Internet Draft E. Terrell
Category: Proposed Standard ETT-R&D Publications
Expires: March 3rd, 2008 September 2007

The IPtX Dynamic Host Configuration Protocol; DHCPvIPtX-MX

'draft-terrell-iptx-mx-dhcp-specification-02'

Status of this Memo

Intellectual Property Rights (IPR) Statement

Requirements Terminology

The keywords Must, Must Not, Required, Shall, Shall Not, Should, Should Not, Recommended, May, and Optional, when they appear in this document, are to be interpreted as described in [RFC-2119].

Conventions

Please note, the mathematical operators that cannot be represented in the 'txt' file format, which represent; the '^' Carrot sign for 'NESTED' Super-Script, and the 'v' sign is used for a 'NESTED' Sub-Script.

Abstract

This document defines the IPtX Specification for the 'Dynamic Host Configuration Protocol'; IPtX / IPtX-MX DHCP (DHCPvIPtX-MX), which provides Backwards Compatibility with the IPv4 Specification without compromise or change to current DHCP Server and Client Configuration and / or Operational requirements. And more importantly, because the IPtX / IPtX-MX Specification represents a 3 State Binary IP Addressing Specification, there are 2 IP Address Band Specifications; Mobile IP Address Pool and a Stationary IP Address Pool, with a 3 Dimensional Locator, which represents a 3 IP Address Coordinate System that uses an EMERGENCY Broadcast [e911] to establish a Synchronized LINK with 3 different [KNOWN] Router Locations and the MAC Address, to Triangulate the Location of any Node Connected to the Network -

[Given that - The 3 IP Address Coordinate System uses the CIDR Network Descriptors, '/0000:00', '+/0000:00', '-/0000:00' to differentiate the IP Address Broadcasting Node's Location to the 3 different [KNOWN] Router Locations, the differentiated IP Addresses however, must use the same IPtX-MX MAC Address to identify the Node's Hardware.]

Table of Contents

Abstract

Introduction

IANA Considerations

I. The DHCPv4 and DHCPv6 Header Design Specification

II. The IPv4 Compatibility DHCPvIPtX-MX 32 Bit Design Specification

III. The 'Preferred' DHCPvIPtX-MX 32 / 64 Bit Design Specification

IV. IPtX / IPtX-MX Mobile IP Addressing Specification

V. IPtX / IPtX-MX Subnet ID

Special IANA Considerations

Security Considerations

References

Work(s) in Progress

Normative References

Informative References

Introduction

The DHCPv4 Header, which was derived from the Bootp Protocol (RFC 951 Bootstrap Protocol), other than Commands, has not changed since 1985. And clearly, the purpose or functional use of DHCPv4 not is obsolete, because there are several viable reasons not to assign a Static IP Address to a Client. Especially when the Client is only a Guest of the Network. In other words, if any improvement in Performance or Use necessitates Change, then the DHCPvIPtX-MX Specification prescribes the logically viable reason(s) for making the changes.

IANA Consideration

I. The DHCPv4 and DHCPv6 Header Design Specification

```
                       IPv4 32 Bit Header

     0                   1                   2                   3
     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 2
1.   |Ver|   IHL   |      TOS       |         Total length          |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
2.   |        Identification        | Flags |    Fragment offset    |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
3.   |      TTL      |   Protocol    |          Header checksum      |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
4.   |                        SOURCE ADDRESS                          |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
5.   |                     DESTINATION ADDRESS                        |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
6.   |                            Options                             |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
7.   |                            Options                             |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
8.   |                            Options                             |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
9.   |                      Options and Padding                       |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
     |----------------------------------------------------------------|
```

DHCPv4 32 Bit Header

```
      0                   1                   2                   3
      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 2
 1.   |Opcode 8 Bits| H-ware type|Hardware address length| Hop count|
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 2.   |                      Transaction ID                           |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 3.   |       Number of seconds     |              Flags              |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 4.   |                      Client IP address                        |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 5.   |                       Your IP address                         |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 6.   |                      Server IP address                        |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 7.   |                     Gateway IP address                        |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 9.   |                  Client hardware address :::                  |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
11.   |                      Server host name :::                     |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
12.   |                      Boot filename :::                        |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
13.   |                          Options :::                          |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
      |---------------------------------------------------------------|
```

IPv6 32 Bit Header

```
         0                                       1                                       2                                       3
         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 2
    1.   |Ver |Traffic Class |              Flow Label                        |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
    2.   |      Payload Length        | Next Header |     HOP LIMIT         |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
    3.   |                       SOURCE ADDRESS                             |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
    4.   |                       SOURCE ADDRESS                             |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
    5.   |                       SOURCE ADDRESS                             |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
    6.   |                       SOURCE ADDRESS                             |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
    7.   |                       DESTINATION ADDRESS                        |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
    8.   |                       DESTINATION ADDRESS                        |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
    9.   |                       DESTINATION ADDRESS                        |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   10.   |                       DESTINATION ADDRESS                        |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   11.   |                        DATA                                      |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   12.   |                        DATA                                      |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   13.   |                        DATA                                      |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   14.   |                        DATA                                      |
         |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
         |--------------------------------------------------------------|
        Note: IPv6 Header Bit-Map Length = 14 x 4 Octets = 56 Octets
                  14 x 32 Bits = 56 Octets = IPv6 Header 448 Bits
```

DHCPv6 32 Bit Header

```
0                   1                   2                   3
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 2
|   Message     |           Data / Options* (variable length)   |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

Data / Options Information Fields*

```
|  msg-type     |                transaction-id                 |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                                                               |
|                           Options                             |
|                  (variable (16 octets) :::: )                 |
|                                                               |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|    msg-type     |   hop-count   |link-address (16 octets) ::::|
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                peer-address (16 octets) ::::                  |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|options (variable number and length) (16 octets) ::::          |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|             1               |    hardware type (16 bits)     |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                        time (32 bits)                         |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|        link-layer address (variable length) (16 octets) ::::|
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|


|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|             2               |       enterprise-number         |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|    enterprise-number (contd)   | Identifier (16 octets) :::: |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|

|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|             3               |    hardware type (16 bits)     |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|        link-layer address (variable length) (16 octets) ::::|
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|

|       OPTION_IA_PD            |         option-length          |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                         IAID (4 octets)                       |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                              T1                               |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                              T2                               |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                                                               |
|                          IA_PD-options                        |
|                                                               |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|


