Network Working Group   Philip J. Nesser II
draft-ietf-v6ops-ipv4survey-trans-00.txt   Nesser & Nesser Consulting
Internet Draft   February 2003
                                                       Expires August 2003

Survey of IPv4 Addresses in Currently Deployed IETF Transport Area Standards

Abstract

This document seeks to document all usage of IPv4 addresses in currently deployed IETF Transport 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.

3.1 RFC 768 User Datagram Protocol

Although UDP is a transport protocol there is one reference to the UDP/IP interface that states; "The UDP module must be able to determine the source and destination internet addresses and the protocol field from the internet header." This does not force a rewrite of the protocol but will clearly cause changes in implementations.

3.2 RFC 793 Transmission Control Protocol

Section 3.1 which specifies the header format for TCP. The TCP header is free from IPv4 references but there is an inconsistency in the computation of checksums. The text says: "The checksum also covers a 96 bit pseudo header conceptually prefixed to the TCP header. This pseudo header contains the Source Address, the Destination Address, the Protocol, and TCP length." The first and second 32-bit words are clearly meant to specify 32-bit IPv4 addresses. While no modification of the TCP protocol is necessitated by this problem, an alternate needs to be specified as an update document, or as part of another IPv6 document.

3.3 NetBIOS Service Protocols. RFC1001, RFC1002

3.3.1 RFC 1001 PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT:

CONCEPTS AND METHODS

Section 15.4.1. RELEASE BY B NODES defines:

A NAME RELEASE DEMAND contains the following information:

- NetBIOS name
- The scope of the NetBIOS name
- Name type: unique or group
- IP address of the releasing node
- Transaction ID

Section 15.4.2. RELEASE BY P NODES defines:

A NAME RELEASE REQUEST contains the following information:

- NetBIOS name
- The scope of the NetBIOS name
- Name type: unique or group
- IP address of the releasing node
- Transaction ID

A NAME RELEASE RESPONSE contains the following information:

- NetBIOS name
- The scope of the NetBIOS name
- Name type: unique or group
- IP address of the releasing node
- Transaction ID
- Result:
  - Yes: name was released
  - No: name was not released, a reason code is provided

Section 16. NetBIOS SESSION SERVICE states:

The NetBIOS session service begins after one or more IP addresses have been found for the target name. These addresses may have been acquired using the NetBIOS name query transactions or by other means, such as a local name table or cache.

Section 16.1. OVERVIEW OF NetBIOS SESSION SERVICE

Session service has three phases:

Session establishment - it is during this phase that the IP address and TCP port of the called name is determined, and a TCP connection is established with the remote party.

16.1.1. SESSION ESTABLISHMENT PHASE OVERVIEW

An end-node begins establishment of a session to another node by somehow acquiring (perhaps using the name query transactions or a local cache) the IP address of the node or nodes purported to own the destination name.

Once the TCP connection is open, the calling node sends session service request packet. This packet contains the following information:

- Calling IP address (see note)
- Calling NetBIOS name
- Called IP address (see note)
- Called NetBIOS name

NOTE: The IP addresses are obtained from the TCP service interface.

If a compatible LISTEN exists, and there are adequate resources, then the session server may transform the existing TCP connection into the NetBIOS data session. Alternatively, the session server may redirect, or "retarget" the caller to another TCP port (and IP address).

If the caller is redirected, the caller begins the session establishment anew, but using the new IP address and TCP port given in the retarget response. Again a TCP connection is created, and again the calling and called node exchange credentials. The called party may accept the call, reject the call, or make a further redirection.

17.1. OVERVIEW OF NetBIOS DATAGRAM SERVICE

Every NetBIOS datagram has a named destination and source. To transmit a NetBIOS datagram, the datagram service must perform a name query operation to learn the IP address and the attributes of the destination NetBIOS name. (This information may be cached to avoid the overhead of name query on subsequent NetBIOS datagrams.)

17.1.1. UNICAST, MULTICAST, AND BROADCAST

NetBIOS datagrams may be unicast, multicast, or broadcast. A NetBIOS datagram addressed to a unique NetBIOS name is unicast. A NetBIOS datagram addressed to a group NetBIOS name, whether there are zero, one, or more actual members, is multicast. A NetBIOS datagram sent using the NetBIOS "Send Broadcast Datagram" primitive is broadcast.

17.1.2. FRAGMENTATION OF NetBIOS DATAGRAMS

When the header and data of a NetBIOS datagram exceeds the maximum amount of data allowed in a UDP packet, the NetBIOS datagram must be fragmented before transmission and reassembled upon receipt.

A NetBIOS Datagram is composed of the following protocol elements:

- IP header of 20 bytes (minimum)
- UDP header of 8 bytes
- NetBIOS Datagram Header of 14 bytes
- The NetBIOS Datagram data.

18. NODE CONFIGURATION PARAMETERS

- B NODES:
  - Node's permanent unique name
  - Whether IGMP is in use
  - Broadcast IP address to use
  - Whether NetBIOS session keep-alives are needed
  - Usable UDP data field length (to control fragmentation)
- P NODES:
  - Node's permanent unique name
  - IP address of NBNS
  - IP address of NBDD
  - Whether NetBIOS session keep-alives are needed
  - Usable UDP data field length (to control fragmentation)
- M NODES:
  - Node's permanent unique name
  - Whether IGMP is in use
  - Broadcast IP address to use
  - IP address of NBNS
  - IP address of NBDD
  - Whether NetBIOS session keep-alives are needed
  - Usable UDP data field length (to control fragmentation)

All of the proceeding sections make implicit use of IPv4 addresses and
a new specification should be defined for use of IPv6 underlying addresses.

