Network Working Group
draft-ietf-v6ops-ipv4survey-int-00.txt
Internet Draft

Philip J. Nesser II
Nesser & Nesser Consulting

February 2003
Expires August 2003

Survey of IPv4 Addresses in Currently Deployed IETF Internet Area Standards

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Status of this Memo

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This document seeks to document all usage of IPv4 addresses in currently deployed IETF Internet Area documented standards. In order to successfully transition from an all IPv4 Internet to an all IPv6 Internet, many interim steps will be taken. One of these steps is the evolution of current protocols that have IPv4 dependencies. It is hoped that these protocols (and their implementations) will be redesigned to be network address independent, but failing that will at least dually support IPv4 and IPv6. To this end, all Standards (Full, Draft, and Proposed) as well as Experimental RFCs will be surveyed and any dependencies will be documented.

1.0 Introduction

This work began as a megolithic document draft-ietf-ngtrans-ipv4survey-XX.txt. In an effort to rework the information into a more manageable form, it has been broken into 8 documents conforming to the current IETF areas (Application, General, Internet, Manangement & Operations, Routing, Security, Sub-IP and Transport).

1.1 Short Historical Perspective

There are many challenges that face the Internet Engineering community. The foremost of these challenges has been the scaling issue. How to grow a network that was envisioned to handle thousands of hosts to one that will handle tens of millions of networks with billions of hosts. Over the years this scaling problem has been overcome with changes to the network layer and to routing protocols. (Ignoring the tremendous advances in computational hardware)

The first "modern" transition to the network layer occurred in during the early 1980's from the Network Control Protocol (NCP) to IPv4. This culminated in the famous "flag day" of January 1, 1983. This version of IP was documented in RFC 760. This was a version of IP with 8 bit network and 24 bit host addresses. A year later IP was updated in RFC 791 to include the famous A, B, C, D, & E class system.

Networks were growing in such a way that it was clear that a need for breaking networks into smaller pieces was needed. In October of 1984 RFC **917 was published formalizing the practice of subnetting.**

By the late 1980's it was clear that the current exterior routing protocol used by the Internet (EGP) was not sufficient to scale with the growth of the Internet. The first version of BGP was documented in 1989 in RFC 1105.

The next scaling issues to became apparent in the early 1990's was the exhaustion of the Class B address space. The growth and commercialization of the Internet had organizations requesting IP addresses in alarming numbers. In May of 1992 over 45% of the Class B space was allocated. In early 1993 RFC 1466 was published directing assignment of blocks of Class C's be given out instead of Class B's. This solved the problem of address space exhaustion but had significant impact of the routing infrastructure.

The number of entries in the "core" routing tables began to grow exponentially as a result of RFC 1466. This led to the implementation of BGP4 and CIDR prefix addressing. This may have solved the problem for the present but there are still potential scaling issues.

Current Internet growth would have long overwhelmed the current address space if industry didn't supply a solution in Network Address Translators (NATs). To do this the Internet has sacrificed the underlying "End-to-End" principle.

In the early 1990's the IETF was aware of these potential problems and began a long design process to create a successor to IPv4 that would address these issues. The outcome of that process was IPv6.

The purpose of this document is not to discuss the merits or problems of IPv6. That is a debate that is still ongoing and will eventually be decided on how well the IETF defines transition mechanisms and how

industry accepts the solution. The question is not "should," but "when."

1.2 A Brief Aside

Throughout this document there are discussions on how protocols might be updated to support IPv6 addresses. Although current thinking is that IPv6 should suffice as the dominant network layer protocol for the lifetime of the author, it is not unreasonable to contemplate further upgrade to IP. Work done by the IRTF Interplanetary Internet Working Group shows one idea of far reaching thinking. It may be a reasonable idea (or may not) to consider designing protocols in such a way that they can be either IP version aware or independent. This idea must be balanced against issues of simplicity and performance. Therefore it is recommended that protocol designer keep this issue in mind in future designs.

Just as a reminder, remember the words of Jon Postel:

> "Be conservative in what you send; be liberal in what
> you accept from others."

2.0 Methodology

To perform this study each class of IETF standards are investigated in order of maturity: Full, Draft, and Proposed, as well as Experimental. Informational RFC are not addressed. RFCs that have been obsoleted by either newer versions or as they have transitioned through the standards process are not covered.

Please note that a side effect of this choice of methodology is that some protocols that are defined by a series of RFC's that are of different levels of standards maturity are covered in different spots in the document. Likewise other natural groupings (i.e. MIBs, SMTP extensions, IP over FOO, PPP, DNS, etc.) could easily be imagined.

2.1 Scope

The procedure used in this investigation is an exhaustive reading of the applicable RFC's. This task involves reading approximately 25000 pages of protocol specifications. To compound this, it was more than a process of simple reading. It was necessary to attempt to understand the purpose and functionality of each protocol in order to make a proper determination of IPv4 reliability. The author has made ever effort to make this effort and the resulting document as complete as possible, but it is likely that some subtle (or perhaps not so subtle) dependence was missed. The author encourage those familiar (designers, implementers or anyone who has an intimate knowledge) with any protocol to review the appropriate sections and make comments.

2.2 Document Organization

The rest of the document sections are described below.

Sections 3, 4, 5, and 6 each describe the raw analysis of Full, Draft, and Proposed Standards, and Experimental RFCs. Each RFC is discussed in its turn starting with RFC 1 and ending with RFC 3247. The comments for each RFC is "raw" in nature. That is, each RFC is discussed in a vacuum and problems or issues discussed do not "look ahead" to see if the problems have already been fixed.

Section 7 is an analysis of the data presented in Sections 3, 4, 5, and 6. **It is here that all of the results are considered as a whole and the** problems that have been resolved in later RFCs are correlated.

3.0 Full Standards

Full Internet Standards (most commonly simply referred to as "Standards") are fully mature protocol specification that are widely implemented and used throughout the Internet.

3.01 Internet Protocol. RFC0791, RFC0950, RFC0919, RFC0922, RFC792, RFC1112

3.01.1 RFC 791 defines IPv4 and will be replaced by the IPv6 specifications.

3.01.2 RFC 950 specifies IPv4 subnetting and will be replaced by the IPv6 specifications.

3.01.3 RFC 919 is not online and is unavailable for review.

3.01.4 RFC 922 specifies how broadcasts should be treated in the presence of subnets. The techniques of this document will be replaced by the IPv6 specifications.

3.01.5 RFC 792 defines ICMP. The specification of ICMPv6 will serve as an update.

3.01.6 RFC 1112 defines IP multicast. A similar updated version for IPv6 multicasting must be written.

3.02 Domain Name System. RFC1034, RFC1035

3.02.1 RFC 1034 Domain Concepts and Facilities

In Section 3.6. Resource Records the definition of A records is:

RDATA which is the type and sometimes class dependent data
       which describes the resource:

       A For the IN class, a 32 bit IP address

In Section 5.2.1. Typical functions defines

1. Host name to host address translation.

   This function is often defined to mimic a previous HOSTS.TXT based function. Given a character string, the caller wants one or more 32 bit IP addresses. Under the DNS, it translates into a request for type A RRs. Since the DNS does not preserve the order of RRs, this function may choose to sort the returned addresses or select the "best" address if the service returns only one choice to the client. Note that a multiple address return is recommended, but a single address may be the only way to emulate prior HOSTS.TXT services.

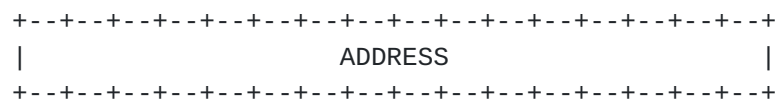
2. Host address to host name translation

   This function will often follow the form of previous functions. Given a 32 bit IP address, the caller wants a character string. The octets of the IP address are reversed, used as name components, and suffixed with "IN-ADDR.ARPA". A type PTR query is used to get the RR with the primary name of the host. For example, a request for the host name corresponding to IP address 1.2.3.4 looks for PTR RRs for domain name "4.3.2.1.IN-ADDR.ARPA".

There are, of course, numerous examples of IPv4 addresses scattered throughout the document. There is currently a large debate ongoing in the DNS community over the use of A6 or AAAA record types for the resolution of IPv6 addresses. The fact that current A records are insufficient to support IPv6 is not unknown to the Internet community.

3.02.2 RFC 1035 DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION

Section 3.4.1. A RDATA format defines the format for A records:



where:

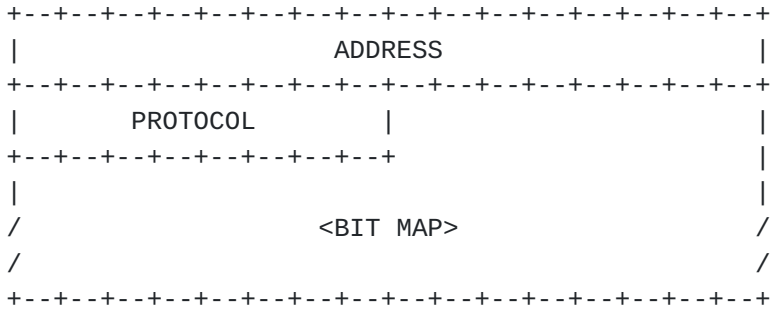
ADDRESS A 32 bit Internet address.

Hosts that have multiple Internet addresses will have

multiple A records.

A records cause no additional section processing. The RDATA section of an A line in a master file is an Internet address expressed as four decimal numbers separated by dots without any imbedded spaces (e.g.,"10.2.0.52" or "192.0.5.6").

Section 3.4.2. WKS RDATA format



where:

ADDRESS	An 32 bit Internet address
PROTOCOL	An 8 bit IP protocol number
<BIT MAP>	A variable length bit map. The bit map must be a multiple of 8 bits long.

The WKS record is used to describe the well known services supported by a particular protocol on a particular internet address. The PROTOCOL field specifies an IP protocol number, and the bit map has one bit per port of the specified protocol. The first bit corresponds to port 0, the second to port 1, etc. If the bit map does not include a bit for a protocol of interest, that bit is assumed zero. The appropriate values and mnemonics for ports and protocols are specified in [RFC-1010].

For example, if PROTOCOL=TCP (6), the 26th bit corresponds to TCP port 25 (SMTP). If this bit is set, a SMTP server should be listening on TCP port 25; if zero, SMTP service is not supported on the specified address.