|      OPTION_IAPREFIX          |         option-length          |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                      preferred-lifetime                       |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                        valid-lifetime                         |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
| prefix-length |   IPv6 prefix (16 octets) ::::               |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                        IAprefix-options                       |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|    OPTION_NIS_SERVERS       |         option-len              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                 NIS server (IPv6 address) (16 octets) ::::    |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                 NIS server (IPv6 address) (16 octets) ::::    |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                  Additional Options ::::                      |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|   OPTION_NIS_DOMAIN_NAME     |         option-len              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                       nis-domain-name ::::                     |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|  OPTION_NISP_DOMAIN_NAME     |         option-len              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                      nisp-domain-name ::::                     |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|   OPTION_SNTP_SERVERS       |         option-len               |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                SNTP server (IPv6 address) (16 octets) ::::    |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                SNTP server (IPv6 address) (16 octets) ::::    |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                  Additional Options ::::                      |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|        option-code           |         option-len              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                     information-refresh-time                   |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|   OPTION_BCMCS_SERVER_A     |         option-len              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|     BCMCS Control server-1 (IPv6 address) (16 octets) ::::   |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|     BCMCS Control server-2 (IPv6 address) (16 octets) ::::   |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                    Additional Options ::::                    |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|     OPTION_SUBSCRIBER_ID     |         option-len              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                          subscriber-id                         |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|         OPTION_FQDN          |         option-len              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|   flags    |          domain-name    (16 octets) ::::         |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|

|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
| OPTION_NEW_POSIX_TIMEZONE   |         option-len              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                      TZ POSIX String (16 octets) ::::         |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

```
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|  OPTION_NEW_TZDB_TIMEZONE    |         option-len              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|                              TZ Name                          |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
```

II. The IPtX / IPtX-MX 32/64 Bit Header Design Specification

```
                IPtX / IPTX-MX 32 / 64 Bit Header

0                 1                   2                   3
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 2
/                    32 Bit Header Scale                      /

0                 2                   4                   6
2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4
/                    64 Bit Header Scale                      /
/                                                             /

 /          IPtX 32 / 64 Bit Header Information Fields        /
 /                                                            /
1 |  IPtX Version = 2E21/53 = 21/53 Bits |  Parity Notify Bit*  |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
2 |    Prefix    |DESTINATION ADDRESS Exponent = 2E 14 / 46 Bits|
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
3 |DESTINATION ADDRESS Exponential Decimal String = 2E22/54 Bits|
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
4 |  TTL / HOP LIMIT |   Option Section FLAGS =   16 / 32 Bits  |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
5 |  IPtX Version = 2E21/53 = 21/53 Bits |  Parity Notify Bit*  |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
6 |    Prefix    | SOURCE ADDRESS Exponent =  2E 14 / 46 Bits   |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
7 | SOURCE ADDRESS Exponential Decimal String = 2E 22 / 54 Bits |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
8 |          2E10.12 Bits = Option Section = 2E24.30 Bits       |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
9 |             2E10.12 Bits = DATA = 2E24.30 Bits              |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  |-------------------------------------------------------------|
    Note*: The 'Parity Notificication Bit' defines the 'PREFIX' as either a
           Character (1 Bit), or an Integer (0 Bit).
```

DHCPvIPtX-MX 32 Bit Header

```
        0                   1                   2                   3
        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 2
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   1.   | Parity Bit      |     Message CALL Flags 32 Bits              |
        | Flip to 64 Bits |                                             |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   2.   | Parity Bit      |     Option CALL Flags 32 Bits               |
        | Flip to 64 Bits |                                             |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   3.   | Authorization Transaction ID = 2E24 = 24 Bits|TTL/HOP LIMIT |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   4.   |    IPtX Version = 2E24 = 24 Bits      |  Parity Notify Bit*  |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   5.   |    Prefix     |    Server ADDRESS Exponent =  2E14 Bits       |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   6.   |   Server ADDRESS Exponential Decimal String = 2E22 Bits       |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   7.   |    IPtX Version = 2E24 = 24 Bits      |  Parity Notify Bit*  |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   8.   |    Prefix     |    Gateway ADDRESS Exponent =  2E14 Bits      |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   9.   |   Gateway ADDRESS Exponential Decimal String = 2E22 Bit       |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  10.   |           Message Section Exponent =  2E54 Bits               |

  11.   |                                                               |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  12.   |    Message Section Exponential Decimal String = 2E22 Bits     |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  13.   |            Option Section Exponent =  2E54 Bits               |
  14.   |                                                               |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  15.   |     Option Section Exponential Decimal String = 2E22 Bits     |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  16.   |                   Requesting  Client's                        |
        |  IPtX / IPtX-MX MAC Address and Hardware Info = 64 Bits       |
  17.   |  2EX.0000... = 2E4,194,304  -  2EQ.0000... = 2E4,194,304      |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  18.   |    Client's Network Account Info / DATA = 2E10.12 Bits        |
        |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
        |---------------------------------------------------------------|
```

Note: Client's MAC Address is used as SOURCE Address when Requesting Client is on the Backbone of the DHCP Server's Network.

In the structure of the Header noted above, for example, where each numbered Line defines a 32 Bit Field in a Transmission Sequence having the format of a Sentence, defines the 'Message and Option CALL Flag Fields' as a Set of Pointers interfacing with the 'Message and Options Section Fields', which defines a Set of Instructions ENCODED by the 'DCE Unit' that Performs a DHCP Task - as given below;

```
0 |01 |02 |03 |04 |05 |06 |07 |08 |09 |10 |11 |12 |...|32 |...|64 |
|                   Message Call Flags 64 Bits                      |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|

            01 - SOLICIT.              09 - DECLINE.

            02 - ADVERTISE.            10 - RECONFIGURE.

            03 - REQUEST.              11 - INFORMATION-REQUEST.

            04 - CONFIRM.              12 - RELAY-FORW.

            05 - RENEW.                13 - RELAY-REPL.

            06 - REBIND.               14 - Undefined.
                                        :       :
            07 - REPLY.                 :       :
                                        :       :
            08 - RELEASE.              64 -  Undefined.
```

```
0 |01 |02 |03 |04 |05 |06 |07 |08 |09 |10 |11 |12 |...|32 |...|64 |
|                      Option CALL Flags 64 Bits                    |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|

  01 - OPTION_CLIENTID.       ******  17 - OPTION_VENDOR_OPTS.

  02 - OPTION_SERVERID.       ******  18 - OPTION_INTERFACE_ID.

  03 - OPTION_IA_NA.          ******  19 - OPTION_RECONF_MSG.

  04 - OPTION_IA_TA.          ******  20 - OPTION_RECONF_ACCEPT.

  05 - OPTION_IAADDR.         ******  21 - SIP Servers Domain Name List.

  06 - OPTION_ORO.            ******  22 - SIP Servers IPtX Address List.

  07 - OPTION_PREFERENCE.     ******  23 - DNS Recursive Name Server.

  08 - OPTION_ELAPSED_TIME.   ******  24 - Domain Search List.

  09 - OPTION_RELAY_MSG.      ******  25 - OPTION_IA_PD

  10 - undefined.             ******  26 - OPTION_IAPREFIX

  11 - OPTION_AUTH.           ******  27 - OPTION_NIS_SERVERS

  12 - OPTION_UNICAST.        ******  28 - OPTION_NISP_SERVERS

  13 - OPTION_STATUS_CODE.    ******  29 - OPTION_NIS_DOMAIN_NAME

  14 - OPTION_RAPID_COMMIT.   ******  30 - OPTION_NISP_DOMAIN_NAME

  15 - OPTION_USER_CLASS.     ******  31 - SNTP server list.

  16 - OPTION_VENDOR_CLASS.   ******  32 - Information Refresh Time.
```

```
0 |01 |02 |03 |04 |05 |06 |07 |08 |09 |10 |11 |12 |...|32 |...|64 |
|                       Option CALL Flags 64 Bits                    |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|

33 - BCMCS Controller Domain Name list.       *** 49 - Undefined.

34 - BCMCS Controller IPtX address list.      *** 50 - Undefined.

35 - undefined.                               *** 51 - Undefined.

36 - OPTION_GEOCONF_CIVIC.                    *** 52 - Undefined.

37 - OPTION_REMOTE_ID.                        *** 53 - Undefined.

38 - Relay Agent Subscriber-ID.               *** 54 - Undefined.

39 - FQDN, Fully Qualified Domain Name.       *** 55 - Undefined.

40 - OPTION_PANA_AGENT.                       *** 56 - Undefined.

41 - OPTION_NEW_POSIX_TIMEZONE.               *** 57 - Undefined.

42 - OPTION_NEW_TZDB_TIMEZONE.                *** 58 - Undefined.

43 - OPTION_ERO.                              *** 59 - Undefined.

44 - OPTION_LQ_QUERY.                         *** 60 - Undefined.

45 - OPTION_CLIENT_DATA.                      *** 61 - Undefined.

46 - OPTION_CLT_TIME.                         *** 62 - Undefined.

47 - OPTION_LQ_RELAY_DATA.                    *** 63 - Undefined.

48 - OPTION_LQ_CLIENT_LINK.                   *** 64 - Undefined.
```