3.3.2 RFC 1002 PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT:
DETAILED SPECIFICATIONS

Section 4.2.1.3. RESOURCE RECORD defines

```
   RESOURCE RECORD RR_TYPE field definitions:

   Symbol      Value   Description:

   A          0x0001   IP address Resource Record (See REDIRECT NAME
                       QUERY RESPONSE)
```

Sections 4.2.2. NAME REGISTRATION REQUEST, 4.2.3. NAME OVERWRITE
REQUEST & DEMAND, 4.2.4. NAME REFRESH REQUEST, 4.2.5. POSITIVE NAME
REGISTRATION RESPONSE, 4.2.6. NEGATIVE NAME REGISTRATION RESPONSE,

4.2.7. END-NODE CHALLENGE REGISTRATION RESPONSE, 4.2.9. NAME RELEASE REQUEST & DEMAND, 4.2.10. POSITIVE NAME RELEASE RESPONSE, **4.2.11. NEGATIVE NAME RELEASE RESPONSE and Sections 4.2.13.** POSITIVE NAME QUERY RESPONSEall contain 32 bit fields labeled "NB_ADDRESS" clearly defined for IPv4 addresses

Sections 4.2.15. REDIRECT NAME QUERY RESPONSE contains a field
"NSD_IP_ADDR"
which also is designed for a IPv4 address.

Section 4.3.5. SESSION RETARGET RESPONSE PACKET

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      TYPE     |     FLAGS     |            LENGTH             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      RETARGET_IP_ADDRESS                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           PORT                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Section 4.4.1. NetBIOS DATAGRAM HEADER

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   MSG_TYPE    |     FLAGS     |           DGM_ID              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           SOURCE_IP                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          SOURCE_PORT          |          DGM_LENGTH           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         PACKET_OFFSET         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.4.2. DIRECT_UNIQUE, DIRECT_GROUP, & BROADCAST DATAGRAM

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   MSG_TYPE    |     FLAGS     |           DGM_ID              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           SOURCE_IP                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          SOURCE_PORT          |          DGM_LENGTH           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         PACKET_OFFSET         |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
   |                                                               |
   /                          SOURCE_NAME                          /
   /                                                               /
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   /                       DESTINATION_NAME                        /
   /                                                               /
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   /                           USER_DATA                           /
   /                                                               /
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Section   4.4.3. DATAGRAM ERROR PACKET

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   MSG_TYPE    |     FLAGS     |           DGM_ID              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           SOURCE_IP                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          SOURCE_PORT          |  ERROR_CODE   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.4.4. DATAGRAM QUERY REQUEST

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   MSG_TYPE    |     FLAGS     |           DGM_ID              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           SOURCE_IP                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          SOURCE_PORT          |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
   |                                                               |
   /                       DESTINATION_NAME                        /
   /                                                               /
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.4.5. DATAGRAM POSITIVE AND NEGATIVE QUERY RESPONSE

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   MSG_TYPE    |     FLAGS     |           DGM_ID              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           SOURCE_IP                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          SOURCE_PORT          |                                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
   |                                                                 |
   /                       DESTINATION_NAME                          /
   /                                                                 /
   |                                                                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.3. NetBIOS DATAGRAM SERVICE PROTOCOLS

The following are GLOBAL variables and should be NetBIOS user configurable:

- BROADCAST_ADDRESS: the IP address B-nodes use to send datagrams with group name destinations and broadcast datagrams. The default is the IP broadcast address for a single IP network.

There is also a large amount of pseudo code for most of the protocols functionality that make no specific reference to IPv4 addresses. However they assume the use of the above defined packets. The pseudo code may be valid for IPv6 as long as the packet formats are updated.

3.4 RFC 1006 ISO Transport Service on top of the TCP (Version: 3)

Section 5. The Protocol defines a mapping specification

Mapping parameters is also straight-forward:

network service -------	TCP ---
CONNECTION RELEASE	
Called address	server's IP address (4 octets)
Calling address	client's IP address (4 octets)

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.

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 1144 Compressing TCP/IP headers for low-speed serial links (IP-CMPRS)

This RFC is specifically oriented towards TCP/IPv4 packet headers and will not work in it's current form. Significant work has already been done on similar algorithms for TCP/IPv6 headers.

5.02 RFC 1323 TCP Extensions for High Performance (TCP-EXT)

There are no IPv4 dependencies in this protocol.

5.03 RFC 1553 Compressing IPX Headers Over WAN Media (CIPX) (CIPX)

There are no IPv4 dependencies in this protocol.

5.04 RFC 1692 Transport Multiplexing Protocol (TMux) (TMUX)

Section 6. Implementation Notes is states:

> Because the TMux mini-header does not contain a TOS field, only segments with the same IP TOS field should be contained in a single TMux message. As most systems do not use the TOS feature, this is not a major restriction. Where the TOS field is used, it may be desirable to hold several messages under construction for a host, one for each TOS value.
>
> Segments containing IP options should not be multiplexed.

