Network Working Group
Request for Comments: 1682
Category: Informational

J. Bound
Digital Equipment Corporation

August 1994

IPng BSD Host Implementation Analysis

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This document was submitted to the IETF IPng area in response to RFC 1550. Publication of this document does not imply acceptance by the IPng area of any ideas expressed within. Comments should be submitted to the big-internet@munnari.oz.au mailing list.

Overview

This IPng white paper, IPng BSD Host Implementation Analysis, was submitted to the IPng Directorate to provide a BSD host point of reference to assist with the engineering considerations during the IETF process to select an IPng proposal. The University of California Berkeley Software Distribution (BSD) TCP/IP (4.3 + 4.4) system implementation on a host is used as a point of reference for the paper.

This document only reflects the author's personal analysis based on research and implementation experience for IPng, and does not represent any product or future product from any host vendor. Nor should it be construed that it is promoting any specific IPng at this time.

Acknowledgments

The author would like to acknowledge the many host implementation discussions and inherent knowledge gained from discussions with the following persons within Digital over the past year: Peter Grehan, Eric Rosen, Dave Oran, Jeff Mogul, Bill Duane, Tony Lauck, Bill Hawe, Jesse Walker, John Dustin, Alex Conta, and Fred Glover. The author would also like to acknowledge like discussions from outside his company with Bob Hinden (SUN), Bob Gilligan (SUN), Dave Crocker (SGI), Dave Piscitello (Core Competence), Tracy Mallory (3Comm), Rob Ullmann (Lotus), Greg Minshall (Novell), J Allard (Microsoft), Ramesh Govinden (Bellcore), Sue Thompson (Bellcore), John Curran (NEARnet),

Christian Huitema (INRIA), and Werner Volgels (INESC). The author would also like to thank Digital Equipment Corporation for the opportunity to work on IPng within the IETF as part of his job.

1. Introduction

A host in the context of this white paper is a system that contains an operating system supporting a network subsystem as one of its parts, and an interprocess communications facility to access that network subsystem. These hosts are often referenced as a Workstation, Server, PC, Super Computer, Mainframe, or an Embedded System (Realtime Devices).

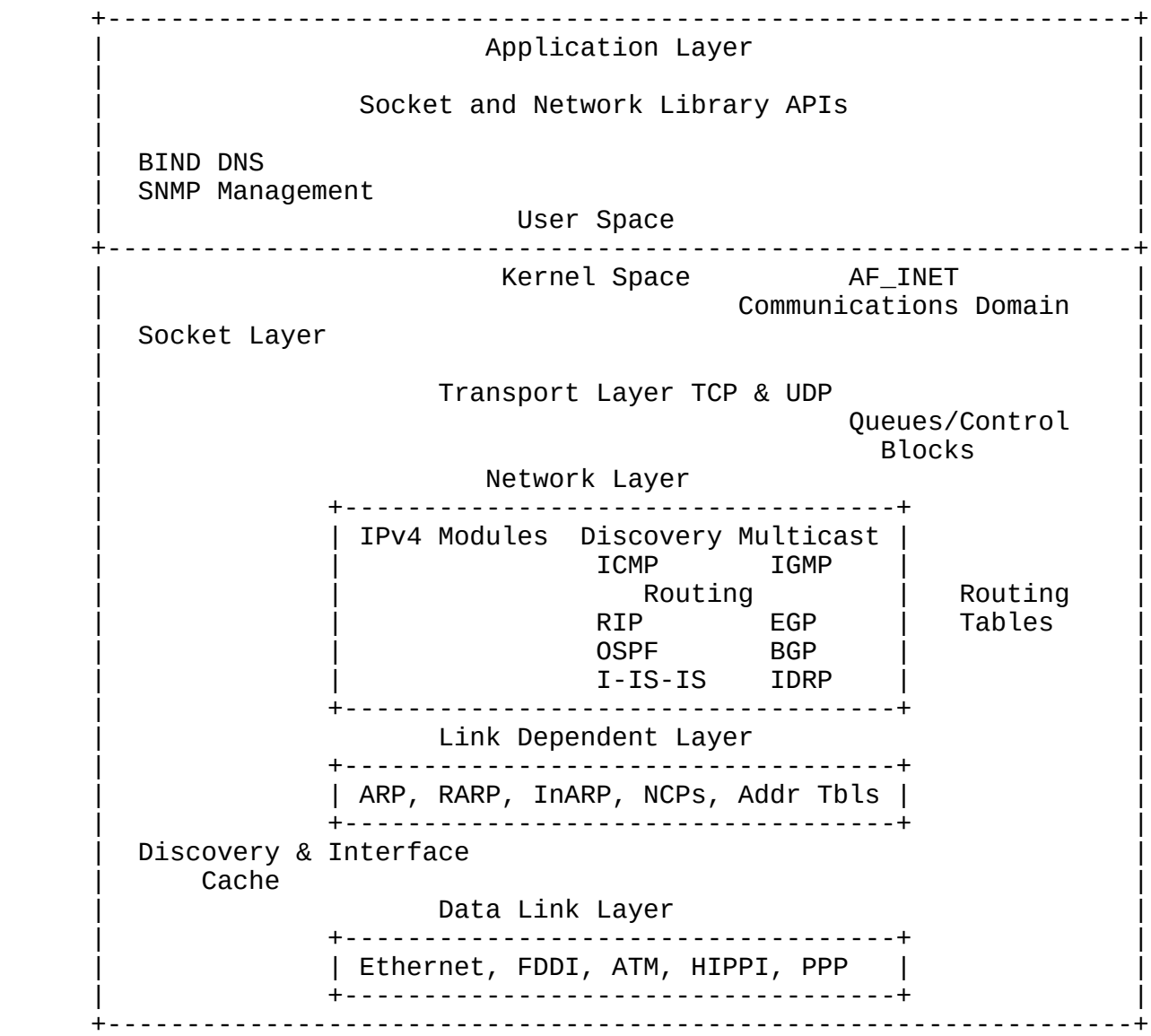
IPng will require changes to a hosts network software architecture. Those changes should be as transparent as possible to the existing IPv4 applications executing on hosts.