RECALL; -- Using the 'Data Compression' Ratio; '2EX : 1', or 2^{x} --

Example of Encoding the Bit-Map of the Equation for the 'Message and Option Section Fields'

Example of Text to encode...

'I went to the store today.'

I = 01001001 = 73 = 2EX ~ 2E8

went = 01110111011001010110111001110100 = 2,003,136,116 = 2EX ~ 2E32

to = 0111010001101111 = 29,807 = 2EX ~ 2E16

the = 011101000110100001100101 = 7,628,901 = 2EX ~ 2E24

store = 0111001101110100011011110111001001100101 = 495,874,699,877 = 2EX ~ 2E40

today = 0111010001101111011001000110000101111001 = 500,085,055,865 = 2EX ~ 2E40

' . ' = 00101110 = 46 = 2EX ~ 2E8

The Equivalent Binary Numerical Conversion to be Transmitted;

```
'I went to the store today.'            'Iwenttothestoretoday.'

010010010010000001110111011001         010010010111011101100101011011
```

Now... 'Taking it Away' yields;

```
And... 'Putting it Together' yields;

        'I  +  went  +  to  +  the  +  store  +  today  +  .'

   I = 01001001 = 73 +

   Blank Space = 00100000 = 32 +

   went = 01110111011001010110111001110100 = 2,003,136,116 +

   Blank Space = 00100000 = 32 +

   to = 0111010001101111 0 = 29,807 +

   Blank Space = 00100000 = 32 +

   the = 011101000110100001100101 0 = 7,628,901 +

   Blank Space = 00100000 = 32 +

   store = 0111001101110100011011110111001001100101 0 = 495,874,699,877 +

   Blank Space = 00100000 = 32 +

   today = 0111010001101111011001000110000101111001 0 = 500,085,055,865 +

   No Blank Space = Zero

   ' . ' = 00101110 = 46
```

```
Assembling (Joining) the Data Stream yields;

   I(73) + Blank(32) + went(2,003,136,116) + Blank(32) + to(29,807) +
   Blank(32) + the(7,628,901) + Blank(32) + store(495,874,699,877) +
   Blank(32) + today(500,085,055,865) + Period(46)

 =  73 + 32 + 2003136116 + 32 + 29807 + 32 + 7628901 + 32 +
    495874699877 + 32 + 500085055865 + 46 = 60 Digit Number

        =  733,220,031,361,163,229,807,327,628,901,324,958,746,
           998,773,250,008,505,586,546 = 2E198.868003799...

      =   2        E        198  .  868003799 ...
      =  11 01000101   11000110  .  11001110111100101011111010111
      =  2E198.868003799... = 48 Bit-Mapped Displacement
                   [ ' . ' = 8 Bits = 00101110 = 46 ]
   48 - 56 Bits  vs  208 Bits    -    6 - 7 Octets  vs  26 Octets
```

- Or -

= 60 Digit Number = 'I went to the store today.' = 2E198.868003799...

And this is equivalent to 26 Bytes, or approximately 208 Bits.

- Or -

2E198.868003799... ~ 2E208 = an approximate Bit-Mapped Displacement of 20 Bits (4 + 8 + 8). Or 20 Bits vs. 208 Bits; represents the difference between Bit-Mapping the 'Data Stream', as compared to Bit-Mapping the Equation of the 'Data Stream'.

Note: The Bit Mapped example used above follows from the Current Binary Translation, which includes the Askew Error!

And more importantly, the Compression Ratio becomes even greater, by some Exponential factor, as the amount of Data, which is to be Compressed increases. - e.g. 100Mbyte (800 MBit ~ 100,000,000 Octets) Document is compressed to '2E800,000,000', or (4 + 8 + 30) 42 Bits (~ 6 Octets) [Approximating a '20,000,000 to 1' Bit-Mapped Compression Ratio].

Furthermore, it should be readily concluded, since each of the numbered Line in the DHCPvIPtX-MX Header defines a 32 Bit Field in a Transmission Sequence having the format of a Sentence, also defines the 'Message and Option CALL Flag Fields' as a Set of Pointers interfacing with the 'Message and Option Section Fields'. Where the 'Message and Options Section Fields' contains the Set of Instructions ENCODED by the 'DCE Unit', can Perform any assigned DHCP Task.

III. The IPtX / IPtX-MX 64 Bit Header Design Specification

DHCPvIPtX-MX 64 Bit Header

```
     0                 2                   4                   6
     2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4
1.   |                   Message CALL Flags 64 Bits                  |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
2.   |                   Option CALL Flags 64 Bits                   |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
3.   |Authorization Transaction ID = 2E24 = 24 Bits |TTL/HOP LIMIT |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
4.   |     IPtX Version = 2E24 = 24 Bits      | Parity Notify Bit*  |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
5.   |    Prefix     |    Server ADDRESS Exponent =  2E46 Bits       |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
6.   |     Server ADDRESS Exponential Decimal String = 2E54 Bits     |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
7.   |     IPtX Version = 2E24 = 24 Bits      | Parity Notify Bit*  |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
8.   |   Prefix      |     Server ADDRESS Exponent =  2E46 Bits      |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
9.   |     Server ADDRESS Exponential Decimal String = 2E54 Bit      |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|

10.   |            Message Section Exponent =  2E118 Bits             |
11.   |                                                               |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
12.   |  Message Section Exponential Decimal String = 2E54 Bits       |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
13.   |             Option Section Exponent =  2E118 Bits             |
14.   |                                                               |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
15.   |  Option Section Exponential Decimal String = 2E54 Bits        |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
      |                   Requesting  Client's                        |
16.   |  IPtX / IPtX-MX MAC Address and Hardware Info = 64 Bits       |
      |  2EX.0000... = 2E4,194,304  -  2EQ.0000... = 2E4,194,304      |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
17.   |  Client's Network Account Info / DATA = 2E10.12 Bits          |
      |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
      |---------------------------------------------------------------|
```

Note: Client's MAC Address is used as SOURCE Address when Requesting Client is on the Backbone of the DHCP Server's Network.

