Internet Draft
Category: Proposed Standard
Expires: January 27th, 2008

E. Terrell
ETT-R&D Publications
July 2007

The IPtX Domain Name Service Specification; IPtX-MX DNS

'draft-terrell-iptx-mx-dns-specification-04'

Status of this Memo

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

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

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

Intellectual Property Rights (IPR) Statement

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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.

This Internet-Draft will expire on January 27th, 2008.

Abstract

This document defines the IPtX Specification for the 'Domain Name Service' (IPtX / IPtX-MX DNS), and eliminates the possibility of an Addressing 'Conflict', or a Mathematical Addressing Error in the IPtX Address Space when using Multiple IPtX Addressing Formats. In other words, the IPtX / IPtX-MX IP Addressing format on the "Back-End", or "Backbone", obtains its uniqueness through the use and / or difference defined by the accuracy of the 'Exponential Decimal String'. However, this uniqueness, if not clarified, would not be discernable on the "Front-End", because the IPtX IP Addressing Specification 'Allows' only a '48' Bit-Mapped IP Address, or 2 Octets and 4 '10 Bit-Quadrants', and up to 60 Bits for every IP Addressing Format greater than IPt1. That is, on the "Front-End", if there is No distinction, because every Addressing Format in the IPtX Specification, when Resolved, is Equal, there will ultimately be Address Conflicts within the Addressing Scheme. Nevertheless, this clarification, when used in conjunction with the IPtX 32 / 64 Bit Header, allows the simultaneous use of (2E24) 16,777,216 different IPtX/IPtX-MX IP Addressing Specifications; the Concurrent use of the IPtX/IPtX-MX IP Addressing Specifications IPt1 thru IPt16,777,216.

Table of Contents

Introduction

The profoundness of the 'IPtX Specification' is that, it represents and defines a real conundrum. In other words, IP Addressing in the IPtX Specification, is a Mathematical Enigma that begs the question; 'How much does anyone really know about the Human Neuronic Processes? Or more specifically; 'Does anyone truly understand the Communication Process of the Neuron (perhaps, Macro and Quantum Levels), to actually develop a 'True Artificial Intelligence'? In which case, it should be understood; Today's Computers cannot discern the 'Identity', or 'Equality', between any two or more IP Addresses having Numerical Value(s) that actually define the same IP Address in the IPtX Specification - e.g.;

```
     213 = 00E0000.0000... ~ 2          E    7   .        73
 11010101                    11   11001010  111  .    1001001

        Bit-Mapped Length  =  110101011111001001  ~  18 Bits

     213 = 00E0000.0000... ~ 2          E    7   .        735
 11010101                    11   11001010  111  .  1011011111

      Bit-Mapped Length  =  110101011111011011111  ~  21 Bits
```

The distinction between the Binary Numerals is defined by the Accuracy of the 'Exponential Decimal String', which represents a Unique Binary Sequence from the Binary Set, {0,1}. However, while this clearly defines a valid conclusion, it is sustained only on the "Back-End". In other words, when converting the Binary Sequence into the Integer representing the IP Address, an additional Tag, which identifies (equaling the 'CIDR Network Descriptor') the Bit-Mapped Length of the Addressing Format being used is necessary when making a distinction - where;

```
11111111 : 11 11111111 'Variable Bit Length' . 'Variable Bit Length'

=    0000 :  2        E                 0000 .  0000...

2E0000 . 0000...  =  XXX:XXX:XXX.XXX.XXX.XXX  ±/0000:00

                  = IPtX IP Address
```

And given that the Display of the 'CIDR Network Descriptor' is replaced with the Display of the Name of the Addressing Format identifying the Bit-Mapped Length of the IPtX Addressing Format being used;

```
±/0000:00  =  IPtX  =  {IPt1, IPt2, ... IPt100, ... IPtX}
```

The User sees the Binary Conversion of '2E0000 . 0000...' only as the Integer which represents the IPtX IP Address - In other words, using the IPtX / IPtX-MX DNS 'IP Addressing Format Tag', which distinguishes the Addressing Specification using an 'A' to represent the number of 32 Bit Groupings the Addressing Format contains, prevents 'Front-End' Address Resolution Conflicts. - As given by;

'Front-End' Address Resolution - Conflict Avoidance Technique

' XXX:XXX:XXX.XXX.XXX.XXX /XA '

-- Or --

' XXX:XXX:XXX.XXX.XXX.XXX /IPtX '

/IPtX = {IPt1, IPt2, ... , IPt10,000, ... , IPtX} = /XA

213:112:238.009.212.001 /XA , or, 213:112:238.009.212.001 /IPtX

Where; 'X' = {Any Integer}, and 'A' = {One '4' Octet Group}
and the 'Preferred', since ; XA = 16 Bits:

/XA = 16 Bits = 2EX ; An 'IPtX / IPtX-MX DNS Tag'

2EX; 3Bit = Exponent = 2^3, 3 Bit Decimal String = 2^3

2EX = 2E8 .8 = The Number of ('A') Octet Groupings

(Yielding a 8 Bit Number with a 8 Bit Decimal String Accuracy)

Or - [/X ≥ 1, X ≤ 2^{24}; and A = Class ID - 'ID' = A, B, C, D, or E] = /XA

```
 ' 213 : 112 : 238 . 009 . 212 . 001  ±/XA '

 2^24 = Unused Bits                       3 State CIDR Network Descriptor
      = Internet Service            8 - 16 Bit -  Switch {'Ø','+','-', '/'}
        Provider 'ID'                  [Where 'Ø' means "No Sign" or '/']
   (Back-End ISP ID)                                  \   /
                                                       \ /
                       THE END-NODE OR FRONT-END        |
                              |  Network IP    |        |   8 - 16 Bits
                              |    Address     |        |     |
  8 Bit (10 Bit > IPt1)       | ~ 40 Bits Or   |        |     |   /XA =/ 2^24 : Class ID
   ZONE IP ADDRESS            | 4-8 Bit Octets |        |\    |  8 - 16 Bits
              \               |/    \     /    \|       | \   |    |
  '2EX' = [ XXX : XXX : 999 . 999 . 999 . 999 '±' / X    A  ] ~ 104 Bits
                   \                                 /     |    \
 8 Bit (10 Bit > IPt1) - IP AREA CODE ADDRESS       /      |     \
                                                   /      / \     \
                   16 thru 48 Bits  -   'IPtX / IPtX-MX DNS Tag'
```

Note: If - (Prefix x Zone IP x IP Area Code) = 2^{24};
(Network IP Address) ~ 2^{40}; (IPtX-MX DNS Tag) = 2^{48};
And - 104 Bit-Mapped Length = (2^{16}) x (2^{40}) x (2^{48})

Then - (2^{24}) x (2^{40}) x (2^{48}) = 0000:2E0000.0000... = IPt1
= [2^{24} = Unused Bits] x (2^{24}) x (2^{40}) x (2^{48}) = 2^{136}

'IPtX / IPtX-MX IP Addressing DNS Tag'

Given that; if the DNS Tag, in essence, represents a Function 'Call' that determines the IPtX Addressing Format being used, then it (the DNS Tag) must also determine the accuracy of the Exponential Decimal String - as given by;

0000:2E0000.0000... - 'DNS Tag' = /XA

0000:2Ē0000.0000... - 'DNS Tag' = /XĀ

Clearly then, any translation and / or distinction available to the user, must also be defined within the code of the Operating System, and in particular, defined within the code of the 'Domain Name Service' for the IPtX / IPtX-MX DNS Specification.