The purpose of WKS RRs is to provide availability information for servers for TCP and UDP. If a server supports both TCP and UDP, or has multiple Internet addresses, then multiple WKS RRs are used.

WKS RRs cause no additional section processing.

Section 3.5. IN-ADDR.ARPA domain describe reverse DNS lookups and is clearly IPv4 dependent.

There are, of course, numerous examples of IPv4 addresses scattered throughout the document.

3.03 RFC 894 Standard for the transmission of IP datagrams over Ethernet networks

This protocol specifically deals with the transmissions of IPv4 packets over Ethernet. A similar RFC must exist for transmission of IPv6 packets.

3.04 RFC 895 Standard for the transmission of IP datagrams over experimental Ethernetnetworks

This protocol specifically deals with the transmissions of IPv4 packets over Ethernet. It is probably unnecessary to provide an updated RFC because of the unlikelihood of the existence of this layer 2 medium.

3.05 RFC 1042 Standard for the transmission of IP datagrams over IEEE 802 networks

This protocol specifically deals with the transmissions of IPv4 packets over Ethernet. A similar RFC must exist for transmission of IPv6 packets, particularly for 802.5 networks.

3.06 RFC 891 DCN Local-Network Protocols

There are many implicit assumptions about the use of IPv4 addresses in this document. It is unlikely to require any updates since no DCN networks are in existence.

3.07 RFC 1044 Internet Protocol on Network System's HYPERchannel: Protocol Specification

There are a variety of methods used in this standard to map IPv4 addresses to 32 bits fields in the HYPERchannel headers. A new version of the standard
will need to be written do support IPv6 on HYPERchannel networks.

3.08 RFC 1201 Transmitting IP traffic over ARCNET networks

The major concerns of this RFC with respect to IPv4 addresses occur in the resolution of ARCnet 8bit addresses to IPv4 addresses in an "ARPlike" method.
A similar method, very similar to this RFC, would need to be written to support
IPv6 addresses over ARCNET.

3.09 RFC 1055 Nonstandard for transmission of IP datagrams over serial lines:
SLIP

This RFC is more of a analysis of the shortcomings of SLIP which is unsurprising. The introduction of PPP as a general replacement of SLIP has made this protocol essentially unused. No update need be considered.

3.10 RFC 1088 Standard for the transmission of IP datagrams over NetBIOS networks

This RFC documents a technique to encapsulate IP packets inside NetBIOS packets.
The technique presented of using NetBIOS names of the form IP.XX.XX.XX.XX will
not work for IPv6 addresses since the length of IPv6 addresses will not fit within the NetBIOS 15 octet name limitation. A new scheme must be invented to similarly encapsulate IPv6 packets.

3.11 The Point-to-Point Protocol (PPP). RFC1661, RFC1662

3.11.1 RFC 1661 The Point-to-Point Protocol (PPP)

The Point-to-Point Protocol (PPP)

3.11.2 RFC 1662 PPP in HDLC-like Framing

There are no IPv4 dependencies in this protocol.

3.12 RFC 1209 The Transmission of IP Datagrams over the SMDS Service

This RFC defines running IPv4 and ARP over SMDS. The methods described could easily be extended to support IPv6 packets, but a new RFC would be required.

4.0 Draft Standards

Draft Standards represent the penultimate standard level in the IETF.
A protocol can only achieve draft standard when there are multiple, independent, interoperable implementations. Draft Standards are usually quite mature and widely used.

4.01 RFC 951 Bootstrap Protocol (BOOTP)

This protocol is designed specifically for use with IPv4. A new version

will be required to support IPv6. For example:

Section 3. Packet Format

> All numbers shown are decimal, unless indicated otherwise. The BOOTP packet is enclosed in a standard IP [8] UDP [7] datagram. For simplicity it is assumed that the BOOTP packet is never fragmented. Any numeric fields shown are packed in 'standard network byte order', i.e. high order bits are sent first.
>
> In the IP header of a bootrequest, the client fills in its own IP source address if known, otherwise zero. When the server address is unknown, the IP destination address will be the 'broadcast address' 255.255.255.255. This address means 'broadcast on the local cable, (I don't know my net number)' [4].

...

```
      FIELD   BYTES   DESCRIPTION
      -----   -----   ---
...
         ciaddr  4       client IP address;
                         filled in by client in bootrequest if known.

         yiaddr  4       'your' (client) IP address;
                         filled by server if client doesn't
                         know its own address (ciaddr was 0).

         siaddr  4       server IP address;
                         returned in bootreply by server.

         giaddr  4       gateway IP address,
                         used in optional cross-gateway booting.
```

Since the packet format is a fixed 300 bytes in length, an updated version of the protocol could easily accommodate an additional 48 bytes (4 IPV6 fields of 16 bytes to replace the existing 4 IPv4 fields of **4 bytes).**

4.02 RFC 1191 Path MTU discovery (IP-MTU)

The entire process of PMTU discovery is predicated on the use of the DF bit in the IPv4 header, an ICMP message (also IPv4 dependent) and TCP MSS option. There clearly needs to an PMTUv6 functionality.

4.03 RFC 1534 Interoperation Between DHCP and BOOTP (DHCP-BOOTP)

There are no IPv4 dependencies in this protocol.

4.04 RFC 1542 Clarifications and Extensions for the Bootstrap Protocol

There are no new issues other than those presented in Section 4.01 above.

4.05 RFC 1629 Guidelines for OSI NSAP Allocation in the Internet (OSI-NSAP)

There are no IPv4 dependencies in this protocol.

4.06 RFC 1762 The PPP DECnet Phase IV Control Protocol (DNCP) (PPP-DNCP)

There are no IPv4 dependencies in this protocol.

4.07 RFC 1989 PPP Link Quality Monitoring (PPP-LINK)

There are no IPv4 dependencies in this protocol.

4.08 RFC 1990 The PPP Multilink Protocol (MP) (PPP-MP)

Section 5.1.3. Endpoint Discriminator Option defines a Class header field.

Class

The Class field is one octet and indicates the identifier address space. The most up-to-date values of the LCP Endpoint Discriminator Class field are specified in the most recent "Assigned Numbers" RFC [7]. Current values are assigned as follows:

0 Null Class

1 Locally Assigned Address

2 Internet Protocol (IP) Address

3 IEEE 802.1 Globally Assigned MAC Address

4 PPP Magic-Number Block

5 Public Switched Network Directory Number

A new class field needs to be defined by the IANA for IPv6 addresses.

4.09 RFC 1994 PPP Challenge Handshake Authentication Protocol (CHAP) (PPP-CHAP)

There are no IPv4 dependencies in this protocol.

4.10 RFC 2067 IP over HIPPI (IP-HIPPI)

Section 5.1 Packet Formats contains the following excerpt:

> EtherType (16 bits) SHALL be set as defined in Assigned Numbers [8]:
> IP = 2048 ('0800'h), ARP = 2054 ('0806'h), RARP = 32,821 ('8035'h).

Section 5.5 MTU has the following definition:

> The MTU for HIPPI-SC LANs is 65280 bytes.
>
> This value was selected because it allows the IP packet to fit in one 64K byte buffer with up to 256 bytes of overhead. The overhead is 40 bytes at the present time; there are 216 bytes of room for expansion.

```
      HIPPI-FP Header                 8 bytes
      HIPPI-LE Header                24 bytes
      IEEE 802.2 LLC/SNAP Headers     8 bytes
      Maximum IP packet size (MTU) 65280 bytes
                                   ------------
                        Total      65320 bytes (64K - 216)
```

This definition is not applicable for IPv6 packets since packets can be larger than the IPv4 limitation of 65280 bytes.

4.11 RFC 2131 Dynamic Host Configuration Protocol (DHCP)

This version of DHCP is highly assumptive of IPv4. Significant work on DHCPv6 has been done and is ongoing.

4.12 RFC 2132 DHCP Options and BOOTP Vendor Extensions (DHCP-BOOTP)

This version of DHCP is highly assumptive of IPv4. Significant work on DHCPv6 has been done and is ongoing.

4.13 RFC 2460 Internet Protocol, Version 6 (IPv6) Specification (IPV6)

This document defines IPv6 and has no IPv4 issues.

4.14 RFC 2461 Neighbor Discovery for IP Version 6 (IPv6) (IPV6-ND)

This document defines an IPv6 related protocol and has no IPv4 issues.

4.15 RFC 2462 IPv6 Stateless Address Autoconfiguration (IPV6-AUTO)

This document defines an IPv6 related protocol and has no IPv4 issues.

4.16 RFC 2463 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification (ICMPv6)

This document defines an IPv6 related protocol and has no IPv4 issues.

5.0 Proposed Standards

Proposed Standards are introductory level documents. There are no requirements for even a single implementation. In many cases Proposed are never implemented or advanced in the IETF standards process. They therefore are often just proposed ideas that are presented to the Internet community. Sometimes flaws are exposed or they are one of many competing solutions to problems. In these later cases, no discussion is presented as it would not serve the purpose of this discussion.

5.01 RFC 1234 Tunneling IPX traffic through IP networks (IPX-IP)

The section "Unicast Address Mappings" has the following text:

> For implementations of this memo, the first two octets of the host number will always be zero and the last four octets will be the node's four octet IP address. This makes address mapping trivial for unicast transmissions: the first two octets of the host number are discarded, leaving the normal four octet IP address. The encapsulation code should use this IP address as the destination address of the UDP/IP tunnel packet.

This mapping will not be able to work with IPv6 addresses.

There are also numerous discussions on systems keeping a "peer list" to map between IP and IPX addresses. The specifics are not discussed in the document and are left to the individual implementation.

The section "Maximum Transmission Unit"

> Although larger IPX packets are possible, the standard maximum transmission unit for IPX is 576 octets. Consequently, 576 octets is the recommended default maximum transmission unit for IPX packets being sent with this encapsulation technique. With the eight octet UDP header and the 20 octet IP header, the resulting IP packets will be 604 octets long. Note that this is larger than the 576 octet maximum size IP implementations are required to accept [3]. Any IP implementation supporting this encapsulation technique must be capable of receiving 604 octet IP packets.
>
> As improvements in protocols and hardware allow for larger, unfragmented IP transmission units, the 576 octet maximum IPX packet size may become a liability. For this reason, it is recommended that

the IPX maximum transmission unit size be configurable in implementations of this memo.

also has some implications on IP addressing.

5.02 RFC 1256 ICMP Router Discovery Messages (ICMP-ROUT)

This RFC documents a protocol that is very specific to IPv4 and a successor will be needed to provide the functionality.