IV. IPtX / IPtX-MX Mobile IP Addressing Specification

There is a far greater growth potential, which expands the IPtX IP Addressing Protocol Family Specification, well beyond the results from the use of a Single IP Address Band Specification. That is, when adding the use of the 'Bar E' (Ě) notation to the 'DCE Unit' {2ĚQ} (given that the Members of the 'Real Number Set' represents every possible Numeral, denoting an Infinite Set), the IP Address Pool Total defined by the IPtX Specification increases to an amount equal to 'Bit-Mapping' every Element, or Member defined by the 'Set of Real' Numbers. In other words, the IPtX /IPtX-MX Specification defines a Logical 3 State Binary (2 Band) IP Addressing Specification, defining a Stationary and a Mobile IP Addressing Bands in a 3 Dimensional Space. – As given by;

- IPtX / IPtX-MX Specification -

Stationary Band = 0000:2EX.0000...

Mobile Band = 0000:2ĒX .0000...

And more importantly, with each of these Address Band Specification there is a corresponding 'MAC Address' Specification – as given by;

- IPtX / IPtX-MX Specification -

Stationary Band 'MAC Address' Specification = 2EX.0000...

Mobile Band 'MAC Address' Specification = 2ĒX .0000...

```
            IPtX / IPTX-MX 32 / 64 Bit Mobile Header

  0                   1                   2                   3
  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 2
  /                          32 Bit Header Scale                 /

  0                   2                   4                   6
  2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4
  /                          64 Bit Header Scale                 /
  /                                                              /
   /         IPtX 32 / 64 Bit Header Information Fields           /
   /                                                              /
1 |  IPtX Version = 2E21/53 = 21/53 Bits |  Parity Notify Bit*   |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
2 |    Prefix    |DESTINATION ADDRESS Exponent = 2E 14 / 46 Bits|
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
3 |DESTINATION ADDRESS Exponential Decimal String = 2E22/54 Bits|
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
4 |  TTL / HOP LIMIT |   Option Section FLAGS =   16 / 32 Bits   |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
5 |  IPtX Version = 2E21/53 = 21/53 Bits |  Parity Notify Bit*   |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
6 |    Prefix    | SOURCE ADDRESS Exponent =  2E 14 / 46 Bits    |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
7 | SOURCE ADDRESS Exponential Decimal String = 2E 22 / 54 Bits |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
8 |          2E10.12 Bits = Option Section = 2E24.30 Bits        |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
9 |             2E10.12 Bits = DATA = 2E24.30 Bits               |
  |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  |--------------------------------------------------------------|
    Note*: The 'Parity Notificication Bit' defines the 'PREFIX' as either a
           Character (1 Bit), or an Integer (0 Bit).
```

V. IPtX / IPtX-MX Subnet ID

It is extremely important to note, the general procedures for Subnetting, or allocating IP Address to a Sub-Division of the Network remain unchanged. That is, while the Subnet Mask has changed, Subnetting or allocating IP Address to a smaller Sub-Division of the Network remains unchanged, because it provides an easy method to account for every Node in the Structure of Network Hierarchal Scheme. However, because of the number of available IP Address in the IPtX / IPtX-MX IP Address Pool, Supernetting, as it were, is no longer a viable procedure or useful concept, especially since the IPtX / IPtX-MX Specification Sequentially counts every available IP Address.

Note: The Subnet Mask, now defined as the Subnet ID for the Stationary and Mobile IP Address Bands is given by;

Stationary Band Subnet ID = 0000:DCE Unit.0000...

Mobile Band Subnet ID = 0000:DC$\overline{\text{E}}$ Unit.0000...

Special IANA Considerations

Clearly, further exploitation of the 'DCE Unit'; since it has been shown that the Binary Exponential Base 2 Algorithm, '2EX', sequentially count using successive additions of "1's". The 'Preferred' Design of the 'Message CALL Flags Field' and the 'Option CALL Flags Field' in the DHCPvIPtX-MX 32 / 64 Bit Header Specification, is given by;

```
Note: The using the 'DCE Unit' to redefine the 32 and 64 Bit Scales to
      represent a 'One to One' Correspondence with the Set of Integers,
      Bit-Maps each Flag as the Incremental Progression from 1 thru 32,
      or 64. And while this defines the Flags Progression in each Field,
      the Sequence Order of the Integer(s) representing the Bit Mapped
      Flag(s) is Function Governed. Hence, from pages 17 thru 20, the
      procedure for converting the first '3' 'Bit-Mapped Flag(s)',
      which represents the Sequence 1, 2, and 3, is given by;
```

```
Given that - the example of the Text to encode...

      '1, 2, 3'                                   '123'

0011000100101100001000000011OO        001100010011001000110011
10001011000010000000110011

      56 Bits                                  24 Bits

Recalling that everything is counted, which includes the 'COMMA(s)'
and the 'BLANK SPACE(s)' Separating every Numeral the Sequence
contains -

    Hence, the Numerical Sequence represents a '56 Bit Sentence';

                BLANK SPACE = 00100000 = 8 Bits

                      COMMA = 00101100 = 8 Bits
```

```
Note: The Bit Mapped example used above follows from the Current
      Binary Translation, which includes the Askew Error!
```

```
      'Taking it Away' yields:          'Putting it Together' yields;
             " 1, 2, 3 "             | " 1 + , + ' ' + 2 , + ' ' + 3 "
|------------------------------------+---------------------------------|
 '1'  =  00110001 = 49 = 2EX ~ 2E8   |       '1'  =   00110001  =  49
                                     |                                +
 'Comma' = 00101100 = 44 = 2EX ~ 2E8 |   'Comma'  =   00101100  =  44
                                     |                                +
 'Space' = 00100000 = 32 = 2EX ~ 2E8 |   'Space'  =   00100000  =  32
                                     |                                +
 '2' = 00110010 = 50 = 2EX ~ 2E8     |       '2'  =   00110010  =  50
                                     |                                +
 'Comma' = 00101100 = 44 = 2EX ~ 2E8 |   'Comma'  =   00101100  =  44
                                     |                                +
 'Space' = 00100000 = 32 = 2EX ~ 2E8 |   'Space'  =   00100000  =  32
                                     |                                +
 '3' = 00110011 = 51 = 2EX ~ 2E8     |       '3'  =   00110011  =  51
|------------------------------------+---------------------------------|
```

