RSVP within IPv6 at the API, Transport, and Network Layer.

> NBMA for Neighbor Discovery (e.g. ATM, PPP).

> Integration of IPv6 Address Configuration and DHCPv6.

> The default behavior defined in all IPv6 specifications.

IPv6 Implementation Contributors

> Engineers:
- Jim Bound
- Alex Conta
- Peter Grehan
- Dan Harrington
- Markus Jork
- Jack Mccann
- Matt Thomas

> Public Domain:
- Christian Huitema and Francis Dupont for the first public domain implementation of SIP 8 bytes, which we all learned from.

> IETF for their hard work and wisdom to define and select an IPng in time before the Internet runs out of IPv4 addresses.

Other Implementations

Worldwide Web Page URL

http://playground.sun.com/pub/ipng/html/ipng-main.html

Digital Implementation Status

Worldwide Web Page URL:

http://playground.sun.com/pub/ipng/html/ipng-main.html

Digital UNIX

Digital Equipment Corporation is doing an IPv6 implementation on OSF/1 running on Alpha workstations, IPv6 Critical Functionality based Specifications:

* **IPv6 BSD Sockets (running)**
* **IPv6 DNS Records Server and Resolver (running)**
* **IPv6 TCP/UDP (running)**
* **IPv6 Network Layer Basic (running)**
* **IPv6 Neighbor Discovery (part running - in process)**
* **IPv6-IPv4 Transition Sockets/Transport./Tunnels (running)**
* **IPv6 DHCPv6 Server/Client (in process)**
* **IPv6 Stateless Address Configuration (part running/in process)**
* **IPv6 OSPF Host Router (planned)**
* **IPv6 RSVP over ATM (planned)**
* **IPv6 DNSIND Server and Resolver (planned)**
* **IPv6 Security (planned)**
* **IPv6-IPv4 Translation (TBD pending IETF Specifications)**

IPv6 Applications:

* **telnet (running)**
* **ftp (running)**
* **ttcp (running)**
* **Client/Server App (in-process)**

IPv6 Network Utilities:

* **netstat (running)**
* **tcpdump (in process)**
* **ipv6_config (running)**
* **arp_ng (running)**
* **ping_ng (running)**
* **route_ng (planned)**