5.03 RFC 1277 Encoding Network Addresses to Support Operation over Non-OSI Lower Layers

Section 4.5 TCP/IP (RFC 1006) Network Specific Format states:

The IDP and 2 digit prefix identifies a TCP/IP network where RFC 1006 is applied. It is necessary to use an IP Address, as there are insufficient bits to fit in a domain. It is structured as follows:

Digit	1-12	13-17 (optional)	18-22 (optional)
Meaning	IP_Address	port	Transport_Set

For TCP/IP there shall be a 20 digit long network-specific part. First 12 digits are for the IP address. The port number can be up to 65535, and needs 5 digits (this is optional, and is defaulted as defined in RFC 1006). Finally, there is a third part to the address, which is defined here as ``transport set'' that indicates what kind of IP-based transport protocols is used. This is a decimal number from 0-65535 which is really a 16-bit flag word. 1 is TCP, 2 is UDP. Further values of this code are assigned by the IANA. If the transport set is not there or no bits are set, it means ``default'' which is TCP. This is encoded in 5 digits.

For example, the IP Address 10.0.0.6 with port 9 over UDP is encoded as:

Part		IDP		DSP		
Component	AFI	IDI	Prefix	IP_Address	Port	T_Set
Octet			1-2	3-14	15-19	20-24
Value	T	elex	007_28722	03	010_000_000_006	00009	00002
Cncrt_Dec		54	007_28722	03	010_000_000_006	00009	00002
Cncrt_Bin		54	00_72_87_22	03	01_00_00_00_00_06	00_00_9	0_00_02

This 12 octet field for decimal versions of IP addresses is insufficient for a decimal version of IPv6 addresses. It is possible to define a new encoding using the 20 digit long IP Address + Port + Transport Set fields in order to accommodate a binary version of an IPv6 address, port number and Transport Set. There are several schemes that could be envisioned.

5.04 RFC 1332 The PPP Internet Protocol Control Protocol (IPCP) (PPP-IPCP)

This document defines a protocol for devices to assign IPv4 addresses to PPP clients once PPP negotiation is completed. Section 3. IPCP Configuration Options defines the following:

The most up-to-date values of the IPCP Option Type field are specified in the most recent "Assigned Numbers" RFC [6]. Current values are assigned as follows:

1 IP-Addresses
2 IP-Compression-Protocol
3 IP-Address

3.1. IP-Addresses

Description

The use of the Configuration Option IP-Addresses has been deprecated. It has been determined through implementation experience that it is difficult to ensure negotiation convergence in all cases using this option. RFC 1172 [7] provides information for implementations requiring backwards compatibility. The IP-Address Configuration Option replaces this option, and its use is preferred.

This option SHOULD NOT be sent in a Configure-Request if a Configure-Request has been received which includes either an IP-Addresses or IP-Address option. This option MAY be sent if a Configure-Reject is received for the IP-Address option, or a Configure-Nak is received with an IP-Addresses option as an appended option.

Support for this option MAY be removed after the IPCP protocol status advances to Internet Draft Standard.

3.3. IP-Address

Description

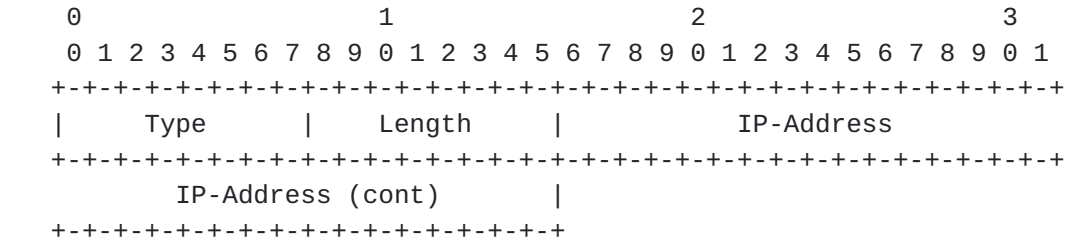
This Configuration Option provides a way to negotiate the IP

address to be used on the local end of the link. It allows the sender of the Configure-Request to state which IP-address is desired, or to request that the peer provide the information. The peer can provide this information by NAKing the option, and returning a valid IP-address.

If negotiation about the remote IP-address is required, and the peer did not provide the option in its Configure-Request, the option SHOULD be appended to a Configure-Nak. The value of the IP-address given must be acceptable as the remote IP-address, or indicate a request that the peer provide the information.

By default, no IP address is assigned.

A summary of the IP-Address Configuration Option format is shown below. The fields are transmitted from left to right.



Type

3

Length

6

IP-Address

The four octet IP-Address is the desired local address of the sender of a Configure-Request. If all four octets are set to zero, it indicates a request that the peer provide the IP-Address information.

Default

No IP address is assigned.

It is clearly designed to allow new Option Types to be added and should offer no problems for use with IPv6 once appropriate options have been defined.

5.05 RFC 1377 The PPP OSI Network Layer Control Protocol (OSINLCP)

(PPP-OSINLC)

There are no IPv4 dependencies in this protocol.

5.06 RFC 1378 The PPP AppleTalk Control Protocol (ATCP) (PPP-ATCP)

There are no IPv4 dependencies in this protocol.

5.07 RFC 1469 IP Multicast over Token-Ring Local Area Networks (IP-TR-MC)

This document defines the usage of IPv4 multicast over IEEE 802.5 Token Ring networks. A new method for IPv6 multicast over these networks will need to be defined.

5.08 RFC 1552 The PPP Internetworking Packet Exchange Control Protocol (IPXCP) (IPXCP)

There are no IPv4 dependencies in this protocol.

5.09 RFC 1570 PPP LCP Extensions (PPP-LCP)

There are no IPv4 dependencies in this protocol.

5.10 RFC 1598 PPP in X.25 PPP-X25

There are no IPv4 dependencies in this protocol.

5.11 RFC 1618 PPP over ISDN (PPP-ISDN)

There are no IPv4 dependencies in this protocol.

5.12 RFC 1663 PPP Reliable Transmission (PPP-TRANS)

There are no IPv4 dependencies in this protocol.

5.13 RFC 1752 The Recommendation for the IP Next Generation Protocol (IPNG)

This document defines a roadmap for IPv6 development and is not relevant to this discussion.

5.14 RFC 1755 ATM Signaling Support for IP over ATM (ATM)

There are no IPv4 dependencies in this protocol.

5.15 RFC 1763 The PPP Banyan Vines Control Protocol (BVCP) (BVCP)

There are no IPv4 dependencies in this protocol.

5.16 RFC 1764 The PPP XNS IDP Control Protocol (XNSCP) (XNSCP)

There are no IPv4 dependencies in this protocol.

5.17 RFC 1886 DNS Extensions to support IP version 6 (DNS-IPV6)

This RFC defines the AAAA record for IPv6 as well as PTR records using the ip6.int domain. There is currently a large debate going on in the IPv6 and DNS community over the validity of AAAA versus A6 records.

5.18 RFC 1973 PPP in Frame Relay (PPP-FRAME)

There are no IPv4 dependencies in this protocol.

5.19 RFC 1981 Path MTU Discovery for IP version 6 MTU-IPV6

This protocol describes an IPv6 related protocol and is not discussed in this document.

5.20 RFC 1982 Serial Number Arithmetic (SNA)

There are no IPv4 dependencies in this protocol.

5.21 RFC 1995 Incremental Zone Transfer in DNS (DNS-IZT)

Although the examples used in this document use IPv4 adddresses, (i.e. A records) there is nothing in the protocol to preclude full and proper functionality using IPv6.

5.22 RFC 1996 A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY) (DNS-NOTIFY)

There are no IPv4 dependencies in this protocol.

5.23 RFC 2002 IP Mobility Support (MOBILEIPSU)

This document is designed for use in IPv4 networks. There are numerous referrals to other IP "support" mechanisms (i.e. ICMP Router Discover Messages) that specifically refer to the IPv4

of ICMP. An IP Mobility protocol for IPv6 is required.

5.24 RFC 2003 IP Encapsulation within IP (IPENCAPIP)

This document is designed for use in IPv4 networks. There are many referenced to a specified IP version number of 4 and 32-bit addresses. An IPv6 Encapsulation within IPv6 is required.

5.25 RFC 2004 Minimal Encapsulation within IP (MINI-IP)

This document is designed for use in IPv4 networks. There are many referenced to a specified IP version number of 4 and 32-bit addresses. A Minimal IPv6 Encapsulation within IPv6 is required.

5.26 RFC 2005 Applicability Statement for IP Mobility Support

This RFC documents the interoperation of IPv4 mobility as documented in the preceding 3 section.

5.27 RFC 2022 Support for Multicast over UNI 3.0/3.1 based ATM Networks (MULTI-UNI)

This protocol specifically maps IPv4 multicast and a new version is required to support IPv6 multicast.

5.28 RFC 2043 The PPP SNA Control Protocol (SNACP) (PPP-SNACP)

There are no IPv4 dependencies in this protocol.

5.29 RFC 2097 The PPP NetBIOS Frames Control Protocol (NBFCP) (PPP-NBFCP)

There are no IPv4 dependencies in this protocol.

5.30 RFC 2113 IP Router Alert Option (ROUT-ALERT)

This document provides a new mechanism for IPv4. It is expected that a similar functionality will be included in IPv6.

5.31 RFC 2125 The PPP Bandwidth Allocation Protocol (BAP) / The PPP Bandwidth Allocation Control Protocol (BACP) (BAP-BACP)

There are no IPv4 dependencies in this protocol.

5.32 RFC 2136 Dynamic Updates in the Domain Name System (DNS UPDATE) (DNS-UPDATE)

There are no IPv4 dependencies in this protocol.

5.33 RFC 2181 Clarifications to the DNS Specification (DNS-CLAR)

There are no IPv4 dependencies in this protocol. The only reference to IP addresses discuss the use of any cast address, so it should be assumed that these mechanisms are IPv6 operable.