After discussing the network software architecture for a BSD host the paper will discuss the perceived network software alterations, extended capabilities, transition software, and a deployment consideration for IPng hosts.

The inclusive OR of all IPng proposals was used to develop the engineering considerations discussed in this paper.

2. Network Software Architecture

The BSD host network software architecture consists essentially of three components: the interprocess communications facility, the network communications subsystem, and the network protocols supported. These three components are tightly coupled and must be integrated in a way that affords high performance for the applications that are dependent on these components to interoperate efficiently. A BSD host implementation view of the TCP/IP protocol suite is depicted in the following network architecture diagram.



2.1 Interprocess Communications Facility

The interprocess communications (IPC) facilities includes three critical parts:

1. The IPC mechanism to the network communications subsystem.
2. The ability to access a network protocol set within that subsystem.
3. The structures supporting the network communications subsystem.

The IPC facility has two implementation parts. The part in user space and the part in kernel space within the operating system. This is often not differentiated and why in the previous network architecture diagram you will see sockets in both user and kernel space. An IPC supports in user space an application program interface (API) which application developers use to access the network communications features of the host. These APIs have corresponding functions in the kernel space which execute the functions requested by the user space requests through the APIs.

The sockets paradigm on a BSD host defines the data structure of the network address within a selected protocol family (communications domain) in the network subsystem. This data structure consists of an address family, a port for the protocol selected, and a network address.

The IPC facility on a host is dependent upon its interface to the BIND DNS application which is the defacto method when using TCP/IP to retrieve network addresses.

Other interfaces that may be required by applications to properly set up the network connection within the IPC facility include: setting/getting options for the protocols used, obtaining/accessing information about networks, protocols, and network services, and sending/transmitting datagrams.

2.2 Network Communications Subsystem

The network communications subsystem consists of the following generic parts as depicted in the previous network architecture diagram: transport layer, network layer, link dependent layer, and data link layer. These may not be implemented as true distinct layers on a BSD host, but they are referenced in this white paper in that manner for purposes of discussion.

The transport layer supports the application interface into the network communications subsystem and sets up the parametric pieces to initiate and accept connections. The transport layer performs these functions through requests to the lower layers of the network communications subsystem. The transport layer also supports the queues and protocol control blocks for specific network connections.