- GLOBAL IPtX / IPtX-MX IP Address 'DISPLAY' Example [NA] -

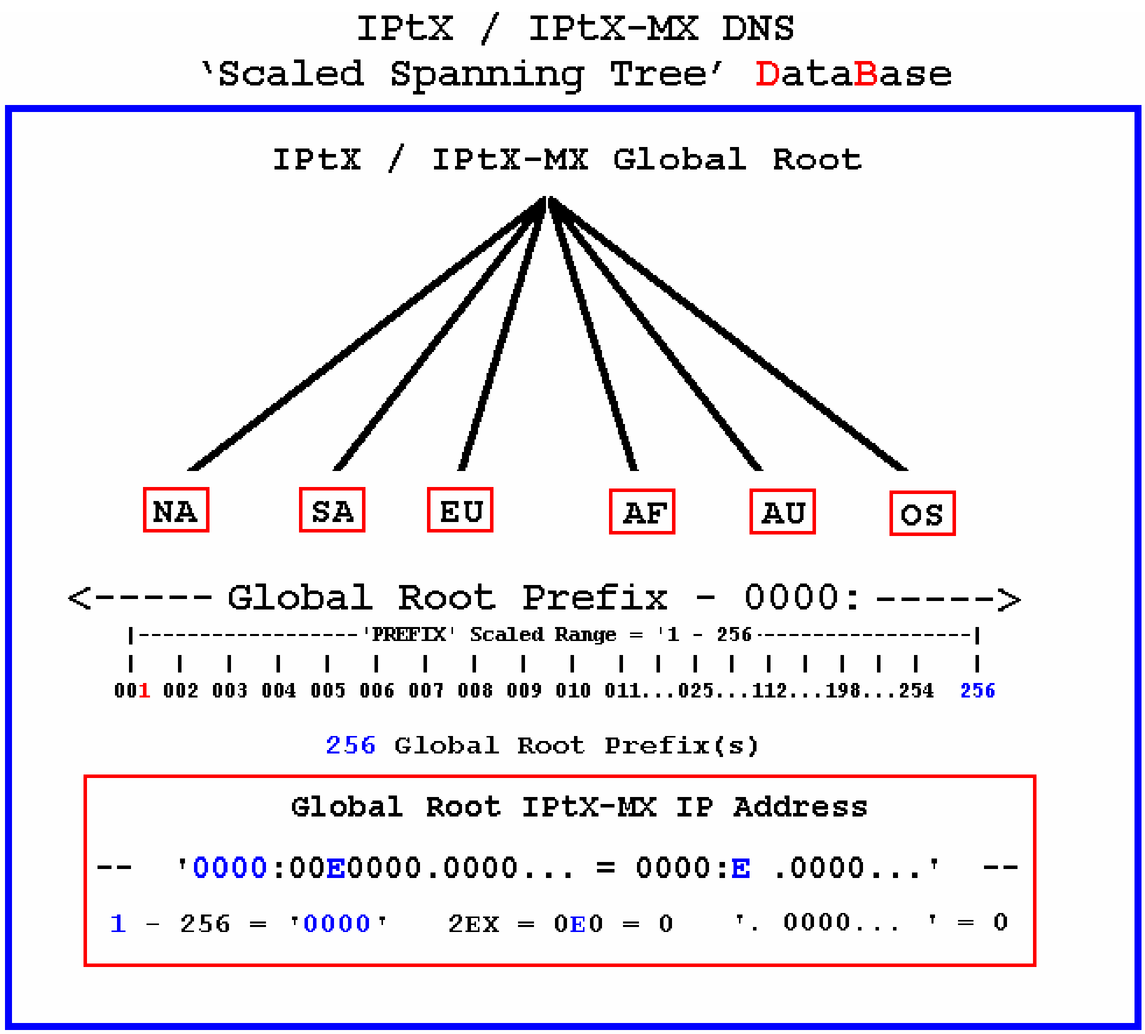
Prefix :	Zone IP :	IP Area Code :	IPtX IP Address	±/XXXX:XX
001:	213:	112:	238.009.112.001	±/XA
001:	213:	112:	238.009.112.001	±/X̄A