This is clearly IPv4 specific, but a simple restatement in IPv6 terms will allow complete functionality.

5.05 RFC 1831 RPC: Remote Procedure Call Protocol Specification Version 2 RPC

There are no IPv4 dependencies in this protocol.

5.06 RFC 1833 Binding Protocols for ONC RPC Version 2

In Section 2.1 RPCBIND Protocol Specification (in RPC Language) there is the following code fragment:

```
 * Protocol family (r_nc_protofmly):
 *   This identifies the family to which the protocol belongs.  The
 *   following values are defined:
 *     NC_NOPROTOFMLY   "-"
 *     NC_LOOPBACK      "loopback"
 *     NC_INET          "inet"
 *     NC_IMPLINK       "implink"
 *     NC_PUP           "pup"
 *     NC_CHAOS         "chaos"
 *     NC_NS            "ns"
 *     NC_NBS           "nbs"
 *     NC_ECMA          "ecma"
 *     NC_DATAKIT       "datakit"
 *     NC_CCITT         "ccitt"
 *     NC_SNA           "sna"
 *     NC_DECNET        "decnet"
 *     NC_DLI           "dli"
 *     NC_LAT           "lat"
 *     NC_HYLINK        "hylink"
 *     NC_APPLETALK     "appletalk"
 *     NC_NIT           "nit"
 *     NC_IEEE802       "ieee802"
 *     NC_OSI           "osi"
 *     NC_X25           "x25"
 *     NC_OSINET        "osinet"
 *     NC_GOSIP         "gosip"
```

It is clear that the value for NC_INET is intended for the IP protocol and is seems clear that it is IPv4 dependent.

5.07 RFC 1889 RTP: A Transport Protocol for Real-Time Applications (RTP)

In general this protocol makes many references to running on UDP over IP unicast, as well as multicast addresses. There seems to be no reason that it will run effectively over IPv6 unicast and multicast.

The only possible point is in Section A.7 Computing the RTCP Transmission Interval which contains the following code fragment:

```
       /*
        * Very first call at application start-up uses half the min
        * delay for quicker notification while still allowing some time
        * before reporting for randomization and to learn about other
        * sources so the report interval will converge to the correct
        * interval more quickly.  The average RTCP size is initialized
```

```
         * to 128 octets which is conservative (it assumes everyone else
         * is generating SRs instead of RRs: 20 IP + 8 UDP + 52 SR + 48
         * SDES CNAME).
         */
        if (initial) {
            rtcp_min_time /= 2;
            *avg_rtcp_size = 128;
        }
```

which assumes an IPv4 header length of 20 bytes. It seems a simple update to this code to check for IP version and pick a value appropriate for the IP version.

5.08 RFC 1890 RTP Profile for Audio and Video Conferences with Minimal Control (RTP-AV)

There are no IPv4 dependencies in this protocol.

5.09 RFC 1962 The PPP Compression Control Protocol (CCP) (PPP-CCP)

There are no IPv4 dependencies in this protocol.

5.10 RFC 2018 TCP Selective Acknowledgement Options (TCP-ACK)

There are no IPv4 dependencies in this protocol.

5.11 RFC 2029 RTP Payload Format of Sun's CellB Video Encoding
(RTP-CELLB)

There are no IPv4 dependencies in this protocol.

5.12 RFC 2032 RTP Payload Format for H.261 Video Streams
(RTP-H.261)

There are no IPv4 dependencies in this protocol.

5.13 RFC 2126 ISO Transport Service on top of TCP (ITOT) (ITOT)

This protocol is IPv6 aware and has no issues.

5.14 RFC 2190 RTP Payload Format for H.263 Video Streams

There are no IPv4 dependencies in this protocol.

5.15 RFC 2198 RTP Payload for Redundant Audio Data (RTP-RAD)

There are no IPv4 dependencies in this protocol.

5.16 RFC 2205 Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification (RSVP)

In Section 1. Introduction the statement is made:

> RSVP operates on top of IPv4 or IPv6, occupying the place of a transport protocol in the protocol stack.

Appendix A defines all of the header formats for RSVP and there are multiple formats for both IPv4 and IPv6.

There are no IPv4 dependencies in this protocol.

5.17 RFC 2207 RSVP Extensions for IPSEC Data Flows (RSVP-IPSEC)

The defined IPsec extensions are valid for both IPv4 & IPv6.
There are no IPv4 dependencies in this protocol.

5.18 RFC 2210 The Use of RSVP with IETF Integrated Services (RSVP-IS)

There are no IPv4 dependencies in this protocol.

5.19 RFC 2211 Specification of the Controlled-Load Network Element Service

There are no IPv4 dependencies in this protocol.

5.20 RFC 2212 Specification of Guaranteed Quality of Service (GQOS)

There are no IPv4 dependencies in this protocol.

5.21 RFC 2215 General Characterization Parameters for Integrated Service Network Elements

There are no IPv4 dependencies in this protocol.

5.22 RFC 2250 RTP Payload Format for MPEG1/MPEG2 Video (RTP-MPEG)

There are no IPv4 dependencies in this protocol.

