Internet Draft **E. Terrell**
Category: Proposed Standard **ETT-R&D Publications**
Expires: October 27th, 2006 **April 2006**

The CIDR Network Descriptor expands the size of the IPtX Address Space beyond the IPv6 IP Addressing Specification

'draft-terrell-cidr-net-descrpt-expands-iptx-add-spc-15.pdf'

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.

This Internet-Draft will expire on October 27th, 2006.

Abstract

This document, which Obsoletes RFC 2373, RFC 1517, RFC 1518, RFC 1519, and IEEE Specification 1541-2002 (Re-defining the Electromagnetic Spectrum and the 'SI Units' as a Base 2 Exponential Binary Conversion - defines the Bit-Map Specification for expressing any Numeral that represents a Decimal Fraction), provides the final clarification of the conclusions that redefines the 'CIDR' notation as the 'Network Descriptor', and proves that the IP Address Pool Total for the IPtX Specification is greater than IPv6. And more importantly, because these conclusions reveal the actual design of the Binary Communication System, the Revolutionary impact sustained, is an upheaval affecting the entire field of Computer Science; 'The Rudiments Finite of Quantum Computing and Finite Quantum Computer Programming'. In other words, IPtX is a more powerful and cost effective IP Addressing Specification, and when using the 'IPtX-MX Protocol' {'2^{X} : 1'; the Compression Ratio for "The Intelligent Quantum Tunneling Worm Protocol" - The Design of the 'Internet Protocol telecommunications Xchange Specification'}, the interface of the "Front-End" can mimic or simulate a 32 Bit-Mapped IP Address. And this, in conjunction with the IPv4 IP Addressing Overlay, provides a 100% Backward Compatibility with the IPv4 Specification (Meeting the Requirements of RFC 1550), in the Backbone environment approaching an unlimited size 'Bit-Map' Address Space.

Table of Contents

Introduction

The "CIDR" Concepts, the 'Network Descriptor' and the 'Bit-Mapped' IP Address, provided the bases for the comparison between IPv4, IPv6, and the IPtX IP Addressing Specifications, which concluded that these are different expressions of equal definitions. That is, mathematically speaking, the IP Address Pools of the IPv4 and the IPv6 Specifications, mathematically defines the variable Coefficient of an Identical Base Pool of IP Addresses, or $X(2^{32})$; given that 'X' respectively equals '1' and '7.9228163×10^{28}'.

Where IPv4 = $1(2^{32})$; and the 128 Bit IPv6 $\approx 7.9228163 \times 10^{28} (2^{32}) = 2^{128} = (2^{4(32)})$.

However, the IP Address Pool for the IPtX Specification is mathematically defined by the equation:

$$IPtX = X(2^{32}) + 16,900,000.$$

Nevertheless, these additional IP Addresses, because only the "Network Descriptors" are different, (approximately 16.9 Million Shared) are Host or Client IP Addresses, which cannot be used to establish a direct Internet Connection because of the Logic problems, the decisional conflicts with the Routers. In other words, the expansion of the 'CIDR' concept(s), which mathematically defines the expression; '$X(2^{32})$', can only provide a Bit-Map that uniquely Identifies every IP Address within every Address Class, or the IP Address Range the expression defines, essentially availing the entire Range of IP Addresses to the Global-Net.

1. The Classless Inter-Domain Routing Architecture, or CIDR

The Classless Inter-Domain Routing Architecture, or CIDR, was derived from the so called; "strategies for address assignment of the existing IP address space with a view to conserve the address space and stem the explosive growth of routing tables in default-route-free routers" [9]. It was in reality, an Expansion of the 'Default Addressing Structures' existing in the Address Class System. The popular claim nevertheless, boasted the elimination of the Address Class System. These Claims needless to say, were fashioned by the Authors, whose works represented their personal interpretation(s), because the works comprising RFC's 1517, 1518, and 1519 were never fully understood. The truth nonetheless, was clearly explained in RFC 1519, whose discourse dealt specifically with the way the Routers, and the Routing Protocols interpreted, or dealt with the IP Address, and not the elimination of the Address Class System per se. In other words, the Routers and the Routing Protocols were limited to using only the 'Default Addressing Formats', which represented Class A, Class B, and the Class C Addressing Specification. And to deal with the prospect, or the possibility of an IP Addressing Shortage, a plan was devised (RFC's 1517, 1518, and 1519), which actually involved not only the initial 'Default Addressing Formats', from Class A, B, and C, but the remaining fractional subcomponents from each of their respective Octets as well. In fact, while RFC 1519 specifically designed the CIDR Architecture to take advantage of Class C, it did not weaver in its mention of the same implementation for the Class A Specification. It could be said in other words, that the CIDR Architecture represents an Un-Finished version of the 'IPtX Protocol Family Specification'. However, because of the MISNOMER, 'CLASSLESS', the process of SUB-DIVIDING a Class (In particular; Class A, and Class C), was never fully understood. Hence, the CIDR Architecture is the Sub-division of a CLASS SYSTEM, or a Class Addressing System that has been SUB-DIVIDED, which represents the Class, or the Whole, having a Greater Number of Constituents.