5.34 RFC 2225 Classical IP and ARP over ATM (IP-ATM)

>From the many references in this document it is clear that this document is designed for IPv4 only. It is only later in the document that it is implicitly stated, as in:

> ar$spln - length in octets of the source protocol address. Value range is 0 or 4 (decimal). For IPv4 ar$spln is 4.
>
> ar$tpln - length in octets of the target protocol address. Value range is 0 or 4 (decimal). For IPv4 ar$tpln is 4.

and

> For backward compatibility with previous implementations, a null IPv4 protocol address may be received with length = 4 and an allocated address in storage set to the value 0.0.0.0. Receiving stations MUST be liberal in accepting this format of a null IPv4 address. However, on transmitting an ATMARP or InATMARP packet, a null IPv4 address MUST only be indicated by the length set to zero and MUST have no storage allocated.

A new specification for IPv6 must be defined.

5.35 RFC 2226 IP Broadcast over ATM Networks

This document is limited to IPv4 multicasting. A new specification for IPv6 must be defined.

5.36 RFC 2236 Internet Group Management Protocol, Version 2 (IGMP)

This document describes of version of IGMP used for IPv4 multicast. A similar methodology for IPv6 multicast needs to be defined.

5.37 RFC 2241 DHCP Options for Novell Directory Services (DHCP-NDS)

This document defines extensions for the IPv4 only version of

DHCP and it is expected that similar options will be present in DHCPv6.

5.38 RFC 2242 NetWare/IP Domain Name and Information (NETWAREIP)

Once again these are options to the IPv4 version of DHCP. It is expected that similar options will for IPv6 will exist in DHCPv6.

PREFERRED_DSS (code 6)

Length is (n * 4) and the value is an array of n IP addresses, each four bytes in length. The maximum number of addresses is 5 and therefore the maximum length value is 20. The list contains the addresses of n NetWare Domain SAP/RIP Server (DSS).

NEAREST_NWIP_SERVER (code 7)

Length is (n * 4) and the value is an array of n IP addresses, each four bytes in length. The maximum number of addresses is 5 and therefore the maximum length value is 20. The list contains the addresses of n Nearest NetWare/IP servers.

PRIMARY_DSS (code 11)

Length of 4, and the value is a single IP address. This field identifies the Primary Domain SAP/RIP Service server (DSS) for this NetWare/IP domain. NetWare/IP administration utility uses this value as Primary DSS server when configuring a secondary DSS server.

5.39 RFC 2290 Mobile-IPv4 Configuration Option for PPP IPCP

This protocol is IPv4 specific. It is expected that similar methods will be developed for Mobile IPv6.

5.40 RFC 2308 Negative Caching of DNS Queries (DNS NCACHE) (DNS-NCACHE)

Although there are numerous examples in this document that use IPv4 "A" records, there is nothing in the protocol that limits its effectiveness to IPv4.

5.41 RFC 2331 ATM Signaling Support for IP over ATM - UNI Signaling 4.0 Update (UNI-SIG)

There are no IPv4 dependencies in this protocol.

5.42 RFC 2363 PPP Over FUNI (PPP-FUNI)

There are no IPv4 dependencies in this protocol.

5.43 RFC 2364 PPP Over AAL5 (PPP-AAL)

There are no IPv4 dependencies in this protocol.

5.44 RFC 2371 Transaction Internet Protocol Version 3.0 TIPV3

This document states:

> TIP transaction manager addresses take the form:
>
> <hostport><path>
>
> The <hostport> component comprises:
>
> <host>[:<port>]
>
> where <host> is either a <dns name> or an <ip address>; and <port> is a decimal number specifying the port at which the transaction manager (or proxy) is listening for requests to establish TIP connections. If the port number is omitted, the standard TIP port number (3372) is used.
>
> A <dns name> is a standard name, acceptable to the domain name service. It must be sufficiently qualified to be useful to the receiver of the command.
>
> An <ip address> is an IP address, in the usual form: four decimal numbers separated by period characters.

and further along it states:

> A TIP URL takes the form:
>
> tip://<transaction manager address>?<transaction string>
>
> where <transaction manager address> identifies the TIP transaction manager (as defined in Section 7 above); and <transaction string> specifies a transaction identifier, which may take one of two forms (standard or non-standard):
>
> i. "urn:" <NID> ":" <NSS>
>
> A standard transaction identifier, conforming to the proposed Internet Standard for Uniform Resource Names (URNs), as specified by RFC2141; where <NID> is the Namespace Identifier, and <NSS> is the Namespace Specific String. The Namespace ID determines the syntactic interpretation of the Namespace Specific String. The

Namespace Specific String is a sequence of characters representing a transaction identifier (as defined by <NID>). The rules for the contents of these fields are specified by [6] (valid characters, encoding, etc.).

This format of <transaction string> may be used to express global transaction identifiers in terms of standard representations. Examples for <NID> might be <iso> or <xopen>. e.g.

tip://123.123.123.123/?urn:xopen:xid

Note that Namespace Ids require registration. See [7] for details on how to do this.

ii. <transaction identifier>

A sequence of printable ASCII characters (octets with values in the range 32 through 126 inclusive (excluding ":") representing a transaction identifier. In this non-standard case, it is the combination of <transaction manager address> and <transaction identifier> which ensures global uniqueness. e.g.

tip://123.123.123.123/?transid1

It is not hard to assume that the production of an updated protocol specification that supports IPv6 could be accomplished.

5.45 RFC 2373 IP Version 6 Addressing Architecture,

This RFC documents IPv6 addressing and is not discussed in this document.

5.46 RFC 2374 An IPv6 Aggregatable Global Unicast Address Format,

This RFC documents IPv6 addressing and is not discussed in this document.

5.47 RFC 2464 Transmission of IPv6 Packets over Ethernet Networks

This RFC documents a method for transmitting IPv6 packets over ethernet and is not considered in this discussion.

5.48 RFC 2470 Transmission of IPv6 Packets over Token Ring Networks

This RFC documents a method for transmitting IPv6 packets over token ring and is not considered in this discussion.

5.49 RFC 2472 IP Version 6 over PPP (IPv6-PPP)

This RFC documents a method for transmitting IPv6 packets over PPP and is not considered in this discussion.

5.50 RFC 2473 Generic Packet Tunneling in IPv6 Specification

This RFC documents an IPv6 aware protocol and is not considered in this discussion.

5.51 RFC 2484 PPP LCP Internationalization Configuration Option

There are no IPv4 dependencies in this protocol.

5.52 RFC 2485 DHCP Option for The Open Group's User Authentication Protocol

This document defines extensions for the IPv4 only version of DHCP and it is expected that similar options will be present in DHCPv6.

5.53 RFC 2486 The Network Access Identifier (NAI)

There are no IPv4 dependencies in this protocol.

5.54 RFC 2491 IPv6 over Non-Broadcast Multiple Access (NBMA) networks (IPv6-NBMA)

This RFC documents a method for transmitting IPv6 packets over NBMA networks and is not considered in this discussion.

5.55 RFC 2492 IPv6 over ATM Networks (IPv6ATMNET)

This RFC documents a method for transmitting IPv6 packets over ATM networks and is not considered in this discussion.

5.56 RFC 2497 Transmission of IPv6 Packets over ARCnet Networks

This RFC documents a method for transmitting IPv6 packets over ARCnet networks and is not considered in this discussion.

5.57 RFC 2507 IP Header Compression

This protocol is both IPv4 and IPv6 aware.

5.58 RFC 2526 Reserved IPv6 Subnet Anycast Addresses

This RFC documents IPv6 addressing and is not discussed in this document.

5.59 RFC 2529 Transmission of IPv6 over IPv4 Domains without Explicit Tunnels

This RFC documents IPv6 transmission methods and is not discussed in this document.

5.60 RFC 2563 DHCP Option to Disable Stateless Auto-Configuration in IPv4 Clients

This document is only designated for IPv4. It is expected that similar functionality is available in DHCPv6.

5.61 RFC 2590 Transmission of IPv6 Packets over Frame Relay Networks Specification

This RFC documents IPv6 transmission method over Frame Relay and is not discussed in this document.

5.62 RFC 2610 DHCP Options for Service Location Protocol

This document is only designated for IPv4. It is expected that similar functionality is available in DHCPv6.

5.63 RFC 2615 PPP over SONET/SDH

There are no IPv4 dependencies in this protocol.

5.64 RFC 2671 Extension Mechanisms for DNS (EDNS0) (EDNS0)

There are no IPv4 dependencies in this protocol.

5.65 RFC 2672 Non-Terminal DNS Name Redirection

This document is only defined for IPv4 addresses. A similar specification for IPv6 addresses should be defined.

5.66 RFC 2673 Binary Labels in the Domain Name System (DNS)

This document is only defined for IPv4 addresses. A similar specification for IPv6 addresses should be defined.

5.67 RFC 2675 IPv6 Jumbograms

This document defines a IPv6 packet format and is therefore not discussed in this document.

5.68 RFC 2686 The Multi-Class Extension to Multi-Link PPP

There are no IPv4 dependencies in this protocol.

5.69 RFC 2687 PPP in a Real-time Oriented HDLC-like Framing

There are no IPv4 dependencies in this protocol.

5.70 RFC 2688 Integrated Services Mappings for Low Speed Networks

There are no IPv4 dependencies in this protocol.

5.71 RFC 2710 Multicast Listener Discovery (MLD) for IPv6 (MLD-IPv6)

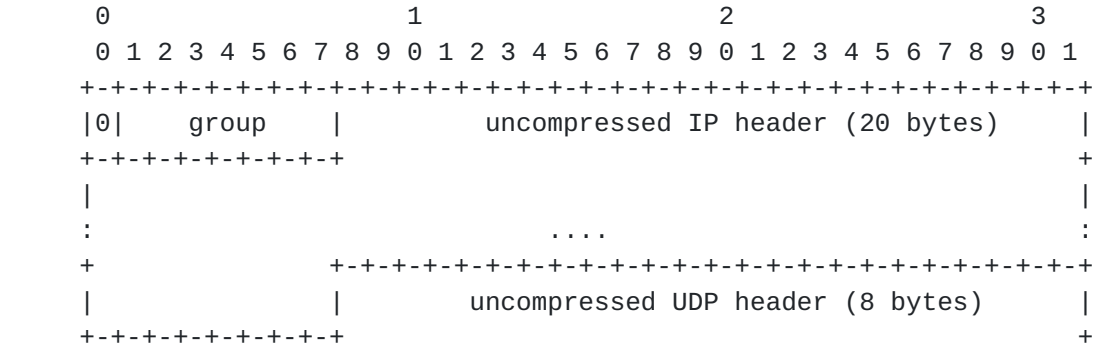
This document defines an IPv6 specific protocol and is not discussed in this document.