5.23 RFC 2326 Real Time Streaming Protocol (RTSP) (RTSP)

Section 3.2 RTSP URL defines:

> The "rtsp" and "rtspu" schemes are used to refer to network resources via the RTSP protocol. This section defines the scheme-specific syntax and semantics for RTSP URLs.

```
rtsp_URL  =   ( "rtsp:" | "rtspu:" )
              "//" host [ ":" port ] [ abs_path ]
host      =   <A legal Internet host domain name of IP address
              (in dotted decimal form), as defined by Section 2.1
              of RFC 1123 \cite{rfc1123}>
port      =   *DIGIT
```

Although later in that section the following text is added:

> The use of IP addresses in URLs SHOULD be avoided whenever possible (see RFC 1924 [19]).

Some later examples show:

```
  Example:

     C->S: DESCRIBE rtsp://server.example.com/fizzle/foo RTSP/1.0
           CSeq: 312
           Accept: application/sdp, application/rtsl, application/mheg

     S->C: RTSP/1.0 200 OK
           CSeq: 312
           Date: 23 Jan 1997 15:35:06 GMT
           Content-Type: application/sdp
           Content-Length: 376

           v=0
           o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
           s=SDP Seminar
           i=A Seminar on the session description protocol
           u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
           e=mjh@isi.edu (Mark Handley)
           c=IN IP4 224.2.17.12/127
           t=2873397496 2873404696
           a=recvonly
           m=audio 3456 RTP/AVP 0
           m=video 2232 RTP/AVP 31
           m=whiteboard 32416 UDP WB
           a=orient:portrait
```

which implies the use of the "IP4" tag and it should be possible to use an "IP6" tag. There are also numerous other similar examples using the "IP4" tag.

There seems to be nothing that requires IPv4, and a small set of updates can be created to document IPv6 functionality.

5.24 RFC 2327 SDP: Session Description Protocol (SDP)

Like the previous document, a sample SDP description is given as:

> An example SDP description is:

```
      v=0
      o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
      s=SDP Seminar
      i=A Seminar on the session description protocol
      u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
      e=mjh@isi.edu (Mark Handley)
      c=IN IP4 224.2.17.12/127
      t=2873397496 2873404696
      a=recvonly
      m=audio 49170 RTP/AVP 0
      m=video 51372 RTP/AVP 31
      m=application 32416 udp wb
      a=orient:portrait
```

Later an explicit discussion of the network addressing scheme is given:

> <network type> is a text string giving the type of network. Initially "IN" is defined to have the meaning "Internet". <address type> is a text string giving the type of the address that follows. Initially "IP4" and "IP6" are defined. <address> is the globally unique address of the machine from which the session was created. For an address type of IP4, this is either the fully-qualified domain name of the machine, or the dotted-decimal representation of the IP version 4 address of the machine. For an address type of IP6, this is either the fully-qualified domain name of the machine, or the compressed textual representation of the IP version 6 address of the machine. For both IP4 and IP6, the fully-qualified domain name is the form that SHOULD be given unless this is unavailable, in which case the globally unique address may be substituted. A local IP address MUST NOT be used in any context where the SDP description might leave the scope in which the address is meaningful.

Although later in the definitions of connection types the following text is found:

> Connection Data
>
> c=<network type> <address type> <connection address>

The "c=" field contains connection data.

A session announcement must contain one "c=" field in each media description (see below) or a "c=" field at the session-level. It may contain a session-level "c=" field and one additional "c=" field per media description, in which case the per-media values override the session-level settings for the relevant media.

The first sub-field is the network type, which is a text string giving the type of network. Initially "IN" is defined to have the meaning "Internet".

The second sub-field is the address type. This allows SDP to be used for sessions that are not IP based. Currently only IP4 is defined.

The third sub-field is the connection address. Optional extra subfields may be added after the connection address depending on the value of the <address type> field.

For IP4 addresses, the connection address is defined as follows:

- Typically the connection address will be a class-D IP multicast

  group address. If the session is not multicast, then the connection address contains the fully-qualified domain name or the unicast IP address of the expected data source or data relay or data sink as determined by additional attribute fields. It is not expected that fully-qualified domain names or unicast addresses will be given in a session description that is communicated by a multicast announcement, though this is not prohibited. If a unicast data stream is to pass through a network address translator, the use of a fully-qualified domain name rather than an unicast IP address is RECOMMENDED. In other cases, the use of an IP address to specify a particular interface on a multi-homed host might be required. Thus this specification leaves the decision as to which to use up to the individual application, but all applications MUST be able to cope with receiving both formats.

- Conferences using an IP multicast connection address must also have a time to live (TTL) value present in addition to the multicast address. The TTL and the address together define the scope with which multicast packets sent in this conference will be sent. TTL values must be in the range 0-255.

  The TTL for the session is appended to the address using a slash as a separator. An example is:

  ```
  c=IN IP4 224.2.1.1/127
  ```

  Hierarchical or layered encoding schemes are data streams where the encoding from a single media source is split into a number of

> layers. The receiver can choose the desired quality (and hence bandwidth) by only subscribing to a subset of these layers. Such layered encodings are normally transmitted in multiple multicast groups to allow multicast pruning. This technique keeps unwanted traffic from sites only requiring certain levels of the hierarchy. For applications requiring multiple multicast groups, we allow the following notation to be used for the connection address:
>
> ```
> <base multicast address>/<ttl>/<number of addresses>
> ```
>
> If the number of addresses is not given it is assumed to be one. Multicast addresses so assigned are contiguously allocated above the base address, so that, for example:
>
> ```
> c=IN IP4 224.2.1.1/127/3
> ```
>
> would state that addresses 224.2.1.1, 224.2.1.2 and 224.2.1.3 are to be used at a ttl of 127. This is semantically identical to including multiple "c=" lines in a media description:
>
> ```
> c=IN IP4 224.2.1.1/127
> c=IN IP4 224.2.1.2/127
> c=IN IP4 224.2.1.3/127
> ```
>
> Multiple addresses or "c=" lines can only be specified on a per-media basis, and not for a session-level "c=" field.
>
> It is illegal for the slash notation described above to be used for IP unicast addresses.