Note:

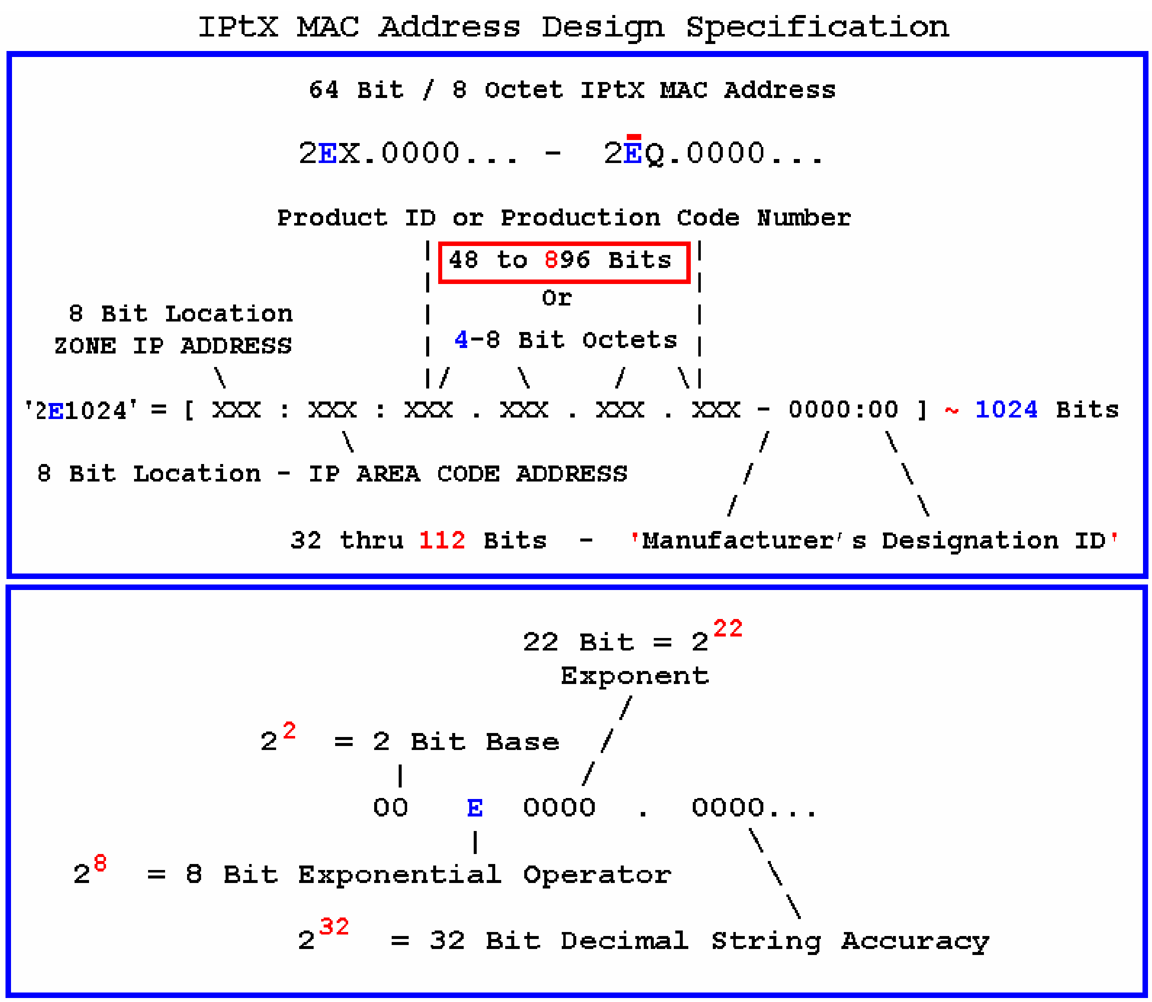
Encoding and Decoding the 'Binary Assembler', '2EX', representing the 'Assembled Data Stream' defining the 'DCE Unit', uses the "Punctuation" (Semi Colon, Commas, Spaces, Carriage Return... etc - in particular, the 'Blank Space') as 8 Bit HOOKS, to Decode or Encoded the 'Set of Instructions' - Given that; an individual Character or Numeral equals 8 Bits (2E8, or 2^8), where the HOOKS are used to define the Boundary Length of a Word or Numerical Sequence that is equal to 1 or more 8 Bit Octets, and contained in the Sentence defining any combination of Characters, Words, Numerals, and / or Punctuations. And clearly, once the 'Blank Spaces' has been identified, Decoding in a Right to Left or Left to Right 8 Bit Pattern would easily identify the remaining 8 Bit (Individual) Characters, Digits, and Punctuation(s) the Sentence contains.

```
And Assembling or Joining the Number Stream, '1, 2, 3', yields;

 1(49) + Comma(44) + Space(32) + 2(50) + Comma(44) + Space(32) + 3(51)

   =  49 + 44 + 32 + 50 + 44 + 32 + 51

   =  49,443,250,443,251

   =  14 Digit Number

   ~  2E45.4598888888...

  =   2        E          45   .  4598888888 ...
  =  11  01000101   00101101   .  10001001000011101011100110111000

  =  2E45.4598888888...  =  51 Bit-Mapped Displacement
```

Furthermore, after realizing that Binary Exponential Base 2 Operations can be 'Nested', employing the same Backbone method used to exhaust all of the available IPtX-MX IP Addresses. Using the Assemblage of the 'Backbone ISP ID', the 'Backbone Network Account ID', the 'Backbone Users Account ID', and the 'Users IP Address' into the 'DCE Unit' of the IP Address (See 'Work in Progress' [12]). The 'Binary Assembler', '2EX', it should also be Mathematically concluded, can also be used to represent a Group or several 'Nested Binary Exponential Base 2 Operations' – as given by;

```
2EX ≥ 2EA + 2EB + 2EC + 2ED + 2EE + 2EF + ... + 2EZ

2EX ≥ 2EA x 2EB x 2EC x 2ED x 2EE x 2EF x ... x 2EZ
```

In other words, for example, the IPtX-MX MAC Address's 'Data Field Section', which represents the 'Product ID or Production Code Number' - The Nesting of several 'Binary Exponential Base 2 Operations' makes it possible to Write a Detailed Description of the Entire Product; or create a 'Product Form', which includes any relevant Date(s), Detailed Description, Use, and a Product ID. And this, together with the 'ZONE and IP AREA CODE Addresses', and the 'Manufacturer's Designation ID' completes the Address Code identifying the 'Hardware' and / or 'Product' - because the IPtX-MX MAC Addressing Specification could also replace the BAR CODE method, which is currently being used for 'Product Identification'.