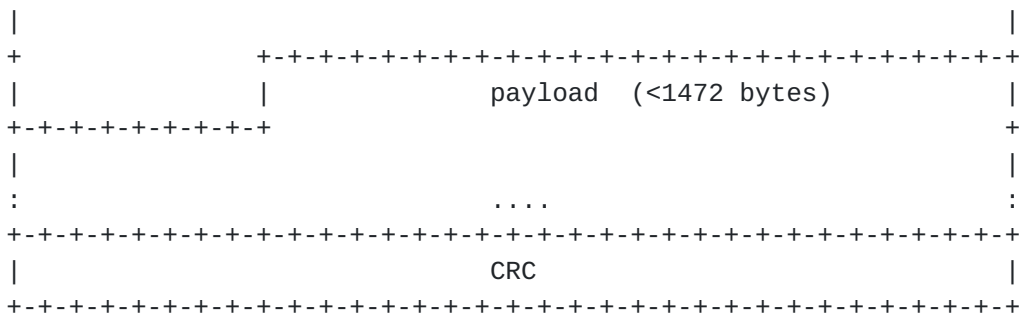
5.72 RFC 2711 IPv6 Router Alert Option

This document defines an IPv6 specific protocol and is not discussed in this document.

5.73 RFC 2728 The Transmission of IP Over the Vertical Blanking Interval of a Television Signal

The following data format is defined:





This protocol is IPv4 dependent. Updates must be made to support IPv6.

5.74 RFC 2734 IPv4 over IEEE 1394

This protocol is IPv4 only. A similar document must be defined for IPv6.

5.75 RFC 2765 Stateless IP/ICMP Translation Algorithm (SIIT) (SIIT)

This protocol defines a method for IPv6 transition and is not discussed in this document.

5.76 RFC 2766 Network Address Translation - Protocol Translation (NAT-PT) (NAT-PT)

This protocol defines a method for IPv6 transition and is not discussed in this document.

5.77 RFC 2776 Multicast-Scope Zone Announcement Protocol (MZAP) (MZAP)

This protocol is both IPv4 and IPv6 aware and needs no changes.

5.78 RFC 2782 A DNS RR for specifying the location of services (DNS SRV) (DNS-SRV)

There are no IPv4 dependencies in this protocol.

5.79 RFC 2794 Mobile IP Network Access Identifier Extension for IPv4

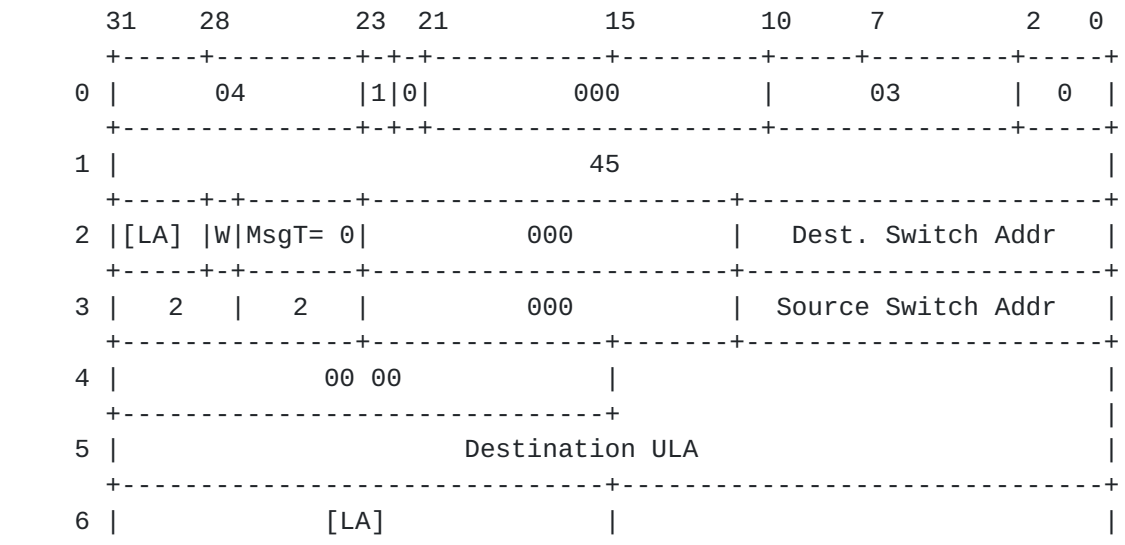
This document defines an IPv4 specific protocol and a similar functionality must be defined for Mobile IPv6.

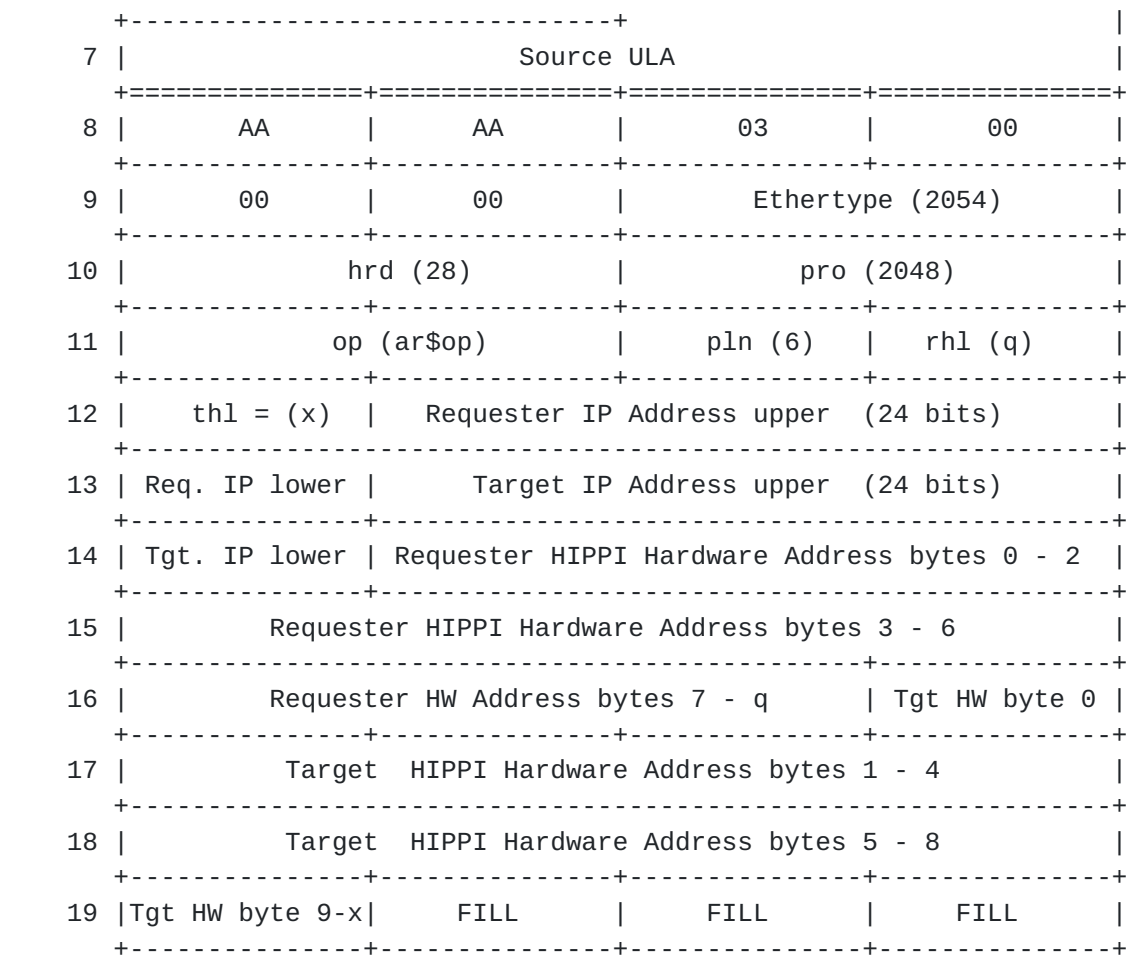
5.80 RFC 2834 ARP and IP Broadcast over HIPPI-800

This document uses the generic term "IP Address" in the text but
it also contains the text:

> The HARP message has several fields that have the following format
> and values:
>
> Data sizes and field meaning:
>
> | | | |
> |---|---|---|
> | ar$hrd | 16 bits | Hardware type |
> | ar$pro | 16 bits | Protocol type of the protocol fields below |
> | ar$op | 16 bits | Operation code (request, reply, or NAK) |
> | ar$pln | 8 bits | byte length of each protocol address |
> | ar$rhl | 8 bits | requester's HIPPI hardware address length (q) |
> | ar$thl | 8 bits | target's HIPPI hardware address length (x) |
> | ar$rpa | 32 bits | requester's protocol address |
> | ar$tpa | 32 bits | target's protocol address |
> | ar$rha | qbytes | requester's HIPPI Hardware address |
> | ar$tha | xbytes | target's HIPPI Hardware address |
>
> Where :
>
> ar$hrd - SHALL contain 28. (HIPARP)
>
> ar$pro - SHALL contain the IP protocol code 2048 (decimal).
>
> ar$op - SHALL contain the operational value (decimal):
>
> | | |
> |---|---|
> | 1 | for HARP_REQUESTs |
> | 2 | for HARP_REPLYs |
> | 8 | for InHARP_REQUESTs |
> | 9 | for InHARP_REPLYs |
> | 10 | for HARP_NAK |
>
> ar$pln - SHALL contain 4.

and later:





HARP - InHARP Message

Which make this protocol only IPv4 aware. An update is required to support IPv6.

5.81 RFC 2835 IP and ARP over HIPPI-6400 (GSN) (GSN)

This document states:

> The Ethertype value SHALL be set as defined in Assigned Numbers [18]:
>
> IP 0x0800 2048 (16 bits)

This is IPv4 limited and as expected (after reviewing the previous section) requires an update to support IPv6. There are numerous other points in the documents that confirms this assumption.

5.82 RFC 2855 DHCP for IEEE 1394

This document is only designated for IPv4. It is expected that similar functionality is available in DHCPv6.

5.83 RFC 2874 DNS Extensions to Support IPv6 Address Aggregation

and Renumbering

This document defines a protocol to interact with IPv6 and is not considered in this document.

5.84 RFC 2893 Transition Mechanisms for IPv6 Hosts and Routers (TRANS-IPV6)

This document defines a transition mechanism for IPv6 and is not considered in this document.