This is probably because the definitions for IPv6 multicast was not standardized at the time of this documents production. A similar mechanism for IPv6 multicast could defined in a straightforward manner.

5.25 RFC 2380 RSVP over ATM Implementation Requirements

This protocol is both IPv4 and IPv6 aware.

5.26 RFC 2381 Interoperation of Controlled-Load Service and Guaranteed Service with ATM

There does not seem any inherent IPv4 limitations in this protocol, but it assumes work of other standards that have IPv4 limitations.

5.27 RFC 2393 IP Payload Compression Protocol (IPComp) (IPCOMP)

This protocol is both IPv4 and IPv6 aware.

5.28 RFC 2429 RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)

There are no IPv4 dependencies in this protocol.

5.29 RFC 2431 RTP Payload Format for BT.656 Video Encoding

There are no IPv4 dependencies in this protocol.

5.30 RFC 2435 RTP Payload Format for JPEG-compressed Video

There are no IPv4 dependencies in this protocol.

5.31 RFC 2474 Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers

This protocol is both IPv4 and IPv6 aware.

5.32 RFC 2508 Compressing IP/UDP/RTP Headers for Low-Speed Serial Links

This protocol is both IPv4 and IPv6 aware.

5.33 RFC 2509 IP Header Compression over PPP (IPCOM-PPP)

This protocol is both IPv4 and IPv6 aware.

5.34 RFC 2543 SIP: Session Initiation Protocol (SIP)

In Section 2 SIP Uniform Resource Locators the following specification is made:

```
  SIP-URL         = "sip:" [ userinfo "@" ] hostport
                    url-parameters [ headers ]

  hostport        = host [ ":" port ]
  host            = hostname | IPv4address
  hostname        = *( domainlabel "." ) toplabel [ "." ]

  IPv4address     = 1*digit "." 1*digit "." 1*digit "." 1*digit
```

Later it states:

> The issue of IPv6 literal addresses in URLs is being looked at elsewhere in the IETF. SIP implementers are advised to keep up to date on that activity.

Further:

> URL parameters: SIP URLs can define specific parameters of the request. URL parameters are added after the host component and are separated by semi-colons. The transport parameter determines the transport mechanism (UDP or TCP). UDP is to be assumed when no explicit transport parameter is included. The maddr parameter provides the server address to be contacted for this user, overriding the address supplied in the host field. This address is typically a multicast address, but could also be the address of a backup server. The ttl parameter determines the time-to-live value of the UDP multicast packet and MUST only be used if maddr is a multicast address and the transport protocol is UDP. The user parameter was described above. For example, to specify to call j.doe@big.com using multicast to 239.255.255.1 with a ttl of 15, the following URL would be used:

```
     sip:j.doe@big.com;maddr=239.255.255.1;ttl=15
```

and then:

```
     sip:alice@10.1.2.3
```

and in Section 4.2.6 REGISTER

> A client uses the REGISTER method to register the address listed in the To header field with a SIP server.
>
> A user agent MAY register with a local server on startup by sending a REGISTER request to the well-known "all SIP servers" multicast address "sip.mcast.net" (224.0.1.75).

There are many examples of transactions which use IPv4 only addresses. This protocol clearly needs to be updated for IPv6.

5.35 RFC 2581 TCP Congestion Control (TCP-CC)

There are no IPv4 dependencies in this protocol.

5.36 RFC 2597 Assured Forwarding PHB Group

This protocol is both IPv4 and IPv6 aware.

5.37 RFC 2598 An Expedited Forwarding PHB

This protocol is both IPv4 and IPv6 aware.

5.38 RFC 2658 RTP Payload Format for PureVoice(tm) Audio

There are no IPv4 dependencies in this protocol.

5.39 RFC 2678 IPPM Metrics for Measuring Connectivity (IPPM-MET)

This protocol only supports IPv4. An updated protocol for IPv6 will need to be defined.

5.40 RFC 2679 A One-way Delay Metric for IPPM

This protocol only supports IPv4. An updated protocol for IPv6 will need to be defined.

5.41 RFC 2680 A One-way Packet Loss Metric for IPPM

This protocol only supports IPv4. An updated protocol for IPv6 will need to be defined.

5.42 RFC 2681 A Round-trip Delay Metric for IPPM

This protocol only supports IPv4. An updated protocol for IPv6 will need to be defined.

5.43 RFC 2730 Multicast Address Dynamic Client Allocation Protocol (MADCAP) (MADCAP)

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

5.44 RFC 2733 An RTP Payload Format for Generic Forward Error Correction

This protocol is dependent on SDP which has IPv4 dependencies. Once that limitation is fixed, then this protocol should support IPv6.