- Using several (Nested) "Binary Assembler's", '2EX', the 'Blank Space', and the 'Carriage Return' defines the beginnings of a Language Syntax utilizing Fundamental Principles, which are key characteristics of 'Binary Assembler' Programming -

```
    - Given the 'Encoded Lines' to be 'Assembled' yields;

         '2E198.868003799...'   =   'I went to the store today.'

Binary Assembler   =   2        E        198  .   868003799... = 2EQ
Binary Conversion  =  11 01000101  11000110  .   10011101111001010111111010111
Encoded Line(s)    =   4        69        198 46   868003799...
```

```
          46,919,846,868,003,799.. = 469198 '46' 868003799...

     Line 1 - 2E198.868003799... = 46,919,846,868,003,799...

          Blank Space = 32

     Line 2 - 2E198.868003799... = 46,919,846,868,003,799...

          Blank Space = 32

     Line 3 - 2E198.868003799... = 46,919,846,868,003,799...

          Blank Space = 32

     Line 4 - 2E198.868003799... = 46,919,846,868,003,799...
```

[Where the 46 = Period, or the Decimal Point in the Equation, defines the Bit-Mapped Pattern; '2 - 8 - 8 - 8(46) - And All Remaining Bits', as the 'Binary Bit-Map Sequence Key'. However, while a Second Level of Compression is impressive, any further Compression could cause a reduction in the Accuracy of the Exponential Decimal String of the Binary Assembler, 2EX, defining each of the Coded Lines.]

- The Lines to Encode...

Line 1 + 'Space' + Line 2 + 'Space' + Line 3 + 'Space' + Line 4

46,919,846,868,003,799 + '32' + 46,919,846,868,003,799
\+ '32' + 46,919,846,868,003,799... + '32' + 46,919,846,868,003,799

= 46,919,846,868,003,799,324,691,984,686,800,379,932,469,198,468,
680,037,993,246,919,846,868,003,799 = 74 Digit Number = 2EQ

≈ 2E244.73... = 2EQ

= 2 E 244 . 73 ...
= 11 01000101 11110100 . 1001001 ... = 25 Bits
[4(208) + 32(46) + 24(32) = 888 Bit Length - Reduced to 25 Bits]

- Given That -

2EX ≥ 2EA + 2EB + 2EC + 2ED + 2EE + 2EF + ... + 2EQ

2EX ≥ 2EA x 2EB x 2EC x 2ED x 2EE x 2EF x ... x 2EQ

Note: The Bit Mapped example used above follows from the Current Binary Translation, which includes the Askew Error!

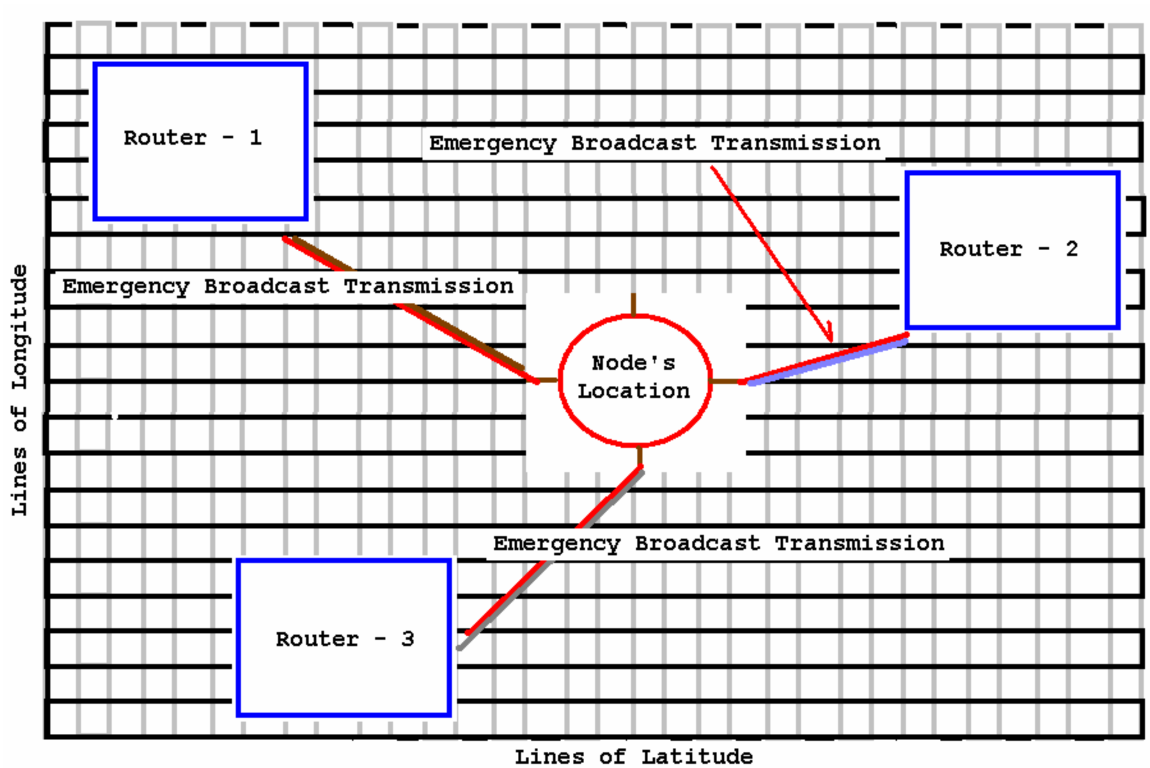
```
                         DHCPvIPtX-MX  32 Bit Header

     0                   1                   2                   3
     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 2
    |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
1.  |                Message CALL Flags = 2E10.12 Bits              |
    |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
2.  |                Option CALL Flags = 2E10.12 Bits               |
    |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|



 3.  | Authorization Transaction ID = 2E24 = 24 Bits|TTL/HOP LIMIT |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 4.  |    IPtX Version = 2E24 = 24 Bits      |  Parity Notify Bit* |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 5.  |    Prefix     |     Server ADDRESS Exponent =  2E14 Bits    |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 6.  |   Server ADDRESS Exponential Decimal String = 2E22 Bits     |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 7.  |    IPtX Version = 2E24 = 24 Bits      |  Parity Notify Bit* |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 8.  |    Prefix     |     Gateway ADDRESS Exponent =  2E14 Bits   |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 9.  |   Gateway ADDRESS Exponential Decimal String = 2E22 Bit     |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
10.  |           Message Section Exponent =  2E54 Bits             |
11.  |                                                             |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
12.  |    Message Section Exponential Decimal String = 2E22 Bits   |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
13.  |            Option Section Exponent =  2E54 Bits             |
14.  |                                                             |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
15.  |     Option Section Exponential Decimal String = 2E22 Bits   |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
16.  |                    Requesting  Client's                     |
     |  IPtX / IPtX-MX MAC Address and Hardware Info = 64 Bits     |
17.  |  2EX.0000... = 2E4,194,304  -  2EQ.0000... = 2E4,194,304    |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
18.  |    Client's Network Account Info / DATA = 2E10.12 Bits      |
     |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
     |-------------------------------------------------------------|
         Note: Client's MAC Address is used as SOURCE Address when Requesting
               Client is on the Backbone of the DHCP Server's Network.
```