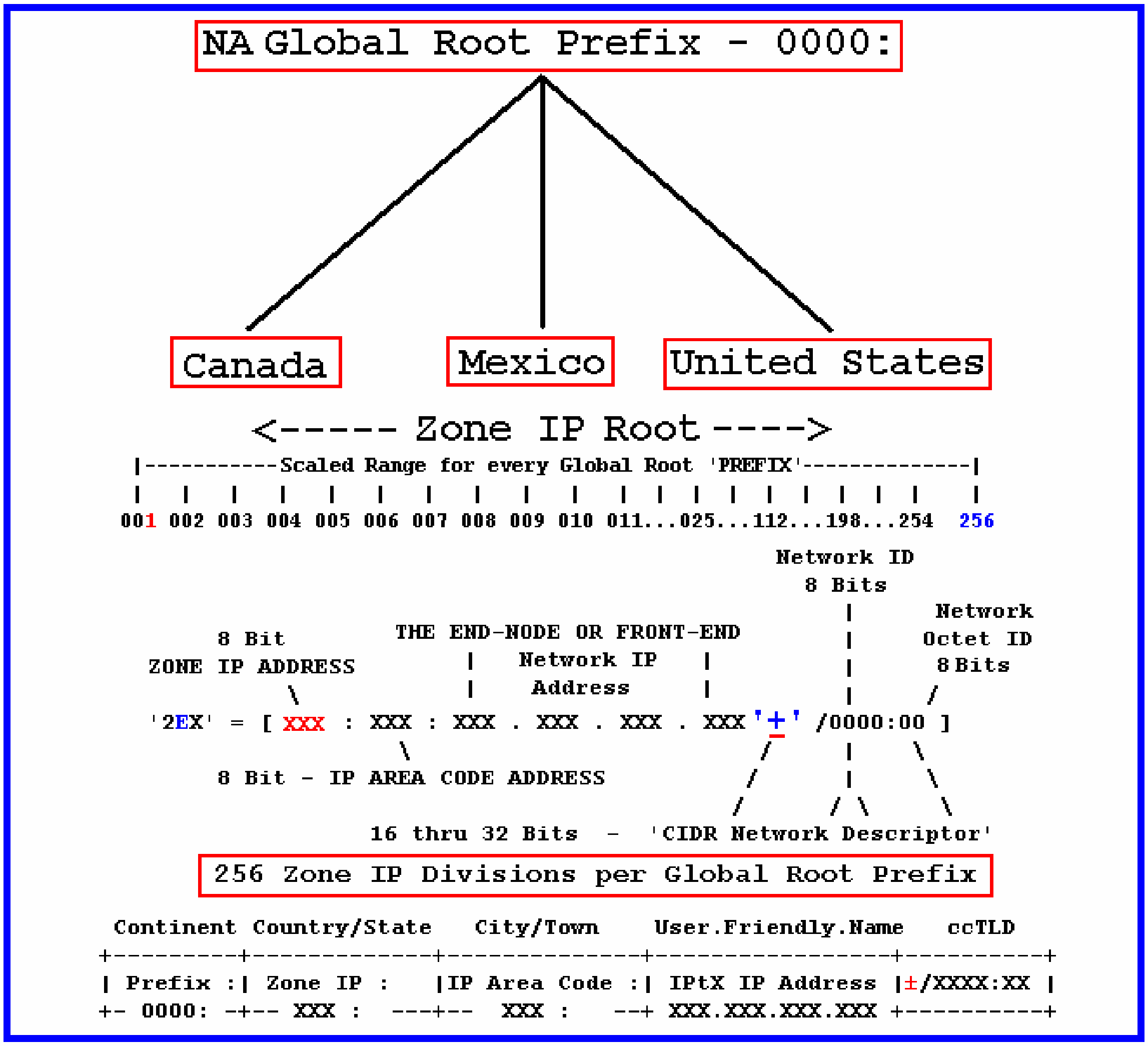
001:213:112:238.009.112.001 ±/XA

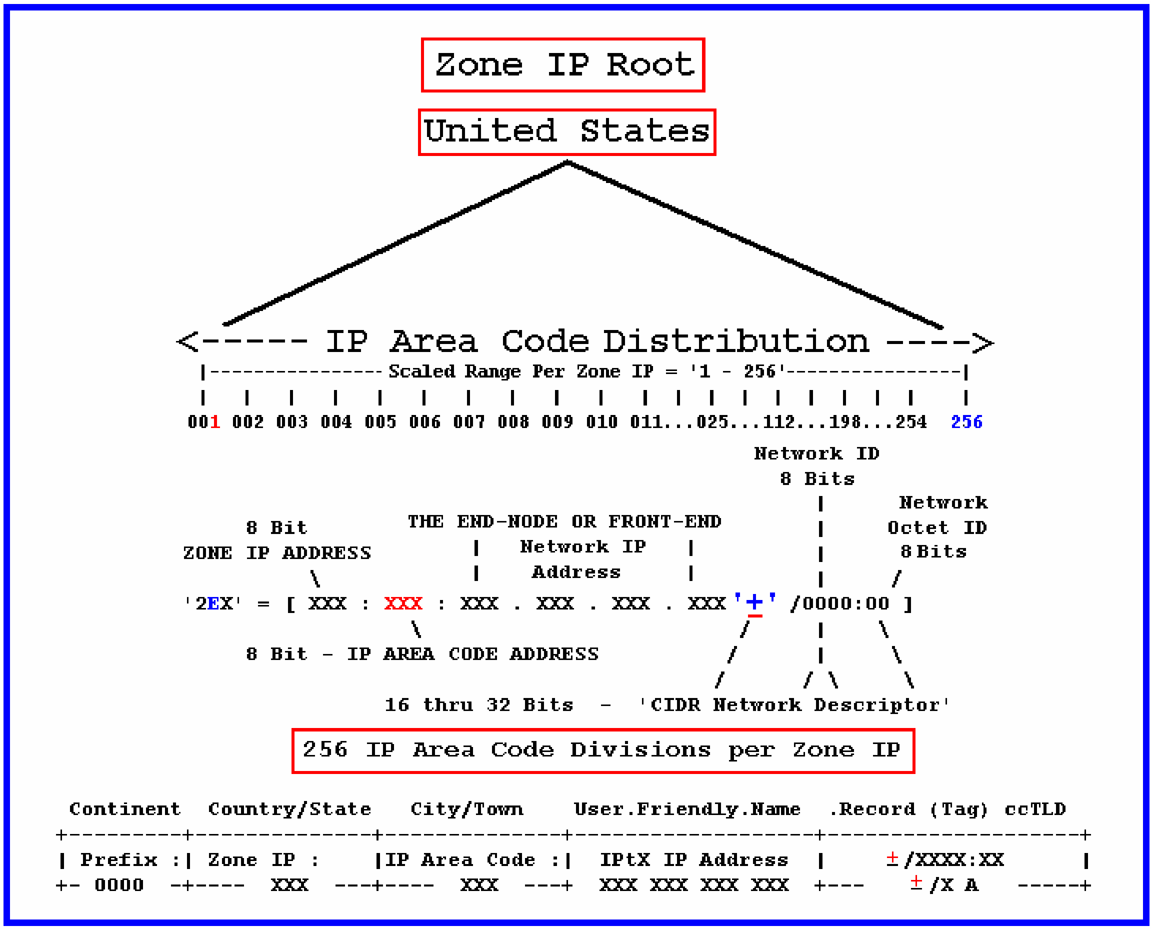
001:213:112:238.009.112.001 ±/X̄A

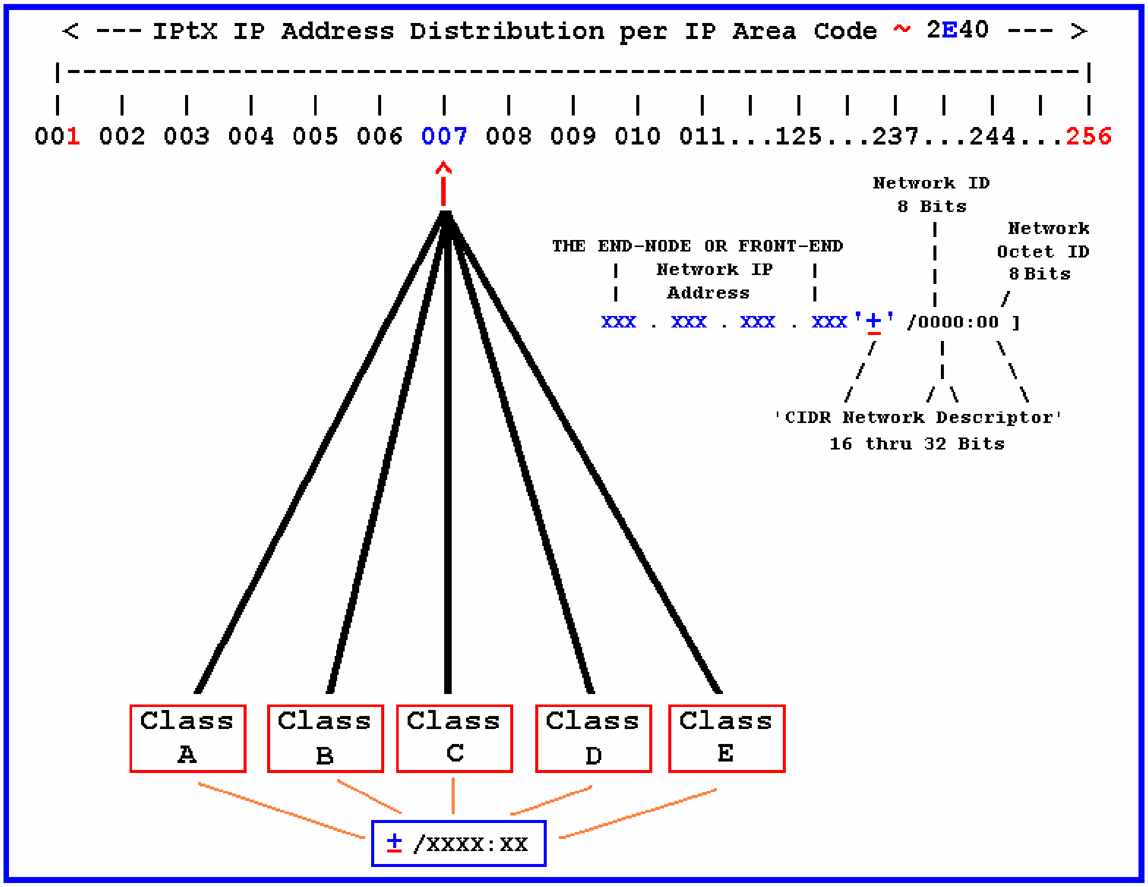
IANA Consideration

I. IPtX / IPtX-MX DNS 'Scaled Spanning Tree' Data Base for an IP Address;




NA Global Root Prefix - 0000:
Canada
Mexico
United States
<------ Zone IP Root ---->
|-----------Scaled Range for every Global Root 'PREFIX'--------------|
001 002 003 004 005 006 007 008 009 010 011...025...112...198...254 256
Network ID
8 Bits
Network
Octet ID
8 Bits
8 Bit
ZONE IP ADDRESS
THE END-NODE OR FRONT-END
Network IP
Address
'2EX' = [XXX : XXX : XXX . XXX . XXX . XXX '±' /0000:00]
8 Bit - IP AREA CODE ADDRESS
16 thru 32 Bits - 'CIDR Network Descriptor'
256 Zone IP Divisions per Global Root Prefix
Continent Country/State City/Town User.Friendly.Name ccTLD
| Prefix :| Zone IP : |IP Area Code :| IPtX IP Address |±/XXXX:XX |
+- 0000: -+-- XXX : ---+-- XXX : --+ XXX.XXX.XXX.XXX +----------+


Zone IP Root
United States
<----- IP Area Code Distribution ---->
|---------------- Scaled Range Per Zone IP = '1 - 256'----------------|
001 002 003 004 005 006 007 008 009 010 011...025...112...198...254 256
Network ID
8 Bits
Network
Octet ID
8 Bits
8 Bit
ZONE IP ADDRESS
THE END-NODE OR FRONT-END
Network IP