5.45 RFC 2745 RSVP Diagnostic Messages

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

5.46 RFC 2746 RSVP Operation Over IP Tunnels

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

5.47 RFC 2750 RSVP Extensions for Policy Control

There are no IPv4 dependencies in this protocol.

5.48 RFC 2751 Signaled Preemption Priority Policy Element (RSVP)

There are no IPv4 dependencies in this protocol.

5.49 RFC 2752 Identity Representation for RSVP

There are no IPv4 dependencies in this protocol.

5.50 RFC 2793 RTP Payload for Text Conversation

There are no IPv4 dependencies in this protocol.

5.51 RFC 2814 SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks

This protocol claims to be both IPv4 and IPv6 aware, but all of the examples are given with IPv4 addresses. That, by itself is not a telling point but the following statement is made:

> a) LocalDSBMAddrInfo -- current DSBM's IP address (initially, 0.0.0.0) and priority. All IP addresses are assumed to be in network byte order. In addition, current DSBM's L2 address is also stored as part of this state information.

which could just be sloppy wording. Perhaps a short document clarifying the text is appropriate.

5.52 RFC 2815 Integrated Service Mappings on IEEE 802 Networks

There are no IPv4 dependencies in this protocol.

5.53 RFC 2833 RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals

There are no IPv4 dependencies in this protocol.

5.54 RFC 2848 The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services

This protocol is dependent on SDP & SIP which has IPv4 dependencies. Once these limitations are fixed, then this protocol should support IPv6.

5.55 RFC 2862 RTP Payload Format for Real-Time Pointers

There are no IPv4 dependencies in this protocol.

5.56 RFC 2872 Application and Sub Application Identity Policy Element for Use with RSVP

There are no IPv4 dependencies in this protocol.

5.57 RFC 2873 TCP Processing of the IPv4 Precedence Field

This protocol documents a technique using IPv4 headers. A similar technique, if needed, will need to be defined for IPv6.

5.58 RFC 2883 An Extension to the Selective Acknowledgement (SACK) Option for TCP (SACK)

There are no IPv4 dependencies in this protocol.

5.59 RFC 2907 MADCAP Multicast Scope Nesting State Option

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

5.60 RFC 2960 Stream Control Transmission Protocol

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

5.61 RFC 2961 RSVP Refresh Overhead Reduction Extensions

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

5.62 RFC 2976 The SIP INFO Method

There are no IPv4 dependencies in this protocol.

5.63 RFC 2988 Computing TCP's Retransmission Timer

There are no IPv4 dependencies in this protocol.

5.64 RFC 2996 Format of the RSVP DCLASS Object

There are no IPv4 dependencies in this protocol.

5.65 RFC 2997 Specification of the Null Service Type

There are no IPv4 dependencies in this protocol.

5.66 RFC 3003 The audio/mpeg Media Type

There are no IPv4 dependencies in this protocol.

5.67 RFC 3006 Integrated Services in the Presence of Compressible Flows

This document defines a protocol that discusses compressible flows, but only in an IPv4 context. When IPv6 compressible flows are defined, a similar technique should also be defined.

5.68 RFC 3010 NFS version 4 Protocol (NFSv4)

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

5.69 RFC 3015 Megaco Protocol Version 1.0 (MEGACO)

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

5.70 RFC 3016 RTP Payload Format for MPEG-4 Audio/Visual Streams

There are no IPv4 dependencies in this protocol.

5.71 RFC 3033 The Assignment of the Information Field and Protocol Identifier in the Q.2941 Generic Identifier and Q.2957 User-to-user Signaling for the Internet Protocol

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

5.72 RFC 3042 Enhancing TCP's Loss Recovery Using Limited Transmit

There are no IPv4 dependencies in this protocol.

5.73 RFC 3047 RTP Payload Format for ITU-T Recommendation G.722.1

There are no IPv4 dependencies in this protocol.

5.74 RFC 3057 ISDN Q.921-User Adaptation Layer

There are no IPv4 dependencies in this protocol.

5.75 RFC 3095 Robust Header Compression (ROHC): Framework and four profiles

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

5.76 RFC 3108 Conventions for the use of the Session Description Protocol (SDP) for ATM Bearer Connections

This protocol is currently limited to IPv4 as amplified below:

> The range and format of the <rtcpPortNum> and <rtcpIPaddr> subparameters is per [1]. The <rtcpPortNum> is a decimal number between 1024 and 65535. It is an odd number. If an even number in this range is specified, the next odd number is used. The <rtcpIPaddr> is expressed in the usual dotted decimal IP address representation, from 0.0.0.0 to 255.255.255.255.

and

<rtcpIPaddr>	IP address for receipt of RTCP packets	Dotted decimal, 7-15 chars

5.77 RFC 3119 A More Loss-Tolerant RTP Payload Format for MP3 Audio

There are no IPv4 dependencies in this protocol.

5.78 RFC 3124 The Congestion Manager

This document is IPv4 limited since it uses the IPv4 TOS header field.

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 908 Reliable Data Protocol (RDP)

This document is IPv4 limited as stated in the following section:

```
    4.1  IP Header Format

         When used in the internet environment, RDP segments are sent
     using  the  version 4 IP header as described in RFC791, "Internet
     Protocol."  The RDP protocol number is ??? (decimal).  The  time-
     to-live  field  should  be  set  to  a  reasonable  value for the
     network.

         All other fields should be set as specified in RFC-791.
```

A new protocol specification would be needed to support IPv6.