DHCPvIPtX-MX 64 Bit Header

```
       0                 2                   4                   6
       2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4
   1.  |               Message CALL Flags = 2E24.30 Bits             |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   2.  |               Option CALL Flags = 2E24.30 Bits              |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|


   3.  |Authorization Transaction ID = 2E24 = 24 Bits |TTL/HOP LIMIT |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   4.  |     IPtX Version = 2E24 = 24 Bits      | Parity Notify Bit* |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   5.  |    Prefix     |    Server ADDRESS Exponent =  2E46 Bits      |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   6.  |     Server ADDRESS Exponential Decimal String = 2E54 Bits   |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   7.  |     IPtX Version = 2E24 = 24 Bits      | Parity Notify Bit* |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   8.  |   Prefix      |     Server ADDRESS Exponent =  2E46 Bits     |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
   9.  |     Server ADDRESS Exponential Decimal String = 2E54 Bit    |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  10.  |           Message Section Exponent =  2E118 Bits            |
  11.  |                                                             |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  12.  |  Message Section Exponential Decimal String = 2E54 Bits     |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  13.  |            Option Section Exponent =  2E118 Bits            |
  14.  |                                                             |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  15.  |  Option Section Exponential Decimal String = 2E54 Bits      |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
       |                    Requesting  Client's                     |
  16.  |   IPtX / IPtX-MX MAC Address and Hardware Info = 64 Bits    |
       |   2EX.0000... = 2E4,194,304  -  2EQ.0000... = 2E4,194,304   |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
  17.  |  Client's Network Account Info / DATA = 2E10.12 Bits        |
       |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
       |-------------------------------------------------------------|
```

Note: Client's MAC Address is used as SOURCE Address when Requesting Client is on the Backbone of the DHCP Server's Network.

Example of a 3D Grid Locator Scheme using a 3 IP Address Coordinate System in conjunction with a 'Longitude and Latitude' Rectangular Grid Overlay -



Noting more specifically that;

IANA EMERGENCY BROADCAST IP ADDRESS PROTOCOL

```
001-256| 001-256:|    All:  | - - - -  e .911 |   All  |IANA/Emer| 7/07
```

Representing on the Backbone, the IPtX-MX IP Address Mask given by;

0000:2E911.0000...

```
 Prefix:  |Zone IP:  |IP Area Code:  |Emergency Response Address  |/XA

   0000:       2           E              911      .   0000...
11111111      11      11001010       1110001111    .   1111...
```

Note: The Exponential Decimal String, '1111...', is used to Derive / Assign the additional IP Addresses on / for the 'e911 Emergency Response Network'.

Note - IANA \ IEEE Special Consideration -

- The Method used for Electronic Signal Propagation to Distinguish between Zero = '0', and Binary 1 = '00' -

1) A Single Position Binary 2 State Switch - Single Position 2 State Switch |X,Y| - Yielding; |*| = 0 or |1| = 1

   a) Single Position / 2 State Switch - True or False

   b) 2 States defines the Choice of either a '0' or a '1'

   c) '0' defines a No Electronic Signal Pattern '*' = 0

   d) '1' defines a True electronic Signal Pattern '1_1 ' = 1

2) A Double Position Binary 3 State Switch - Double Position 3 State Switch |*,X,Y| - Yielding; |*,0| = 0, |0,0| = 1, |0,1| = 2, |1,0| = 3, and |1,1| = 4 -

   a) Double Position / 3 State Switch - True, False, or No Response

   b) 3 States defines the Choice or Combination of either '*0', '00', '01'

   c) '0' defines No Electronic Signal Pattern '*' = 0

   d) '00' defines a True electronic Signal Pattern '00_1 ' = 1

   e) '01' defines a True electronic Signal Pattern '01_1 ' = 2

Security Considerations

There are No Security Considerations presented in this document.

Work(s) in Progress;

These drafts represent the twelve chapters of the Networking Bible, designing a Network IP Addressing Specification that maintains a 100 Percent backward compatibility with the IPv4 Specification. In other words, this is a design specification developed from the Theory of the Expansion of the IPv4 IP Addressing Specification, which allowed the representation of the Network for the entire World on paper, and the possibility of an Infinite IP Address Pool. Nevertheless, the Internet-Drafts listed below, "Cited as Work(s) in Progress', explain the design Specification for the development of the IPtX (IP Telecommunications Specification) Protocol Addressing System and the correction of the Mathematical Error in the Binary System.

Computer Science / Internet Technology:

1. **http://www.ietf.org/internet-drafts/draft-terrell-logic-analy-bin-ip-spec-ipv7-ipv8-10.txt**
(Foundational Theory for the New IPtX family IP Addressing Specification, and the Binary Enumeration error discovery after the correction.) - "Work(s) in Progress'

2. **http://www.ietf.org/internet-drafts/draft-terrell-simple-proof-support-logic-analy-bin-02.txt**
(The 2nd proof for the existence of another Binary System, resulting from the Error Correction.) - "Work(s) in Progress'

3. **http://www.ietf.org/internet-drafts/draft-terrell-visual-change-redefining-role-ipv6-01.pdf**
(Argument against the Machine dependant IPv6 deployment.) - "Work(s) in Progress'

4. **http://www.ietf.org/internet-drafts/draft-terrell-schem-desgn-ipt1-ipt2-cmput-tel-numb-02.pdf**
(The foundation of the New IPtX Addressing Spec compared to the Telephone Numbering System.) - "Work(s) in Progress'

5. **http://www.ietf.org/internet-drafts/draft-terrell-internet-protocol-t1-t2-ad-sp-06.pdf**
(The IPtX Addressing Specification Address Space / IP Address Allocation Table; establishes the visual perspective that actually represents Networking Schematic Networking the entire World on Paper.) - "Work(s) in Progress'

6. **http://www.ietf.org/internet-drafts/draft-terrell-iptx-spec-def-cidr-ach-net-descrip-01.pdf**
(Re-Defines CIDR) {Classes Inter-Domain Routing Architecture} and introduces the Network Descriptor for the IPtX Addressing Standard.) - "Work(s) in Progress'

7. **http://www.ietf.org/internet-drafts/draft-terrell-math-quant-new-para-redefi-bin-math-04.pdf**
(The 3rd Proof for the New Binary System, correcting the error in Binary Enumeration.) - "Work(s) in Progress'

8. **http://www.ietf.org/internet-drafts/draft-terrell-gwebs-vs-ieps-00.pdf**
(Defining the GWEBS – The Global Wide Emergency Broadcast System) - "Work(s) in Progress'

9. **http://www.ietf.org/internet-drafts/draft-terrell-iptx-dhcp-req-iptx-ip-add-spec-00.pdf**
(The development of the DHCP {Dynamic Host Configuration Protocol} for the IPTX IPSpec) - "Work(s) in Progress'

11. http://www.ietf.org/internet-drafts/draft-terrell-math-quant -ternary-logic-of-binary-sys-10.pdf
(Derived the Binary System from the proof of "Fermat's Last Theorem", and Developed the Ternary Logic for the Binary System) - "Work(s) in Progress"

12. http://www.ietf.org/internet-drafts/draft-terrell-cidr-net-descrpt-expands-iptx-add-spc-20.pdf
- "Work(s) in Progress"
(An application of Quantum Scale Theory, the 2^X : 1 Compression Ratio, the Expansion derived from the 'CIDR Network Descriptor, and the Mathematics of Quantification provided the foundation for the development of the "Intelligent Quantum Tunneling Worm Protocol"; A Routable Mathematical Exponential Expression, Backend IP Addressing Protocol that provides an (nearly) Unlimited IP Address Space using the Compression Ratio 2^X : 1.)