5.85 RFC 2915 The Naming Authority Pointer (NAPTR) DNS Resource Record (NAPTR)

There are no IPv4 dependencies in this protocol.

5.86 RFC 2916 E.164 number and DNS

There are no IPv4 dependencies in this protocol.

5.87 RFC 2937 The Name Service Search Option for DHCP

This document is only designated for IPv4. It is expected that similar functionality is available in DHCPv6.

5.88 RFC 3004 The User Class Option for DHCP

This document is only designated for IPv4. It is expected that similar functionality is available in DHCPv6.

5.89 RFC 3011 The IPv4 Subnet Selection Option for DHCP

This document is specifically designed for IPv4.

5.90 RFC 3021 Using 31-Bit Prefixes on IPv4 Point-to-Point Links

This document is IPv4 specific and a similar technique could also be defined for IPv6.

5.91 RFC 3024 Reverse Tunneling for Mobile IP, revised

This protocol assumes IPv4 addressing. An updated Mobile IPv6 specification should include this functionality.

5.92 RFC 3046 DHCP Relay Agent Information Option

This document is only designated for IPv4. It is expected that similar functionality is available in DHCPv6.

5.93 RFC 3056 Connection of IPv6 Domains via IPv4 Clouds

This is an IPv6 related document and is not discussed in this document.

5.94 RFC 3068 An Anycast Prefix for 6to4 Relay Routers

This is an IPv6 related document and is not discussed in this document.

5.95 RFC 3074 DHC Load Balancing Algorithm

There are no IPv4 dependencies in this protocol.

5.96 RFC 3077 A Link-Layer Tunneling Mechanism for Unidirectional Links

This protocol is both IPv4 and IPv6 aware and needs no changes.

5.97 RFC 3115 Mobile IP Vendor/Organization-Specific Extensions

This is an enhancement for Mobile IPv4. It is expected that a similar capability will be in Mobile IPv6.

5.98 RFC 3145 L2TP Disconnect Cause Information

There are no IPv4 dependencies in this protocol.

6.0 Experimental RFCs

Experimental RFCs typically define protocols that do not have widescale implementation or usage on the Internet. They are often propriety in nature or used in limited arenas. They are documented to the Internet community in order to allow potential interoperability or some other potential useful scenario. In a few cases they are presented as alternatives to the mainstream solution to an acknowledged problem.

6.01 RFC 1183 New DNS RR Definitions (DNS-RR)

There are no IPv4 dependencies in this protocol.

6.02 RFC 1226 Internet protocol encapsulation of AX.25 frames (IP-AX.25)

There are no IPv4 dependencies in this protocol.

6.03 RFC 1241 Scheme for an internet encapsulation protocol: Version 1 (IN-ENCAP)

This protocol specifies a protocol that assumes IPv4 but does not actually have any limitations which would limit its operation in an IPv6 environment.

6.04 RFC 1393 Traceroute Using an IP Option (TRACE-IP)

This document uses an IPv4 option. It is therefore limited to IPv4 networks. A different technique must be developed for IPv6.

6.05 RFC 1433 Directed ARP (DIR-ARP)

There are no IPv4 dependencies in this protocol.

6.06 RFC 1464 Using the Domain Name System To Store Arbitrary String Attributes

There are no IPv4 dependencies in this protocol.

6.07 RFC 1475 TP/IX: The Next Internet (TP-IX)

This document defines IPv7 and has been abandoned by the IETF as a feasible design. It is not considered in this document.

6.08 RFC 1561 Use of ISO CLNP in TUBA Environments (CLNP-TUBA)

This document defines the use of NSAPA addressing and does not use any version of IP, so there are no IPv4 dependencies in this protocol.

6.09 RFC 1712 DNS Encoding of Geographical Location (DNS-ENCODE)

There are no IPv4 dependencies in this protocol.

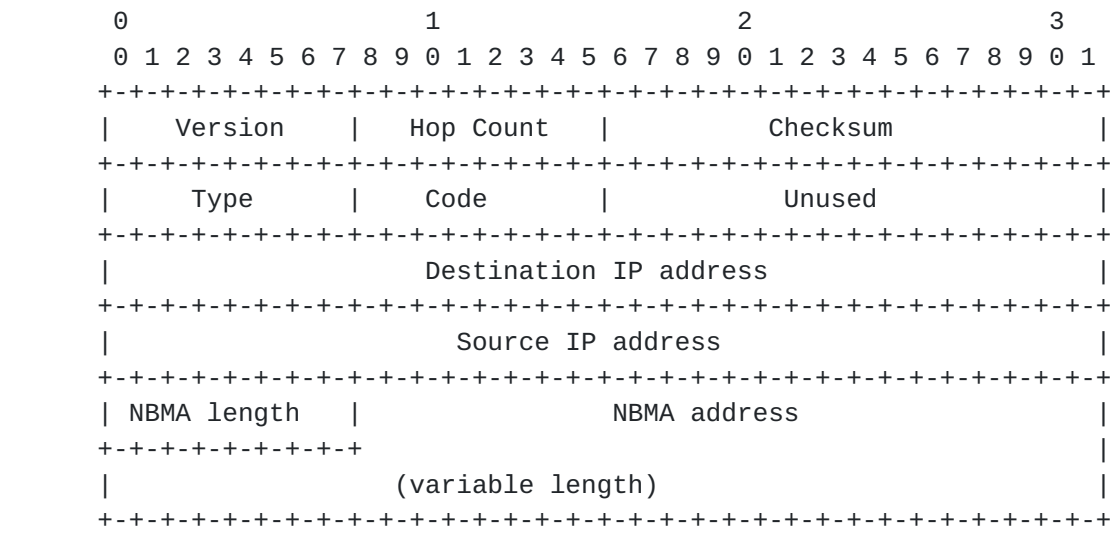
6.10 RFC 1735 NBMA Address Resolution Protocol (NARP) (NARP)

This document defines a protocol that is IPv4 specific. A new version would need to be documented to support IPv6.

4. Packet Formats

NARP requests and replies are carried in IP packets as protocol type 54. This section describes the packet formats of NARP requests and replies:

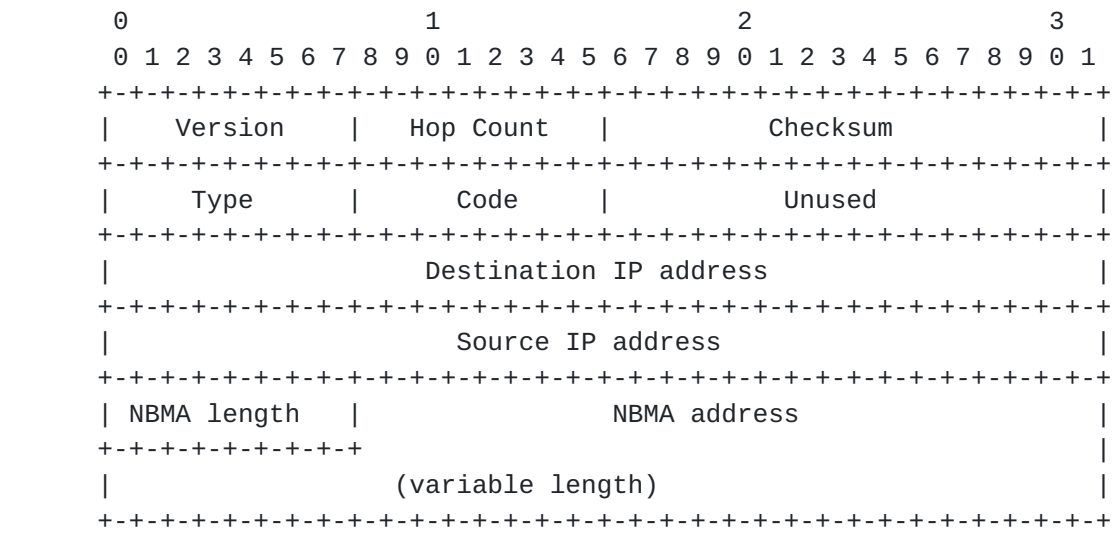
NARP Request



Source and Destination IP Addresses
Respectively, these are the IP addresses of the NARP requestor and the target terminal for which the NBMA address is desired.

and

NARP Reply



Source and Destination IP Address
Respectively, these are the IP addresses of the NARP requestor and the target terminal for which the NBMA address is desired.

6.11 RFC 1768 Host Group Extensions for CLNP Multicasting (CLNP-MULT)

This document defines an IPv9 multicast protocol and has been abandoned by the IETF as a feasible design. It is not considered in this document.

6.12 RFC 1788 ICMP Domain Name Messages (ICMP-DM)

This protocol is used for updates to the in-addr.arp reverse DNS maps, and is limited to IPv4.

6.13 RFC 1797 Class A Subnet Experiment

This document is specific to IPv4.

6.14 RFC 1819 Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+ (ST2)

This protocol is IPv4 limited. In fact it is the definition of IPv5. See below.

Both ST2 and IP apply the same addressing schemes to identify different hosts. ST2 and IP packets differ in the first four bits, which contain the internetwork protocol version number: number 5 is reserved for ST2 (IP itself has version number 4). As a network layer protocol, like IP, ST2 operates independently of its underlying subnets. Existing implementations use ARP for address resolution, and use the same Layer 2 SAPs as IP.

8.2 Group Name Generator

GroupName generation is similar to Stream ID generation. The GroupName includes a 16-bit unique identifier, a 32-bit creation timestamp, and a 32-bit IP address. Group names are globally unique. A GroupName includes the creator's IP address, so this reduces a global uniqueness problem to a simple local problem.

IP-encapsulated ST packets begin with a normal IP header. Most fields of the IP header should be filled in according to the same rules that apply to any other IP packet. Three fields of special interest are:

o Protocol is 5, see [RFC1700], to indicate an ST packet is enclosed, as opposed to TCP or UDP, for example.

and

The following permanent IP multicast addresses have been assigned to ST:

224.0.0.7 All ST routers (intermediate agents)
224.0.0.8 All ST hosts (agents)

In addition, a block of transient IP multicast addresses, 224.1.0.0 - 224.1.255.255, has been allocated for ST multicast groups. For instance, the following two functions could be made available:

The ST Header also includes an ST Version Number, a total length field, a header checksum, a unique id, and the stream origin 32-bit IP address. The unique id and the stream origin 32-bit IP address form the stream id (SID). This is shown in Figure 10. Please refer to Section 10.6 for an explanation of the notation.