6.02 RFC 938 Internet Reliable Transaction Protocol functional and interface specification (IRTP)

This protocol specification states:

```
   4.1 State Variables

      Each IRTP is associated with a single internet address.  The
      synchronization mechanism of the IRTP depends on the requirement
      that each IRTP module knows the internet addresses of all modules
      with which it will communicate.  For each remote internet address,
      an IRTP module must maintain the following information (called the
      connection table):

      rem_addr     (32 bit remote internet address)
```

A new specification that is IPv6 aware would need to be created.

6.03 RFC 998 NETBLT: A bulk data transfer protocol (NETBLT)

This RFC states:

```
   The active end specifies a passive client through a client-specific
   "well-known" 16 bit port number on which the passive end listens.
   The active end identifies itself through a 32 bit Internet address
   and a unique 16 bit port number.
```

Clearly, this is IPv4 dependent, but could easily be modified to support IPv6 addressing.

6.04 RFC 1045 VMTP: Versatile Message Transaction Protocol (VMTP)

This protocol has many IPv4 dependencies in its implementation appendices. For operations over IPv6 a similar implementation procedure must be defined. The IPv4 specific information is show below.

IV.1. Domain 1

For initial use of VMTP, we define the domain with Domain identifier 1 as follows:

```
 +-----------+----------------+-----------------------+
 | TypeFlags | Discriminator  |    Internet Address    |
 +-----------+----------------+-----------------------+
    4 bits          28 bits                32 bits
```

The Internet address is the Internet address of the host on which this entity-id is originally allocated. The Discriminator is an arbitrary value that is unique relative to this Internet host address. In addition, the host must guarantee that this identifier does not get reused for a long period of time after it becomes invalid. ("Invalid" means that no VMTP module considers in bound to an entity.) One technique is to use the lower order bits of a 1 second clock. The clock need not represent real-time but must never be set back after a crash. In a simple implementation, using the low order bits of a clock as the time stamp, the generation of unique identifiers is overall limited to no more than 1 per second on average. The type flags were described in Section 3.1.

An entity may migrate between hosts. Thus, an implementation can heuristically use the embedded Internet address to locate an entity but should be prepared to maintain a cache of redirects for migrated entities, plus accept Notify operations indicating that migration has occurred.

Entity group identifiers in Domain 1 are structured in one of two forms, depending on whether they are well-known or dynamically allocated identifiers. A well-known entity identifier is structured as:

```
 +-----------+----------------+-----------------------+
 | TypeFlags |  Discriminator |Internet Host Group Addr|
 +-----------+----------------+-----------------------+
    4 bits          28 bits                32 bits
```

with the second high-order bit (GRP) set to 1. This form of entity identifier is mapped to the Internet host group address specified in the low-order 32 bits. The Discriminator distinguishes group identifiers using the same Internet host group. Well-known entity group identifiers should be allocated to correspond to the basic services provided by hosts that are members of the group, not specifically because that service is provided by VMTP. For example, the well-known entity group identifier for the domain name service should contain as its embedded Internet host group address the host group for Domain Name servers.

A dynamically allocated entity identifier is structured as:

```
 +-----------+----------------+-----------------------+
 | TypeFlags |  Discriminator |   Internet Host Addr   |
 +-----------+----------------+------------------------+
    4 bits          28 bits             32 bits
```

with the second high-order bit (GRP) set to 1. The Internet address in the low-order 32 bits is a Internet address assigned to the host that dynamically allocates this entity group identifier. A dynamically allocated entity group identifier is mapped to Internet host group address 232.X.X.X where X.X.X are the low-order 24 bits of the Discriminator subfield of the entity group identifier.

We use the following notation for Domain 1 entity identifiers <10> and propose it use as a standard convention.

```
        <flags>-<discriminator>-<Internet address>
```

where <flags> are [X]{BE,LE,RG,UG}[A]

```
    X = reserved
    BE = big-endian entity
    LE = little-endian entity
    RG = restricted group
    UG = unrestricted group
    A  = alias
```

and <discriminator> is a decimal integer and <Internet address> is in standard dotted decimal IP address notation.

V.1. Authentication Domain 1

A principal identifier is structured as follows.

```
 +---------------------------+------------------------+
 |     Internet Address      | Local User Identifier  |
 +---------------------------+------------------------+
             32 bits                    32 bits
```

VI. IP Implementation

VMTP is designed to be implemented on the DoD IP Internet Datagram Protocol (although it may also be implemented as a local network protocol directly in "raw" network packets.)

The well-known entity identifiers specified to date are:

```
VMTP_MANAGER_GROUP   RG-1-224.0.1.0
                Managers for VMTP operations.

VMTP_DEFAULT_BECLIENT  BE-1-224.0.1.0
                Client entity identifier to use when a (big-endian) host
                has not determined or been allocated any client entity
                identifiers.
```

VMTP_DEFAULT_LECLIENT LE-1-224.0.1.0

Client entity identifier to use when a (little-endian) host has not determined or been allocated any client entity identifiers.

Note that 224.0.1.0 is the host group address assigned to VMTP and to which all VMTP hosts belong.

6.05 RFC 1146 TCP alternate checksum options (TCP-ACO)

There are no IPv4 dependencies in this protocol.