The network layer supports the modules to build and extend the network layer datagram, the control protocol datagrams, and the routing abstraction on the host. This layer of the network communications subsystem on a BSD host is often extended to provide both interior and exterior routing functionality.

The link dependent layer supports the modules that provide an interface for the network communications subsystem to map network addresses to physical addresses, and build the necessary cache so this information is available to the host network software.

On a BSD host the network layer and link dependent layer together provide system discovery for hosts and routers.

The data link layer supports the modules that define the structures for communicating with the hardware media used by the host on the local network.

2.3 Network Protocols

The TCP/IP protocol suite as defined by the IETF RFC specifications are the set of network protocols used by this white paper for reference.

3. Network Software Alterations

The IPng network software alterations to a BSD host perceived at this time are as follows:

1. Applications Embedding IPv4 Addresses.
2. Transport Interfaces and Network APIs.
3. Socket Layer and Structures.
4. Transport Layer.
5. Network Layer Components.
6. Link dependent Layer.

3.1 Applications Embedding IPv4 Addresses

Internet style applications in this white paper are the set of protocols defined for an end user using TCP/IP to exchange messages, transfer files, and establish remote login sessions.

Applications use the sockets network APIs to maintain an opaque view of the network addresses used to support connections across a network. Opaque in this context means that the application determines the network address for the connection and then binds that address to a socket. The application then uses the reference defined for that socket to receive and transmit data across a network.

An application that embeds an IPv4 network address within its datagram has made an underlying assumption that the format of that address is permanent. This will cause a great problem when IPng causes addresses to change. Thus far only one Internet style application has been determined to cause this problem and that is FTP

[1,2].

3.2 Transport Interfaces and Network APIs

The transport interface and network API enhancements that must take place on a BSD host because of IPng are alterations that affect the size of the network address used by the socket data structure. Depending on how this is implemented on the host, supporting both IPv4 and IPng could require existing IPv4 applications to be recompiled. In the worst case it could require modifications to the existing IPv4 applications software that accesses the network communications subsystem.

There will have to be enhancements to the network APIs that an application uses to retrieve BIND DNS records to differentiate between IPv4 and IPng address requests.

The network API enhancements and how they are implemented will affect the capability of any IPng proposal on a BSD host to be able to interoperate between an IPv4 only, an IPng only, and an IPng-IPv4 host system.

Depending on the IPng proposal selected the network options, services, and management objects will have to be extended at the transport interface so those features can be accessed by applications software.

3.3 Socket Layer and Structures

The socket layer and structures will require changes to support any IPng proposals network address. In addition new or removed options and services will need to be incorporated into the socket abstraction within the network communications subsystem.

3.4 Transport Layer

The transport layer will need to be modified to support any new or removed services proposed by an IPng solution set. The transport layer will become more overloaded to support the binding of either the IPv4 or IPng network layer components to differentiate the services and structures available to a host application. The overload will also take place to support functionality removed in the network layer and moved to the transport layer if proposed by an IPng solution.

It will also take some design thought to implement IPng so the hundreds of man years invested in performance improvements in the host transport layer are maintained. This must be analyzed in depth

and should be part of the operational testing of any IPng proposal.

3.5 Network Layer Components

The network layer components for IPng will require the greatest alterations on a host. In addition a host will be required to maintain an integrated network layer below the transport layer software to support either the IPng or IPv4 network layer and associated components.

Depending on the IPng selected the host alterations to the network layer components will range from complete replacement with new protocols to extensions to existing IPv4 network layer protocols to support IPng.

All IPng proposals will affect the BSD host routing abstraction to maintain host software that supports interior and exterior routing. Depending on the proposal selected those changes can cause either a complete new paradigm or an update to the existing IPv4 paradigm.

System discovery of nodes on the local subnetwork or across an internetwork path in all IPng proposals will require changes to the BSD host software network layer component.

3.6 Link dependent Layer

The link dependent layer on a host will need to accommodate new IPng addresses and the system discovery models of any IPng proposal.

4. Extended Capabilities with IPng

Extended capabilities that could be implemented by BSD hosts are listed below. Many of these capabilities exist today with IPv4, but may require changes with the implementation of IPng. Some of them will be new capabilities.