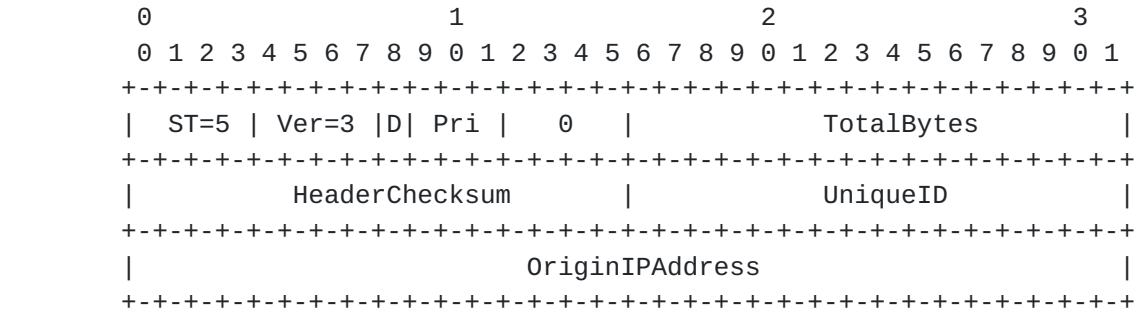


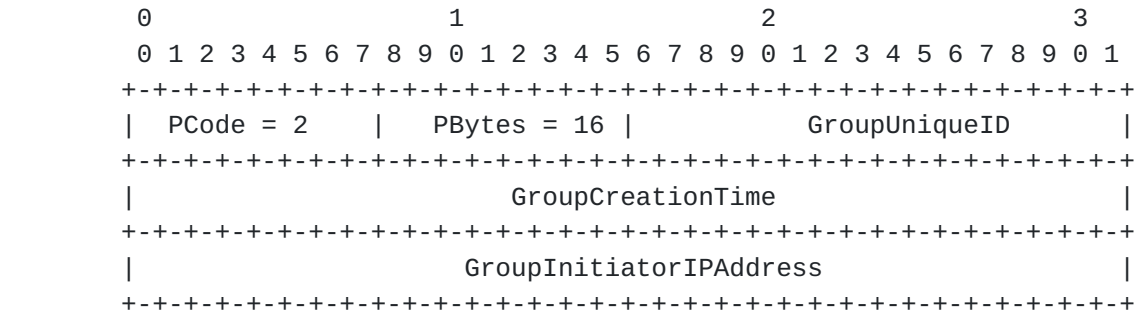
Figure 10: ST Header

o ST is the IP Version Number assigned to identify ST packets. The value for ST is 5.

o OriginIPAddress is the second element of the SID. It is the 32-bit IP address of the stream origin, see Section 8.1.

10.3.2 Group

The Group parameter (PCode = 2) is an optional argument used to indicate that the stream is a member in the specified group.



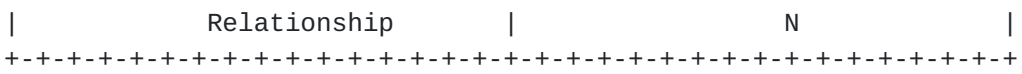


Figure 13: Group Parameter

o GroupUniqueID, GroupInitiatorIPAddress, and GroupCreationTime together form the GroupName field. They are allocated by the group name generator function, see Section 8.2. GroupUniqueID and GroupCreationTime are implementation specific and have only local definitions.

10.3.3 MulticastAddress

The MulticastAddress parameter (PCode = 3) is an optional parameter that is used when using IP encapsulation and setting up an IP multicast group. This parameter is used to communicate the desired IP multicast address to next-hop ST agents that should become members of the group, see Section 8.8.

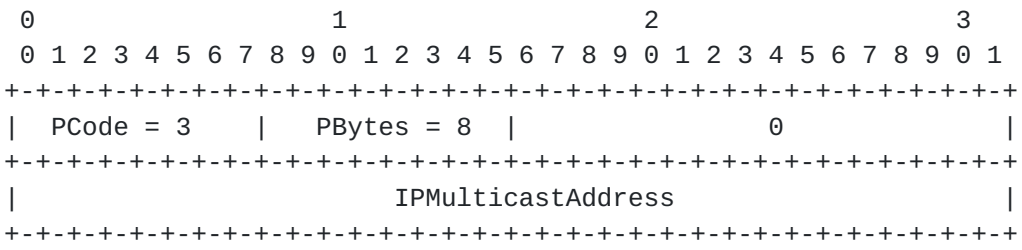


Figure 15: MulticastAddress

o IPMulticastAddress is the 32-bit IP multicast address to be used to receive data packets for the stream.

10.3.5 RecordRoute

The RecordRoute parameter (PCode = 5) is used to request that the route between the origin and a target be recorded and delivered to the user application. The ST agent at the origin (or target) including this parameter, has to determine the parameter's length, indicated by the PBytes field. ST agents processing messages containing this parameter add their receiving IP address in the position indicated by the FreeOffset field, space permitting. If no space is available, the parameter is passed unchanged. When included by the origin, all agents between the origin and the target add their IP addresses and this information is made available to the application at the target. When included by the target, all agents between the target and the origin, inclusive, add their IP addresses and this information is made available to the application at the origin.

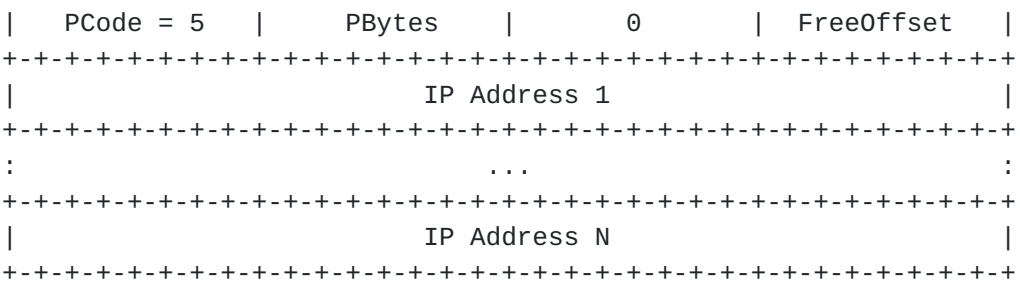


Figure 17: RecordRoute

- PBytes is the length of the parameter in bytes. Length is determined by the agent (target or origin) that first introduces the parameter. Once set, the length of the parameter remains unchanged.

- FreeOffset indicates the offset, relative to the start of the parameter, for the next IP address to be recorded. When the FreeOffset is greater than, or equal to, PBytes the RecordRoute parameter is full.

- IP Address is filled in, space permitting, by each ST agent processing this parameter.

10.3.6 Target and TargetList

Several control messages use a parameter called TargetList (PCode = 6), which contains information about the targets to which the message pertains. For each Target in the TargetList, the information includes the 32-bit IP address of the target, the SAP applicable to the next higher layer protocol, and the length of the SAP (SAPBytes). Consequently, a Target structure can be of variable length. Each entry has the format shown in Figure 18.

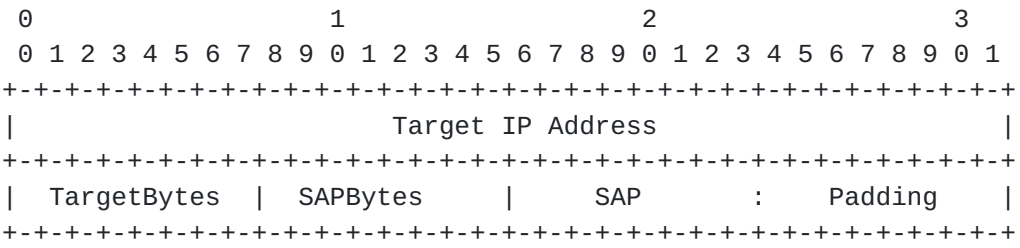


Figure 18: Target

There are many other examples, but it does not serve any purpose to include them all.

6.15 RFC 1868 ARP Extension - UNARP (UNARP)

This protocol specifies IPv4 ARP. It is expected that a similar method should be implemented in IPv6.

6.16 RFC 1876 A Means for Expressing Location Information in the Domain Name System (DNS-LOC)

This document defines a methodology for applying this technology which is IPv4 dependent. The protocol itself has no IPv4 dependencies.

6.17 RFC 1888 OSI NSAPs and IPv6

This is an IPv6 related document and is not discussed in this document.

6.18 RFC 2009 GPS-Based Addressing and Routing (GPS-AR)

The document states:

> The future version of IP (IP v6) will certainly have a sufficient number of bits in its addressing space to provide an address for even smaller GPS addressable units. In this proposal, however, we assume the current version of IP (IP v4) and we make sure that we manage the addressing space more economically than that. We will call the smallest GPS addressable unit a GPS-square.

6.19 RFC 2143 Encapsulating IP with the Small Computer System Interface (IP-SCSI)

This protocol will only operate using IPv4. As stated in the document:

> It was decided that the ten byte header offers the greatest flexibility for encapsulating version 4 IP datagrams for the following reasons:

6.20 RFC 2345 Domain Names and Company Name Retrieval

There are no IPv4 dependencies in this protocol.

6.21 RFC 2471 IPv6 Testing Address Allocation

This is an IPv6 related document and is not discussed in this document.

6.22 RFC 2481 A Proposal to add Explicit Congestion Notification (ECN) to IP (ECN-IP)

This protocol is both IPv4 and IPv6 aware and needs no changes.

6.23 RFC 2521 ICMP Security Failures Messages (ICMP-SEC)

There are no IPv4 dependencies in this protocol.

6.24 RFC 2540 Detached Domain Name System (DNS) Information (DNS-INFO)

There are no IPv4 dependencies in this protocol.

6.25 RFC 2770 GLOP Addressing in 233/8

This document is specific to IPv4.

6.26 RFC 2823 PPP over Simple Data Link (SDL) using SONET/SDH with ATM-like framing (PPP-SDL)

There are no IPv4 dependencies in this protocol.

6.27 RFC 3123 A DNS RR Type for Lists of Address Prefixes (APL RR)

This protocol is both IPv4 and IPv6 aware and needs no changes.

7.0 Summary of Results

In the initial survey of RFCs 62 positives were identified out of a total of 159, broken down as follows:

Standards	16 of 18 or 88.89%
Draft Standards	6 of 16 or 37.50%
Proposed Standards	35 of 98 or 35.71%
Experimental RFCs	5 of 27 or 18.52%

Of those identified many require no action because they document outdated and unused protocols, while others are document protocols that are actively being updated by the appropriate working groups. Additionally there are many instances of standards that SHOULD be updated but do not cause any operational impact if they are not updated. The remaining instances are documented below.