6.06 RFC 1151 Version 2 of the Reliable Data Protocol (RDP) (RDP)

There are no IPv4 dependencies in this protocol.

6.07 RFC 1644 T/TCP -- TCP Extensions for Transactions Functional Specification (T/TCP)

There are no IPv4 dependencies in this protocol.

6.08 RFC 1693 An Extension to TCP : Partial Order Service (TCP-POS)

There are no IPv4 dependencies in this protocol.

6.09 RFC 1791 TCP And UDP Over IPX Networks With Fixed Path MTU

There are no IPv4 dependencies in this protocol.

6.10 RFC 2343 RTP Payload Format for Bundled MPEG (RTP-MPEG)

There are no IPv4 dependencies in this protocol.

6.11 RFC 2414 Increasing TCP's Initial Window (TCP-WIN)

There are no IPv4 dependencies in this protocol.

6.12 RFC 2582 The NewReno Modification to TCP's Fast Recovery Algorithm

There are no IPv4 dependencies in this protocol.

6.13 RFC 2762 Sampling of the Group Membership in RTP

This protocol is IPv4 limited. It is also reliant on the underlying assumptions of RTP which is also IPv4 specific.

6.14 RFC 2859 A Time Sliding Window Three Colour Marker (TSWTCM)

(TSWTCM)

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

6.15 RFC 2861 TCP Congestion Window Validation

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

6.16 RFC 2909 The Multicast Address-Set Claim (MASC) Protocol

(MASC)

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

7.0 Summary of Results

In the initial survey of RFCs 24 positives were identified out of a total of 99, broken down as follows:

Standards	4 of 5 or 80.00%
Draft Standards	0 of 0
Proposed Standards	15 of 78 or 19.23%
Experimental RFCs	5 of 16 or 31.25%

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

7.1.1 STD 7 Transmission Control Protocol (RFC 793)

Section 3.1 defines the technique for computing the TCP checksum that uses the 32 bit source and destination IPv4 addresses. This problem is addressed in RFC 2460 Section 8.1.

7.1.2 STD 19 Netbios over TCP/UDP (RFCs 1001 & 1002)

These two RFCs have many inherent IPv4 assumptions and a new set of protocols MUST be defined.

7.1.3 STD 35 ISO Transport over TCP (RFC 1006)

This problem has been fixed in RFC 2126, ISO Transport Service on top of TCP.

7.2 Draft Standards

7.3 Proposed Standards

7.3.01 TCP/IP Header Compression over Slow Serial Links (RFC 1144)

This problem has been resolved in RFC2508, Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. See also RFC 2507 & RFC 2509.

7.3.02 ONC RPC v2 (RFC 1833)

The problems can be resolved with a definition of the NC_INET6 protocol family.

7.3.03 RTP (RFC 1889)

A modification of the algorithm defined in A.7 to support both IPv4 and IPv6 addresses SHOULD be defined.

7.3.04 RTSP (RFC 2326)

Problem has been acknowledged by the RTSP developer group and will be addressed in the move from Proposed to Draft Standard. This problem is also addressed in RFC 2732, IPv6 Literal Addresses in URL's.

7.3.05 SDP (RFC 2327)

One problem is addressed in RFC 2732, IPv6 Literal Addresses in URL's. The other problem can be addressed with a minor textual clarification. This MUST be done if the document is to transition from Proposed to Draft.

7.3.06 SIP (RFC 2543)

One problem is addressed in RFC 2732, IPv6 Literal Addresses in URL's. The other problem is being addressed by the SIP WG and many IDs exist correcting the remaining problems.

7.3.07 IPPM Metrics (RFC 2678)

The IPPM WG is working to resolve these issues.

7.3.08 IPPM One Way Delay Metric for IPPM (RFC 2679)

The IPPM WG is working to resolve these issues. An ID is available (draft-ietf-ippm-owdp-03.txt).

7.3.09 IPPM One Way Packet Loss Metric for IPPM (RFC 2680)

The IPPM WG is working to resolve these issues.

7.3.10 Round Trip Delay Metric for IPPM (RFC 2681)

The IPPM WG is working to resolve these issues.

7.3.11 The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services(RFC 2848)

This protocol is dependent on SDP & SIP which has IPv4 dependencies. Once these limitations are fixed, then this protocol should support IPv6.

7.3.12 TCP Processing of the IPv4 Precedence Field (RFC 2873)

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

7.3.13 Integrated Services in the Presence of Compressible Flows (RFC 3006)

This document defines a protocol that discusses compressible flows, but only in an IPv4 context. When IPv6 compressible flows are defined, a similar technique should also be defined.

7.3.14 SDP For ATM Bearer Connections (RFC 3108)

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

7.3.15 The Congestion Manager (RFC 3124)

An update to this document can be simply define the use of the IPv6 Traffic Class field since it is defined to be exactly the same as the IPv4 TOS field.

7.4 Experimental RFCs

7.4.1 Reliable Data Protocol (RFC 908)

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

7.4.2 Internet Reliable Transaction Protocol functional and interface specification (RFC 938)

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

7.4.3 NETBLT: A bulk data transfer protocol (RFC 998)

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

7.4.4 VMTP: Versatile Message Transaction Protocol (RFC 1045)

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

7.4.5 OSPF over ATM and Proxy-PAR (RFC 2844)

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