Address
'2EX' = [XXX : XXX : XXX . XXX . XXX . XXX '+' /0000:00]
8 Bit - IP AREA CODE ADDRESS
16 thru 32 Bits - 'CIDR Network Descriptor'
256 IP Area Code Divisions per Zone IP
Continent Country/State City/Town User.Friendly.Name .Record (Tag) ccTLD
| Prefix :| Zone IP : |IP Area Code :| IPtX IP Address | ±/XXXX:XX |
+- 0000 -+---- XXX ---+---- XXX ---+ XXX XXX XXX XXX +--- ±/X A -----+



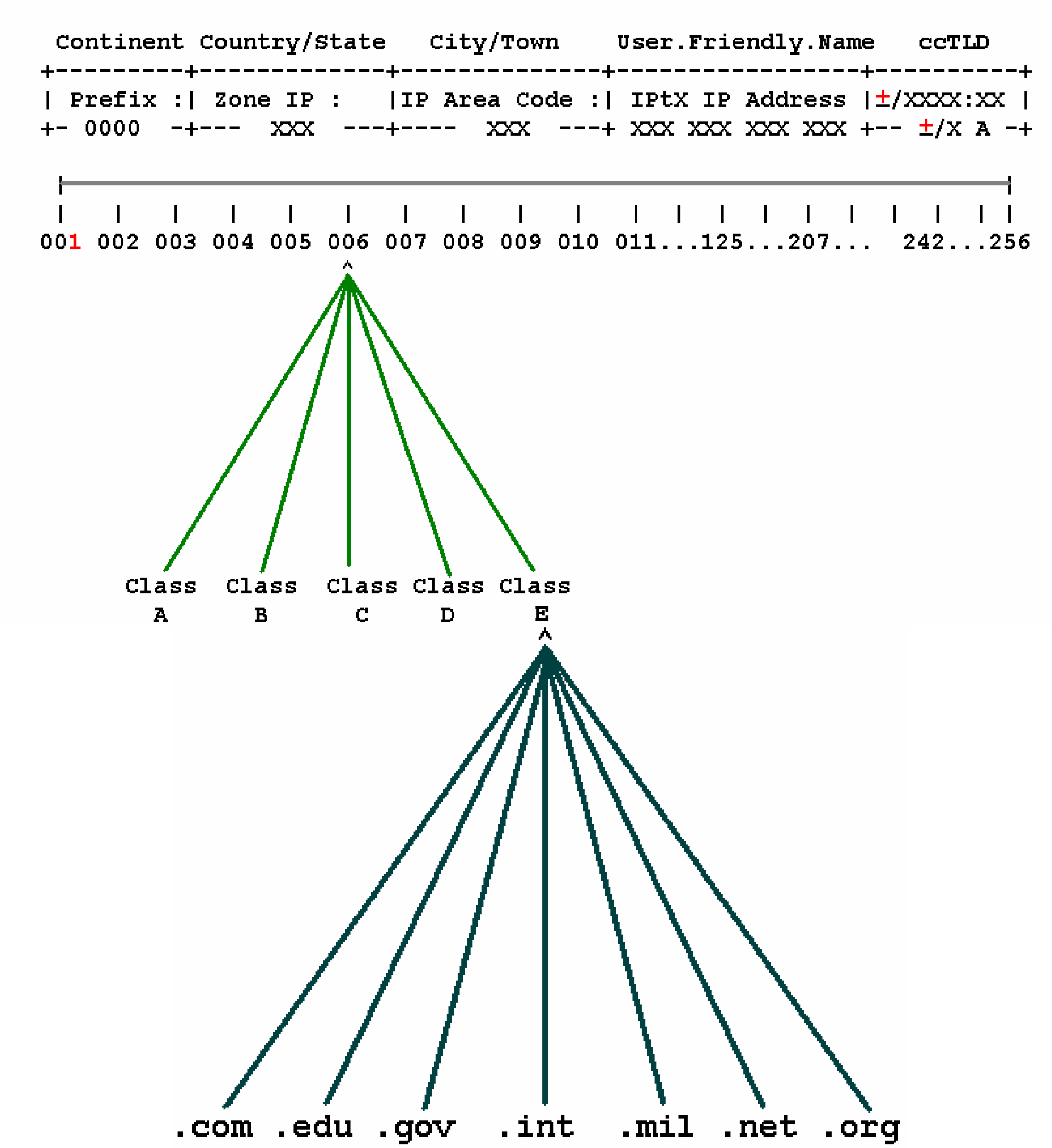
"IPtX Default 'Network Domain Name Address' Design Specification"

```
    Zone IP        IP Area Code          IP Address          ccTLD (Tag)
|-----XXX:-----|-------XXX:-------|----------------------|-------------|
 Country/State: City/Town/Village:    User.Friendly.Name    .Record Name

        |
        |
        V

Country/Province City/Town/Village    User.Friendly.Name     Record Name
|-----XXX:-----|-------XXX:-------|--- XXX XXX XXX XXX ---| ± /0000:00 |
```

Default IPtX Specification 'Network Domain Name Specification'



```
NOTE: Logical Argument supporting the introduction of the CLASS
      System Specification - defining the IP Area Code Sector(s);

   1) Backward Compatibility with the IPv4 Addressing Specification

   2) A IPtX / IPtX-MX DNS IP Addressing Tag used to distinguish
      between 2 or more different IPtX / IPtX-MX Addressing formats
```