13. http://www.ietf.org/internet-drafts/draft-terrell-iptx-mx-dns-specification-04.pdf
(The development of the IPtX / IPtX-MX DNS {Domain Name Service} for IPTX IP Addressing Spec) 'Work(s) in Progress'

14. http://www.ietf.org/internet-drafts/draft-terrell-iptx-mx-dhcp-specification-02.pdf
(The development of the IPtX / IPtX-MX DHCP {Dynamic Host Configuration Protocol } for IPTX IP Addressing Spec) 'Work(s) in Progress'

Note: These Drafts has Expired at www.ietf.org Web Site. However, you can still find copies posted at Web Sites all over the World. {Suggestion; Perform Internet search using "Yahoo" or "Google", Key word: "ETT-R&D Publications"}.

Normative References:

Pure Mathematics:

1. **The Proof of Fermat's Last Theorem; The Revolution in Mathematical Thought {Nov 1979}**
   Outlines the significance of the need for a thorough understanding of the Concept of Quantification and the Concept of the Common Coefficient. These principles, as well many others, were found to maintain an unyielding importance in the Logical Analysis of Exponential Equations in Number Theory.

2. **The Rudiments of Finite Algebra; The Results of Quantification {July 1983}**
   Demonstrates the use of the Exponent in Logical Analysis, not only of the Pure Arithmetic Functions of Number Theory, but Pure Logic as well. Where the Exponent was utilized in the Logical Expansion of the underlining concepts of Set Theory and the Field Postulates. The results yield another Distributive Property that is Conditional, which supports the existence of a Finite Field (i.e. Distributive Law for Exponential Functions) and emphasized the possibility of an Alternate View of the Entire Mathematical field.

3. **The Rudiments of Finite Geometry; The Results of Quantification {June 2003}**
   Building upon the preceding works from which the Mathematics of Quantification was derived. Where by it was logically concluded that there existed only 2 mathematical operations; Addition and Subtraction. In other words, the objectives this treatise maintained, which was derived from the foundation of the Mathematics of Quantification; involves not only the clarification of the misconceptions concerning Euclid's Fifth Postulate, and the logical foundation of his work, or the existence of 'Infinity in a Closed Bound Finite Space'. But, the logical derivation of the Foundational Principles that are consistence with the foundation presented by Euclid, which would establish the logical format for the Unification of all the Geometries presently existing.

4. **The Rudiments of Finite Trigonometry; The Results of Quantification {July 2004}**
   The development of the concepts for Finite Trigonometry from the combined foundations derived from numbers 3 and 5, and the Mathematics of Quantification.

5. **The Mathematics of Quantification and the Metamorphosis of π : τ { October 2004}**
   The logical derivation of the exact relationship between the Circumference and the Diameter of the Circle, which defines the measurement of the exact length of the Circle's Circumference,τ when the Radius is equal to '1'.

6. **Squaring the Circle? First! What is the Circle's Area? {January 2005}**
   The Rhind Papyrus Tale, and the 10,000 year old quest involving "Squaring the Circle"; Derivation of the equation resolving the Area of the Circle. An illusion perplexing the Sight and Mind of the greatest mathematicians for about 10,000 years, which maintains an elementary algebraic solution: $(\pi r \div 2)^2$ = Area of Circle.

Physics:

7. **The Mathematics of Quantification & The Rudiments of Finite Physics**
   The Analysis of Newton's Laws of Motion…the Graviton' {December 2004}
   Through the use of Finite Algebra, Geometry, Trigonometry, and # 5, investigation of the Laws of Classical Physics were found to be erroneous. This allowed the presentation of the initial work, which correct the flaws in Classical Physics, and establishes the foundation upon which there exist the possibility of a Grand Unified Field Theory for the Natural Sciences.

Informative References

1. G Boole (Dover publication, 1958) "An Investigation of The Laws of Thought" On which is founded The Mathematical Theories of Logic and Probabilities; and the Logic of Computer Mathematics.

2. R Carnap (University of Chicago Press, 1947 / 1958) "Meaning and Necessity" A study in Semantics and Modal Logic.

3. R Carnap (Dover Publications, 1958) " Introduction to Symbolic Logic and its Applications"

4. Regis Desmeules (Cisco Press, April 24, 2003) " Cisco Self-Study: Implementing Cisco IPv6 Networks "

5. Gary C. Kessler (Auerbach Press, August 1997) " Handbook on Local Area Networks "

6. R. Hinden (Nokia) and S. Deering (Cisco Systems) RFC 2373 - " IP Version 6 Addressing Architecture "

7. Hartley, R.V.L; "Transmission of Information," Bell System Technical Journal, July 1928

8. Reza, Fazlollah M.; An Introduction to Information Theory. New York: Dover, 1994.

9. David J. C. MacKay; Information Theory, Inference, and Learning Algorithms Cambridge: Cambridge University Press, 2003.

10. DHCP Implementation and Security RFCs: 2939, 3004, 3011, 3046, 3118, 3203, 3256, 3361, 3396, 3397, 3442, 3456, 3495, 3527, 3594, 3634, 3679, 3825, 3925, 3942, 3993, 4014, 4030, 4039, 4174, 4280, 4361, 4388, 4390, 4578, 1541, 2489, 3315, 3319, 3646, 3633, 3898, 4075, 4242, 4280, 4776, 2855, 1542, 1534, 2131, 4361, 2132, 3942, 2485, 2563, 2610, 2855, 2937, 4649, 4580, 4704, 4833, 3315, 4361, 3319, 3633, 3646, 3736, 3898, 4075, 4076, 4280, and 4339.

Author:

Eugene Terrell

Principle Director
Research & Development

Engineering Theoretical Technologies
Research & Development Publications
(ETT-R&D Publications)

3312 64th Avenue Place
Oakland, CA. 94605
Voice: 510-636-9885
E-Mail: eterrell00@netzero.net

"This work is Dedicated to my first and only child, 'Princess Yahnay', because she is the gift of Dreams, the true treasure of my reality, and the 'Princess of the Universe'. (E.T. 2006)"

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).