The author has attempted to organize the results in a format that allows easy reference to other protocol designers. The following recommendations uses the documented terms "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" described in RFC 2119. They should only be interpreted in the context of RFC 2119 when they appear in all caps. That is, the word "should" in

the previous SHOULD NOT be interpreted as in RFC 2119.

The assignment of these terms has been based entirely on the authors perceived needs for updates and should not be taken as an official statement.

7.1 Standards

7.1.01 STD3 Requirements for Internet Hosts (RFC 1122 & 1123)

RFC 1122 is essentially a requirements document for IPv4 hosts and a similar document for IPv6 hosts SHOULD be written.

RFC 1123 SHOULD be updated to include advances in application protocols, as well as tightening language regarding IP addressing.

7.1.02 STD 5 Internet Protocol (RFC 791, 922, 792, & 1122)

RFC 791 has been updated in the definition of IPv6 in RFC 2460.

RFC 922 has been included in the IPv6 Addressing Architecture, RFC 2373.

RFC 792 has been updated in the definition of ICMPv6 in RFC 2463.

RFC 1122 has been updated in the definition of Multicast Listener Discovery in RFC 2710.

7.1.03 STD 13 Domain Name System (RFCs 1034 & 1035)

New resource records for IPv6 addresses have been defined (AAAA & A6).

7.1.04 STD 41 IP over Ethernet (RFC 894)

This problem has been fixed by RFC2464, A Method for the Transmission of IPv6 Packets over Ethernet Networks.

7.1.05 STD 42 IP over Experimental Ethernets (RFC 895)

See above section.

7.1.06 STD 43 IP over IEEE 8.02 (RFC 1042)

The functionality of this RFC is included in subsequent standards defining IPv6 over XXX.

7.1.07 STD 44 DCN Networks (RFC 891)

This protocol is no longer used and an updated protocol SHOULD NOT be created.

7.1.08 STD 45 IP over HyperChannel (RFC 1044)

No updated document exists for this protocol. It is unclear whether one is needed. An updated protocol MAY be created.

7.1.09 STD 46 IP over Arcnet (RFC 1201)

This problem has been fixed by RFC 2497, A Method for the Transmission of IPv6 Packets over ARCnet Networks.

7.1.10 STD 48 IP over Netbios (RFC 1088)

A new protocol specification for tunneling IPv6 packets through Netbios networks SHOULD be defined.

7.1.11 STD 52 IP over SMDS (RFC 1209)

An updated protocol for the transmission of IPv6 packets over SMDS MUST be written.

7.2 Draft Standards

7.2.1 Boot Protocol (RFC 951)

This problem has been fixed in the DHCPv6 and Auto Configuration protocols of IPv6: RFC 2462: IPv6 Stateless Address Autoconfiguration, and Dynamic Host Configuration Protocol for IPv6 (DHCPv6) currently an Internet Draft.

7.2.2 Path MTU Discovery (RFC 1191)

This problem has been fixed in RFC 1981, Path MTU Discovery for IP version 6.

7.2.3 The PPP Multilink Protocol (RFC 1990)

A new class identifier for IPv6 packets MUST be registered with the IANA. It is RECOMMENDED that the (currently unassigned) value of

6 be assigned by the IANA with a description of "Internet Protocol (IPv6) Address." An application for this assignment has been sent to the IANA.

7.2.4 IP over HIPPI (RFC 2067)

An updated protocol for the transmission of IPv6 packets over HIPPI MAY be written.

7.2.5 DHCP (RFC 2131)

The problems are being fixed by the work of the DHC WG. Several very advanced IDs address all the issues.

7.2.6 DHCP Options (RFC 2132)

The problems are being fixed by the work of the DHC WG. Several very advanced IDs address all the issues.

7.3 Proposed Standards

7.3.01 Tunneling IPX over IP (RFC 1234)

This problem remains unresolved and a new protocol specification MUST be created.

7.3.02 ICMP Router Discovery (RFC 1256)

This problem has been resolved in RFC 2461, Neighbor Discovery for IP Version 6 (IPv6)

7.3.03 Encoding Net Addresses to Support Operation Over Non OSI Lower Layers (RFC 1277)

This problem is unresolved, but it MAY be resolved with the creation of a new encoding scheme definition.

7.3.04 PPP Internet Protocol Control Protocol (RFC 1332)

This problem has been resolved in RFC 2472, IP Version 6 over PPP.

7.3.05 IP Multicast over Token Ring (RFC 1469)

The functionality of this specification has been essentially covered in RFC 2470, IPv6 over Token Ring in section 8.

7.3.06 IP Mobility Support (RFC 2002)

The problems are being resolved by the Mobile IP WG and there is a mature ID (draft-ietf-mobileip-ipv6-15.txt)

7.3.07 IP Encapsulation within IP (RFC 2003)

This functionality for Mobile IPv6 is accomplished using the Routing Header as defined in RFC 2460, Internet Protocol, Version 6 (IPv6) Specification.

7.3.08 Minimal Encapsulation within IP (RFC 2004)

See Section 7.3.27

7.3.09 Applicability Statement for IP Mobility Support (2005)

See Section 7.3.26

7.3.10 IP Router Alert Option (RFC 2113)

The problems identified are resolved in RFC 2711, IPv6 Router Alert Option.

7.3.11 SLP (RFC 2165)

The problems have been addressed in RFC 3111, Service Location Protocol Modifications for IPv6.

7.3.12 Classical IP & ARP over ATM (RFC 2225)

The problems have been resolved in RFC 2492, IPv6 over ATM Networks.

7.3.13 IP Broadcast over ATM (RFC 2226)

The problems have been resolved in RFC 2492, IPv6 over ATM Networks.

7.3.14 IGMPv2 (RFC 2236)

The problems have been resolved in RFC 2710, Multicast Listener Discovery (MLD) for IPv6.

7.3.15 DHCP Options for NDS (RFC 2241)

The problems are being fixed by the work of the DHC WG. Several very advanced IDs address all the issues.

7.3.16 Netware/IP Domain Name and Information (RFC 2242)

The problems are being fixed by the work of the DHC WG. Several very advanced IDs address all the issues.

7.3.17 Mobile IPv4 Comfit Options for PPP IPCP (RFC 2290)

The problems are not being addressed and MUST be addressed in a new protocol.

7.3.18 Transaction IP v3 (RFC 2371)

The problems identified are not addressed and a new standard MAY be defined.

7.3.19 DHCP Option for Open Group User Authentication Protocol (RFC 2485)

The problems are being fixed by the work of the DHC WG. Several very advanced IDs address all the issues.

7.3.20 DHCP Option to Disable Stateless Autoconfiguration (RFC 2563)

The problems are being fixed by the work of the DHC WG. Several very advanced IDs address all the issues.

7.3.21 Non-Terminal DNS Redirection (RFC 2672)

The problems have not been addressed and a new specification MAY be defined.

7.3.22 Binary Labels in DNS (RFC 2673)

The problems have not been addressed and a new specification MAY be defined.

7.3.23 IP over Vertical Blanking Interval of a TV Signal (RFC 2728)

The problems have not been addressed and a new specification MAY be defined.

7.3.24 IPv4 over IEEE 1394 (RFC 2734)

This problem is being addressed by the IPv6 WG and an ID exists (draft-ietf-ipngwg-ipngwg-1394-02.txt).

7.3.25 Mobile IP Network Access Identity Extensions for IPv4 (RFC 2794)

The problems are not being addressed and MUST be addressed in a new protocol.

7.3.26 ARP & IP Broadcasts Over HIPPI 800 (RFC 2834)

The problems are not being addressed and MAY be addressed in a new protocol.

7.3.27 ARP & IP Broadcasts Over HIPPI 6400 (RFC 2835)

The problems are not being addressed and MAY be addressed in a new protocol.

7.3.28 DHCP for IEEE 1394 (RFC 2855)

This problem is being dually addressed by the IPv6 and DHC WGs and IDs exists that address this issue.

7.3.29 DHCP Name Server Search Option (RFC 2937)

The problem is being fixed by the work of the DHC WG. Several very advanced IDs address all the issues.

7.3.30 DHCP User Class Option (RFC 3004)

The problem is being fixed by the work of the DHC WG. Several very advanced IDs address all the issues.

7.3.31 IPv4 Subnet Selection DHCP Option (RFC 3011)

The problem is being fixed by the work of the DHC WG. Several very

advanced IDs address all the issues.

7.3.32 Using 31-Bit Prefixes for IPv4 P2P Links (RFC 3021)

No action is needed.

7.3.33 Reverse Tunneling for Mobile IP (RFC 3024)

The problems are not being addressed and MUST be addressed in a new protocol.

7.3.34 DHCP Relay Agent Information Option (RFC 3046)

The problem is being fixed by the work of the DHC WG. Several very advanced IDs address all the issues.

7.3.35 Mobile IP Vender/Organization Specific Extensions (RFC 3115)

The problems are not being addressed and MUST be addressed in a new protocol.

7.4 Experimental RFCs

7.4.1 Traceroute using an IP Option (RFC 1393)

This protocol relies on IPv4 and a new protocol standard MAY be produced.

7.4.2 NBMA ARP (RFC 1735)

This functionality has been defined in RFC 2491, IPv6 over Non-Broadcast Multiple Access (NBMA) networks and RFC 2332, NBMA Next Hop Resolution Protocol.

7.4.3 ST2+ Protocol (RFC 1819)

This protocol relies on IPv4 and a new protocol standard MAY be produced.

7.4.4 ARP Extensions (RFC 1868)

This protocol relies on IPv4 and a new protocol standard MAY be produced.

7.4.5 IP Over SCSI (RFC 2143)

This protocol relies on IPv4 and a new protocol standard MAY be produced.

8.0 Acknowledgements

The author would like to acknowledge the support of the Internet Society in the research and production of this document. Additionally the author would like to thanks his partner in all ways, Wendy M. Nesser.

9.0 Authors Address

Please contact the author with any questions, comments or suggestions at:

Philip J. Nesser II
Principal
Nesser & Nesser Consulting
13501 100th Ave NE, #5202
Kirkland, WA 98034

Email: phil@nesser.com
Phone: +1 425 481 4303
Fax: +1 425 48