Furthermore, because the IPtX Specification Counts Sequentially, even with the inclusion of the 'CLASS ID', the IPtX Specification remains CIDR Compliant. In other words, because it Counts sequentially, on the 'Back-End', everything is either a Large or a Small Binary Number; there are No Classes on the 'Backbone'. However, the actual benefit gained by using the CLASS System, is the Expansion of the IP Address Pool for the IPtX / IPtX-MX Specification, which defines the simultaneous use of Multiple IPtX IP Addressing Formats - As viewed from the 'Front-End', e.g., every 'IP Area Code Address' can be Divided into an Infinite Number of Sectors defined by up to 26 CLASS ID categories, which represents an Individual IPtX / IPtX-MX IP Addressing Specification; a distinct IPtX / IPtX-MX Addressing Format. (See pages 4 thru 7);

Recall that the IPtX-MX DNS IP Addressing Tag - '±/XA', defines;

```
X = Any Integer and A = One 4 Octet Group

Where; A =  Class A, Class B, Class C, ..., Class Z
- And every CLASS ID represents ONE 4 OCTET GROUP; /XA, /XB, /XC, ..., /XZ.
```

In other words, the 'CLASS ID' Designation allows the creation of a Rectangular Array, representing the expansion capabilities of the IPtX / IPtX-MX Addressing Specification - as given below;

```
        Consistent Rows representing an Identical IPtX Addressing Formats

 IPt1 = X = 1  Class A, Class B, Class C, ..., Class Z = /1 Class ID
 IPt2 = X = 2  Class A, Class B, Class C, ..., Class Z = /2 Class ID
 IPt3 = X = 3  Class A, Class B, Class C, ..., Class Z = /3 Class ID
 IPt4 = X = 4  Class A, Class B, Class C, ..., Class Z = /4 Class ID
 IPt5 = X = 5  Class A, Class B, Class C, ..., Class Z = /5 Class ID

        :      :        :        :       :     :     :      :
        :      :        :        :       :     :     :      :
        :      :        :        :       :     :     :      :

 IPtX*= X = ∞  Class A, Class B, Class C, ..., Class Z = /∞ Class ID


-- OR --


        Staggered Rows representing different IPtX Addressing Formats

 IPt1 = Class A, IPt2 = Class B, IPt3 = Class C, IPt6 = Class D, ... etc.
```

```
[*The Number of IPtX IP Addressing Formats that can be used Simultaneous, is
  Specified by the Range Limits of the Default Source and Destination Address
  Fields in the '32 / 64' Bit IPtX Header - See page [17]. e.g.; Currently,
  using IPt1, the maximum number of IPtX Addressing Formats (Values for 'X',
  the 32 Bit incremental progression defining a different IPtX Addressing
  Format) defined by the IPtX-MX DNS Tag, is equal to 2E24 (2^24  or
  'IPt16,777,216'); which means, 16,777,216 different IPtX Addressing Formats
  can be used simultaneously.]
```

The Expansion the CLASS ID System clearly provides, since every Column is defined by an Alpha Character following an Alphabet Sequence, defines a 'Rectangular Array' having;

26 Columns

The Number of ROWS however, is defined by the Limits of the IPtX IP Address Bit-Mapped Length specified in the IPtX 'Header'.

Furthermore, it should be understood, when analyzing the IPtX 32 / 64 Bit Header Specification (see page 17), only the Bit-Map of the Integer representing the Bit-Mapped Length for the Binary Numeral defining the Exponent and Exponential Decimal String, for every IP Address, is specified in the IPtX Header. In other words, the Exponent and the Exponential Decimal String defined in IPtX 32 / 64 Bit Header and the IPtX IP Addressing Specification is a Variable, which represents a Nested Exponential Base 2 Operation defining only the Bit-Map Length for the Exponent and the Exponential Decimal String. And more importantly, because the Exponent and the Exponential Decimal String defines a Binary Numeral instead of an Integer, the Number of IPtX Addressing Formats that can be used Concurrently and Handled by the IPtX Header increases astronomically. - As given by the examples below, we have;

```
                         Example 1.a

e.g. - If 0000:2E0000.0000... = 0000:2E78.32 Bits
     - where 2E78.32 = 78 Bit Exponent = 2^78 , and a
       32 Bit Exponential Decimal String = 2^32 ;
     - converts to the respective Integer(s) given by;

       a) 78 Bit Exponent =  2^78  = 302,231,454,903,657,293,676,544

       b) 32 Bit Exponential Decimal String =  2^32  = 4,294,967,296
```

In which case, only the Bit-Map of 2^{78}, representing the Exponent, and the Bit-Map of 2^{32} (defining the Bit-Mapped of the Result from a Base 2 Exponential Operation), defining the Exponential Decimal String, as in 2E78.32, are defined by the Bit-Mapped Address Fields in the IPtX 'Header' (see page 17). Or, as given by the second example;

```
                         Example 2.a

    where, e.g. - If 0000:2E0000.0000... = 2E198.868003799... Bits
      - where 2E78.32 = 198 Bit Exponent = 2^198 , and the
        868003799... Bit Exponential Decimal String ~ 2^30 ;
      - converts to the respective Integer(s) given by;

        a) 198 Bit Exponent = 2^198
           2^198  = 11000110

        b) 868003799... Bit Exponential Decimal String ~ 2^30
           2^30  ~ 868003799... = 110011101111001010111111010111

      - And 2E198.868003799... =  733,220,031,361,163,229,807,327,
                                  628,901,324,958,746,998,773,250,
                                  008,505,586,546 = 60 Digit Integer
```

In any case, once again, only the Bit-Map of 198, representing the Exponent, and the Bit-Map of '868003799...', defining the Exponential Decimal String, as in 2E198.868003799..., are defined by the Bit-Mapped Address Fields in the IPtX 'Header'. In other words, the Mathematical Operations involving the Masking and Un-Masking procedures, which resolves the Integer representing an IPtX / IPtX-MX IP Address from the Conversion of the Binary Bit-Mapped Transmission, are function(s) that are Hardware Specific - See TCP/IP and OSI Model(s). Hence, the size specification for the measurement of the Length of the Bit-Mapped IP Address Field(s) for the IPtX Header, represents a Nested Exponential Operation – as given by;

```
                       Example 3.a

SOURCE & DESTINATION ADDRESS Field(s)  =  2E 14 / 46 Bits

  =  2E 11111111111111 Bits = 14 Bits

  = 14 Bits =  2^14  = 16,384 Bit Exponent

  =  2E 1111111111111111111111111111111111111111111111 = 46 Bits

  = 46 Bits = 2^46  = 70,368,744,177,664 Bit Exponent
```

In other words, the Bit-Mapped Displacement defining the Length of the 'SOURCE and DESTINATION ADDRESS' Field(s) in the IPtX Header, just as with any IPtX-MX IP Addressing Format, as noted in Example 1.a, represents the Mathematical equation involving a NESTED Exponential Base 2 Operation. That is, given that the Exponent actually equals the Binary Conversion for the Integer equaling the Result from an Exponential Base 2 Operation. Then Example(s) '1.a' and '3.a' must represent the Bit-Map of the Binary Numeral defining the Integer (or Irrational Number) representing the Exponent and the Exponential Decimal String. While in Example 2.a, the Integer (or Irrational Number) representing the Exponent and the Exponential Decimal String, defines the Bit-Map Displacement or Binary Conversion of the Result from the Exponential Base 2 Operation.

```
[*Please Note: The use of Separate Fields to define the Exponent
  and the Exponential Decimal String in the IPtX Header is Preferred,
  because this method increases the size of the IP Address Pool for
  the IPtX / IPtX-MX Specification and the accuracy Exponential
  Decimal String.]
```