4.1 Autoconfiguration and Autoregistration

Today hosts can provide autoconfiguration with DHCP using IPv4 addresses. IPng hosts will be faced with having to provide support for existing IPv4 addresses and the new IPng addresses. In addition the boot-strap protocol BOOTP used to boot minimal BSD host configurations (e.g., diskless nodes) will need to be supported by IPng hosts.

4.2 PATH MTU Discovery

PATH MTU discovery appears to be something each proposal is considering. Alterations to the existing implementation of PATH MTU are perceived because changes are expected in system discovery.

4.3 Multicast

Each proposal has depicted alterations to Multicast that will affect present BSD host implementations of IPv4 Multicast. In addition it appears that the IPv4 unicast broadcast will be replaced by a multicast broadcast.

4.4 Flow Specification and Handling

This will be an extended capability proposed by all IPngs'.

4.5 System Discovery

Each proposal has depicted a new model for IPng system discovery of a host.

4.6 Translation and Encapsulation

The routing abstraction in a BSD host will have to deal with the affect of any translation or encapsulation of network layer datagrams, if they are required by an IPng.

4.7 Network Layer Security

It is perceived that network layer security will be required at the network layer component of IPng and this will have to be implemented by a BSD host.

4.8 Socket Address Structure

The network kernel socket address structure will change because of IPng.

4.9 Network APIs

The network APIs for a BSD host will have to be enhanced to support IPng. In addition any new options available to the applications because of the IPng network service will have to be added as an option to the APIs.

4.10 Network Management

Network management for IPng will have to support new network objects as defined by the IPng proposal. In addition the data structures in the BSD host network kernel used as information to display network topology will be altered by a new network layer datagram and associated components.

5. Transition Software

Transition software in this white paper references the network software alterations on a host to support both IPv4 and IPng for applications and the hosts operating system network kernel. It is the subject of another set of papers to identify the transition software required by network managers to transition their users from IPv4 to IPng.

Transition software on a host will be required to maintain compatibility between IPv4 and IPng, and to manage both the existing IPv4 and IPng environments as follows:

1. BIND DNS record updates and handling by the application.
2. SNMP management interface and monitoring of host network structures.
3. APIs supporting IPv4 and IPng differentiation for the application.
4. Defacto network tools altered (e.g., tcpdump, traceroute, netstat).
5. ARP to new system discovery.
6. BOOTP diskless node support for IPng.
7. DHCP integration with IPng Autoconfiguration.
8. Routing table configuration on the BSD host (e.g., routed, ifconfig).
9. Selection of the network layer (IPv4 or IPng) at the transport layer.
10. New options and services provided by an IPng protocol.
11. IPv4 and IPng routing protocols in the network layer.
12. IPv4 and IPng system discovery in the network layer.

These are only the highlights of the transition software that a host will have to deal with in its implementation of IPng. The host network architecture diagram depicted previously will require software enhancements to each label in the diagram.

It is very important that each IPng proposal provide a specification for a transition plan from IPv4 to IPng and their technical criteria for the interoperation between IPv4 and IPng.

It should also be a requirement that existing IPv4 applications not have to be recompiled when a host has implemented both an IPv4 and an IPng network layer and associated components.

It is very desirable that when a host implements both an IPv4 and an IPng network layer and associated components that there is no performance degradation on the host compared to the performance of an existing IPv4 only host.

It should not be a requirement by IPng that a host must support both an IPv4 and an IPng network layer.

6. A Deployment Consideration

Complete and extensive technical specifications must be available for any IPng proposal, and a selection of any proposal must accommodate multiple implementations. The IPng Directorate should review proposed specifications for completeness.

It is important that the IPng Directorate determine how long the CIDR IPv4 address plan can extend the life of IPv4 addresses on the Internet. This variable can affect the time we have to deploy IPng and the proposed transition plans.

References

[1] Gilligan, B., et. al., "IPAE: The SIPP Interoperability and Transition Mechanism", Work in Progress.

[2] Piscitello, D., "FTP Operation Over Big Address Records (FOOBAR)", RFC 1639, Core Competence, Inc., June 1994.

Security Considerations

Security issues are discussed in Section 4.7.

Author’s Address

Jim Bound
Digital Equipment Corporation
110 Spitbrook Road ZK3-3/U14
Nashua, NH 03062-2698

Phone: +1 603 881 0400
EMail: bound@zk3.dec.com