II. IPtX 32 / 64 Bit Header Design Specification - 'Variable IP Addressing Format Range Bit-Mapped Capacity' - e.g.; IPt1 thru IPtX

The current IP Bit-Mapped Transmission of an IP Address, is nothing more than the 'End or Station' Node Software Translation of a Binary Numerical Conversion. Clearly, utilizing the same principles and continuing the exploitation of the 'DCE Unit'. The Compression Range of a 22 to 54 Bit-Mapped IPtX IP Addressing Format, can be reduces to the Bit-Mapped Length of a 32 or 64 Bit Header.

```
                  IPtX 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           /
/                                                             /
|  IPtX Version = 2E21/53 = 21/53 Bits |  Parity Notify Bit*  |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|    Prefix    |DESTINATION ADDRESS Exponent = 2E 14 / 46 Bits|
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|DESTINATION ADDRESS Exponential Decimal String = 2E22/54 Bits|
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|  TTL / HOP LIMIT |   Option Section FLAGS =   16 / 32 Bits  |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|  IPtX Version = 2E21/53 = 21/53 Bits |  Parity Notify Bit*  |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|    Prefix     | SOURCE ADDRESS Exponent =  2E 14 / 46 Bits  |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
| SOURCE ADDRESS Exponential Decimal String = 2E 22 / 54 Bits |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|          2E10.12 Bits = Option Section = 2E24.30 Bits       |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|             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).
```

Internet Protocol tX (32 / 64 Bit) Address Space

IPtX/IPtX-MX IP Address = 0000:2E0000.0000...

```
 8 Bits   |8 Or 10 Bits| 8 Or 10 Bits |    ~ 40 Bits       | 16-48 Bits
---------+------------+--------------+------------------+----------
 Prefix :| Zone IP :  |IP Area Code :| IPtX IP Address  | ± /XXXX:XX
```

Prefix -8 BIT	Zone IP -8 BIT	IP Area Code -8 BIT	IPtX IP Address	CIDR Network Descriptor	Distribution Purpose	Date
None	None	None	000.000.000.000	None	None	7/2007
001	001-256:	All:	XXX.XXX.XXX.XXX	All	NA	7/2007
002	001-256:	All:	XXX.XXX.XXX.XXX	All	SA	7/2007
003	001-256:	All:	XXX.XXX.XXX.XXX	All	EU	7/2007
004	001-256:	All:	XXX.XXX.XXX.XXX	All	OS	7/2007
005	001-256:	All:	XXX.XXX.XXX.XXX	All	AU	7/2007
006	001-256:	All:	XXX.XXX.XXX.XXX	All	AF	7/2007
007-256	001-256:	All:	XXX.XXX.XXX.XXX	All	IANA/RESERVED	7/2007
001-256	001-256:	All:	— — XXX.XXX.XXX	All *	IANA/TELEPHONY	7/2007
None	None	None	127.000.000.000	None	IANA/LoopBack	7/2007
001-256	001-256:	All:	— — — — YYY.XXX	All	IANA/EMERGENCY	7/2007

SA = South America, NA = North America,
EU = European Union, AU = African Union,
AF = Asian Federation, OS = Oceania States

Note: For IPtX-MX Specifications > IPt1 the Zone IP and IP Area Code Divisions = 1 - '999'

IANA Emergency Broadcast IP Address

001-256	001-256:	All:	— — — — e.911	All	IANA/EMERGENCY	7/2007

Note: In addition to Character usage, Numerical Values can be used to allow the remaining IP Addresses in IANA EMERGENCY Section, to create a 'Network AnyCast/BroadCast Address Pool'. And more importantly, these are not Radio 'Broadcast Frequency Channel(s)' [See - 'Work(s) in Progress' [12].

Internet Protocol tX (32 / 64 Bit) Address Space

IPtX/IPtX-MX IP Address = 0000:2E0000.0000...

```
 8 Bits   |8 Or 10 Bits| 8 Or 10 Bits |    ~ 40 Bits       | 16-48 Bits
---------+------------+--------------+------------------+----------
 Prefix :| Zone IP :  |IP Area Code :| IPtX IP Address  | ± /XXXX:XX
```

Prefix -8 BIT	Zone IP -8 BIT	IP Area Code -8 BIT	IPtX IP Address	CIDR Network Descriptor	Distribution Purpose	Date
None	None	None	1/000,000,000,000	None	None	7/2007
001	001-256:	All:	1/XXX.XXX.XXX.XXX	All	NA	7/2007
002	001-256:	All:	1/XXX.XXX.XXX.XXX	All	SA	7/2007
003	001-256:	All:	1/XXX.XXX.XXX.XXX	All	EU	7/2007
004	001-256:	All:	1/XXX.XXX.XXX.XXX	All	OS	7/2007
005	001-256:	All:	1/XXX.XXX.XXX.XXX	All	AU	7/2007
006	001-256:	All:	1/XXX.XXX.XXX.XXX	All	AF	7/2007
007-256	001-256:	All:	1/XXX.XXX.XXX.XXX	All	IANA/RESERVED	7/2007
001-256	001-256:	All:	1/— — XXX.XXX.XXX	All *	IANA/TELEPHONY	7/2007
None	None	None	1/127.000.000.000	None	IANA/LoopBack	7/2007
001-256	001-256:	All:	1/— — — — YYY.XXX	All	IANA/EMERGENCY	7/2007

SA = South America, NA = North America,
EU = European Union, AU = African Union,
AF = Asian Federation, OS = Oceania States

Note: For IPtX-MX Specifications > IPt1 the Zone IP and IP Area Code Divisions = 1 - '999'

Note: For IPtX-MX IP Addressing Specifications > IPt1, the Zone IP and IP Area Code Divisions = '001 - 999'; resulting in a 60 Bit Users IPtX / IPtX-MX IP Address:

" 999: 999: 999.999.999.999 '±/XA' = 2EX ".

III. IPtX 32 / 64 Bit - DNS Header, DNS Query, DNS Resource Record, TCP Header, TCP Pseudo Header, UDP Header, and UDP Pseudo Header, Design Specification(s) -

CHANGES: IPtX DNS Services 32 / 64 Bit Header

```
 DNS Header for IPtx      |  DNS Query for IPtX  | DNS RR Record for IPtX
                          |                      |
   Identification =       |  Type = 2E18.20 Bits |  Type = 2E18.20 Bits
   16 / 2E10.12 Bits      |                      |
   Opcode = 4 Bits        |    Class = 16 Bits   |    Class = 16 Bits
                          |                      |
 Rcode = 4 Bits           |   Length Rdata =     |   TTL = Variable to
                          |      2E12.20 Bits    |      2E22.40 Bits
------------------------|----------------------|-----------------------
                             |         4 New "TYPE" Categories
TQuestions = 2E12.20 Bits    |
                             | 1. TYPE 43 = 'RNN'
TAnswers RR = 2E12.20 Bits   |            = "Reverse Network Domain Name"
                             |    Title: IN-ADDR.APARA NAME
                             |            = IN-ADDR.RNN
TAuthority RR = 2E12.20 Bits |
                             | 2. TYPE 44 = 'RNN-PTR'
                             |            = "Reverse Network Domain
                             |               Name-Domain Name Pointer"
TAdditional RR = 2E12.20 Bits|
                             |  2 New IPtX DNS Tag(s) Specifications
                             |
CIDRNetDes = XXXX:XX         |3. TYPE 45 = XA = "IPtX (IP Address)"
  /XXXX:XX = 8 Bits          |  Where X = Integer Variable ≥ 1
                             |  - e.g. IPt1 = A, IPt2 = AA = 2A,
                             |  IPt3 = AAA = 3A, IPt4 = AAAA = 4A, etc
                             |
                             |4. TYPE 46 = XĀ = "IPtX (IP Address)"
```

Note: Approximating the Number of Digits representing the Integer from a Binary Conversion, or the Bit-Mapped Displacement / Length of an Integer, is more accurately defined as;

(Binary) Bit-Map Length ÷ 3 ≈ Number of Digits (Integers)

e.g.; 1GB = 2E30 = 2^{30} = 1,073,741,824 Bit Length
≈ 357,913,941 Digit Number (Integer)

```
                       DNS Header 32 Bit IPtX

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
| Identification-16 Bits |QR|Opcode|AA|TC|RD|RA|Z|AD|CD|Rcode |
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
      Total Questions          |        Total Answer RRs
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
     Total Authority RRs       |     Total Additional RRs
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
                           Questions
                        = 2E10.12 Bits
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
                           Answer RRs
                        = 2E10.12 Bits
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
                           Authority RRs
                        = 2E10.12 Bits
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
                          Additional RRs
                        = 2E10.12 Bits
```

```
                       DNS Header 64 Bit IPtX

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
|     Identification
       2E10.12 Bits       |QR|Opcode|AA|TC|RD|RA|Z|AD|CD|Rcode |
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
      Total Questions =   2E10.12 Bits =   Total Answer RRs
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
  Total Authority RRs = 2E10.12 Bits = Total Additional RRs
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
                           Questions
                      =    2E24.30 Bits
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
                           Answer RRs
                      =   2E24.30  Bits
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
                           Authority RRs
                      =    2E24.30 Bits
 + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
                          Additional RRs
                      =    2E24.30 Bits
```

```
                  DNS Query 32 / 64 Bit IPtX

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
/                    DNS Query 32 Scale IPtX                   /
/                                                              /

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
/                    DNS Query 64 Scale IPtX                   /
/                                                              /

/              Field Information - IPtX DNS Query              /
|                         Query Name                           |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|   32 Bit = 2E10.12 Bits   :::  64 Bit = 2E24.30 Bits         |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|   32 Bit = 2E10.12 Bits   :::  64 Bit = 2E24.30 Bits         |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
| Type = 16 Bit / 2E10.12 Bit |  Class =  16 Bit / 2E10.12 Bit |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|   32 Bit = 2E10.12 Bits   :::  64 Bit = 2E24.30 Bits         |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|   32 Bit = 2E10.12 Bits   :::  64 Bit = 2E24.30 Bits         |
|--------------------------------------------------------------|
```

```
                DNS Resource Record 32 / 64 Bit IPtX

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
/                                                               /
/            DNS Resource Record 32 Scale IPtX                  /

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
/                                                               /
/            DNS Resource Record 64 Scale IPtX                  /

/                                                               /
/      Field Information - DNS Resource Record                  /
|                             Name                              |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|           32 / 64 Bit = 2E10.12 Bit / 2E24.30 Bits            |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|             32 / 64 Bit = 2E10.12 Bit / 2E24.30 Bits          |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
| Type = 16 Bit / 2E10.12 Bit | Class =  16 Bit / 2E10.12 Bit |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|   32 Bit = 2E10.12 Bits    TTL    64 Bit = 2E24.30 Bits       |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|Length Rdata = 16 Bit/2E10.12 Bit|Rdata = 16 Bit / 2E10.12 Bit|
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|           32 / 64 Bit = 2E10.12 Bit / 2E24.30 Bits            |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|           32 / 64 Bit = 2E10.12 Bit / 2E24.30 Bits            |
|---------------------------------------------------------------|
```

```
                      IPtX TCP Header 32 64 Bit

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
/                    TCP Header 32 Scale IPtX                       /
/                                                                   /

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
/                    TCP Header 64 Scale IPtX                       /
/                                                                   /

 /                  Field Information - TCP Header                   /
 /                                                                   /
 |Source Port= 16/2E10.12 Bits|Destination Port =16/2E10.12 Bits|
 |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 |      2E10.12 Bits  = Sequence Number = 2E24.30 Bits            |
 |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 | 2E10.12 Bits  =  Acknowledgment Number = 2E24.30  Bits         |
 |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 |DataOffset 4Bit|Resrvd|ECN|Control Bits6|Window 48Bit HEX No.|
 |+ + + + + +2+ + + + + + + + + + + + + +3+ + + + 24+ + + + + + +|
 |Checksum=16 Bit/2E10.12 Bit|Urgent Pointer=16 Bit/2E10.12 Bit
 |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 |   2E10.12 Bits  = Options and padding  = 2E24.30 Bits          |
 |                            :::                                 |
 |+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
 |          2E10.12 Bits  =  Data  = 2E24.30  Bits                |
 |                            :::                                 |
 |----------------------------------------------------------------|
```

```
                 IPtX 32 / 64 Bit TCP Pseudo 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               /
/                                                                     /
|  IPtX Version = 2E21/53 = 21/53 Bits |  Parity Notify Bit*  |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|    Prefix    |DESTINATION ADDRESS Exponent = 2E 14 / 46 Bits|
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|DESTINATION ADDRESS Exponential Decimal String = 2E22/54 Bits|
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|  TTL / HOP LIMIT |   Option Section FLAGS =   16 / 32 Bits   |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|  IPtX Version = 2E21/53 = 21/53 Bits |  Parity Notify Bit*  |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|    Prefix    | SOURCE ADDRESS Exponent =  2E 14 / 46 Bits   |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
| SOURCE ADDRESS Exponential Decimal String = 2E 22 / 54 Bits |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|          2E10.12 Bits = Option Section = 2E24.30 Bits        |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|               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).
```

```
                UDP header for IPtX 32 / 64 Bits

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                        /
/                                                               /
|Source Port = 16 / 2E10.12 Bits | Destination Port  = 16 / 2E10.12 Bits |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
| Length  =  16 / 2E10.12 Bits    |   Checksum  = 16 / 2E10.12 Bits      |
|+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +|
|            2E10.12 Bits = Data = 2E24.30 Bits                 |
|                             :::                               |
|                             :::                               |
|---------------------------------------------------------------|
```

```
NOTE: There is room for improvement, i.e. TTL = 2E24.30... Bits! This is
      clearly Ridiculous... However, this represents Backwards
      Compatibility at its Best. At least now, Networking the Moon, or
      perhaps Mars, seems plausible. Nevertheless, the difference between
      the Fields defining the Headers for the 'Binary' IPtX-MX IP
      Addressing Specification, which pertains specifically to IP
      Addressing, only involves the Mathematics associated with Changing
      the Exponential Operator, E, to Ē. However, it also necessitates the
      specification defining the use of a "T" for 'TELe' and an 'M' for
      Mobile, as the Operational Function(s) defined 'by / in' the CIDR
      Network Descriptor for Transmission and Front-End' Address
      Resolution. In which case, 'X'becomes; 'X = T (TELe)', or 'X = M
      (Mobile)', in '/XA', which respectively represents the '/TA', and
      '/MA' IPtX-MX DNS Tags for use in Telephony and Mobile
      Transmissions.
```

Security Considerations

- IANA Isolation and Privatization Options -

There are many instances, not only for Businesses, Governments, and the Military, but ordinary Citizens also require a Secure Networking and Telephony Environment - However, while there are a number of methods employed today, which addresses these concerns, the IPtX Specification provides IANA with additional tools to make these environments even more secure - e.g.;

I. IANA/Telephony* - IPtX / IPtX-MX DNS Tag "/TA"
'Depending upon whether or not the Telephony Address Pool is Shared by every IPtX Addressing Format'

Provides every IPtX Addressing Format with Enhanced Isolation when using the IPtX / IPtX-MX DNS Tag "/TA" Specification

II. ISP - "Backbone ID"
Enhanced Security for 'All IPtX Addressing Formats'

III. Users - "Backbone Account Number ID" - Enhanced Privatization

Enhanced Security for 'All IPtX Addressing Formats' beyond the IPt1 Specification - an Address Bit-Mapped Length assigned by IANA, with respect to the IPtX Addressing Format being used.

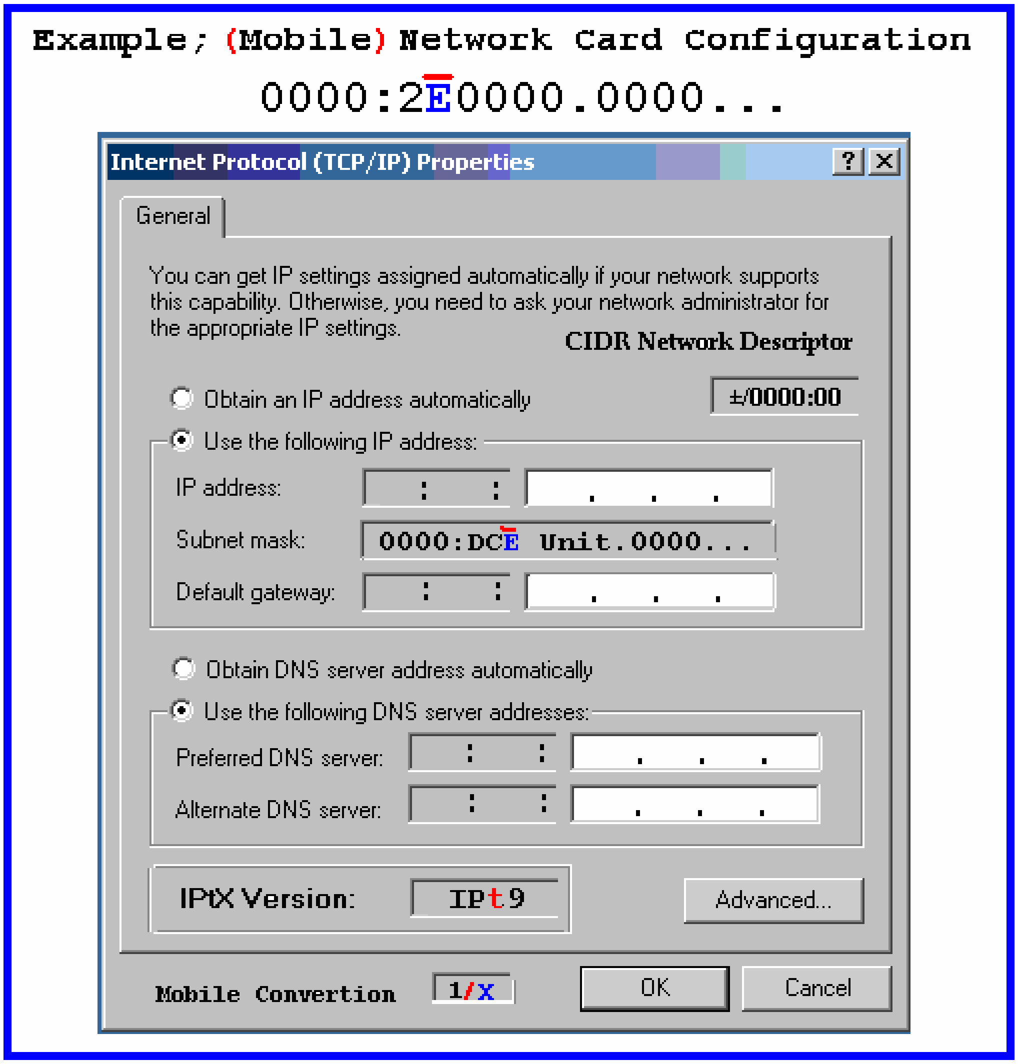
IV. Network - "Backbone Account Number ID"- Isolation and Privatization

Enhanced Security for 'All IPtX Addressing Formats' beyond the IPt1 Specification - an Address Bit-Mapped Length assigned by IANA, with respect to the IPtX Addressing Format being used – Network Isolation.

V. IANA/EMERGENCY - "BROADCAST" - Isolation and Privatization
Defined by / in the 'CIDR Network Descriptor Specification'

VI. Options for the "IPtX-MX '0000:2Ē0000.0000...' Address Band"

If a 'Binary IP Addressing Protocol Specification' is unnecessary, then the more suitable application for the '0000:2Ē0000.0000...' Address Band Specification defines a 'Routable Encrypted / Encryption Protocol, which can be used to Encrypt; the 'ISP Backbone ID', the 'Users Backbone Account Number ID', the 'Network Backbone Account Number ID', and the Users IPtX IP Address: 'The Isolation and Privatization (or 'Cloaking') of any Network sharing the 'Backbone Connections'.



The 'IPt 16,777,216' "Intelligent Quantum Tunneling Worm Protocol"

```
   536,870,912 Bit-Mapped Length     =     0000:00E0000.0000...


         2E8 = 8 Bit Prefix
                  \
                   \     2E402,653,166 = 402,653,166 Bit Exponent
                    \                        /
                     \   2E2 = 2 Bit Base   /
                      \        |           /
                     0000  :  00    E  0000  .  0000...
                                    |            \
              2E8 = 8 Bit Exponential Operator    \
                                                   \
      2E134,217,728  = 134,217,728 Bit Exponential Decimal String
```

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-00.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 $\pi : \tau$ { 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. DNS Implementation and Security RFCs: 2535, 2931, 2135, 1035, 1996, 2845, 2930, 2671, 1183, 1706, 2163, 1712, 1886, 1876, 1002, 2052, 2782, 2168, 2915, 2538, 2230, 2671, 2672, 2874, 1995, 3123, 1996, 2182, 1101, 1123, 1279, 1296, 1383, 1401, 1464, 1480, 1535, 1536, 1591, 1611, 1612, 1713, 1794, 1876, 1886, 2163, 2168, 2219, 2230, 2308, 2517, 2538, 2539, 2541, 2606, 2845, 2870, 2915, 2929, 2930, 2931, 3007, 3008, 3090, 3110, 3027, 3071, 3130, 3123, 3152, 2537, 2137, and 2065.

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).