In other words, the CIDR Architecture actually represents: The 'Inter-Domain IP Bit Mapped Address Routing Architecture'. Because this, in essence, is what is really happening to the IP Address, and this is the IP Addressing Format that the Router and the Routing Protocols are dealing with. Therefore, the Class Addressing System is a format that implements a Network IP Address using a specified number of BITs, and in this case, it is either '8', '16', '24', or '32' Bits. Needless to say, any further Sub-Division, or use of some Smaller Portion or Constituent, does not constitute a departure nor eliminate the Existence of the Address Class System. Hence, the CIDR Architecture actually reinforced the Class Concept and proved that, without changing the entire Addressing Architecture defining the 'IP Bit Mapped Address or the IP Bit Mapped Address Space', the Whole is indeed the Sum of its Parts.

2. The Interpretation of the conclusion's Expansion of 'CIDR'; Defining the "Network Descriptor"

When defining the New 'CIDR' Architecture as representing the collective Extension for RFC's 1517, 1518, and 1519, we must first list the functional components, or Highlights, noted as the objectives or purpose supporting each of these papers, individually. That is, there must be comparison between the definition or description of the functional purpose of the 'CIDR' Architecture as represented in each of these papers, compared with the New 'CIDR' Architecture this paper actually represents.

RFC 1517 (Maintained promoted a fear of IP Address Loss, and Astronomical growth in the size of the Routing Tables):

"- Exhaustion of the class-B network address space. One fundamental cause of this problem is the lack of a network class of a size that is appropriate for a mid-sized organization. Class-C, with a maximum of 254 host addresses, is too small, while class-B, which allows up to 65534 addresses, is too large to be densely populated. The result is inefficient utilization of class-B network numbers.

- Routing information overload. The size and rate of growth of the routing tables in Internet routers is beyond the ability of current software (and people) to effectively manage.

- Eventual exhaustion of IP network numbers."

Argument in Opposition (Justification of the New 'CIDR' Architecture):

Clearly, using the expanded 'CIDR' Architecture, when using the 'CIDR' Network Descriptor, the Reality of IP Address Exhaustion now defines a wasted use of IP Addresses. In other words, Viable IP Address that could have been use to establish an Internet Connection, Connecting the Network's Backbone-Domain to the Internet, were assigned for the Host IP Addresses. This is further clarified by a Comparison of the "Internet Protocol v4 Address Space, and the use of the CIDR Network Descriptor displayed in Table I:

TABLE I

IPtX - Internet Protocol t1 Address Space INDEX

IPaddNum = Network IP Address
CIDRNetDescrip = CIDR Network Descriptor

Current Number of IP Network Addresses Issued
Accounts for = 253 IP Network Addresses

Class A	CIDR Network Descriptor
A-1: Issued = 127 , Remaining = 1,040,513,921	/0000:08
A-2: Issued = None, Remaining = 516,160,512	/0000:16
A-3: Issued = None, Remaining = 256,048,128	/0000:24
A-4: Issued = None, Remaining = 252,047,376	/0000:32

TABLE I

Class B	CIDR Network Descriptor
B-1: Issued = 64 , Remaining = 784,514,496	/1000:08
B-2: Issued = None, Remaining = 197,672,960	/1000:16
B-3: Issued = None, Remaining = 49,807,360	/1000:24
B-4: Issued = None, Remaining = 16,777,216	/1000:32
Class C	
C-1: Issued = 32 , Remaining = 458,321,632	/1100:08
C-2: Issued = None, Remaining = 57,741,312	/1100:16
C-3: Issued = None, Remaining = 7,274,496	/1100:24
C-4: Issued = None, Remaining = 1,048,576	/1100:32

TABLE I	
Class D	**CIDR Network Descriptor**
D-1: Issued = 16 , Remaining = 245,676,912	**/1110:08**
D-2: Issued = None, Remaining = 15,475,712	**/1110:16**
D-3: Issued = None, Remaining = 974,848	**/1110:24**
D-4: Issued = None, Remaining = 65,536	**/1110:32**
Class E	
E-1: Issued = 15 , Remaining = 231,289,845	**/1111:08**
E-2: Issued = None, Remaining = 13,658,850	**/1111:16**
E-3: Issued = None, Remaining = 806,625	**/1111:24**
E-4: Issued = None, Remaining = 50,625	**/1111:32**

RFC 1518 (Which deals more with the actual Structure of the Internet, or its Hierarchical Structure , and IP Address allocation and Routing, than the actual 'CIDR' Architecture) where by, the points are specified as:

There are two aspects of interest when discussing IP address allocation within the Internet. The first is the set of administrative requirements for obtaining and allocating IP addresses; the second is the technical aspect of such assignments, having largely to do with routing, both within a routing domain (intra-domain routing) and between routing domains (inter-domain routing). This paper focuses on the technical issues.

The architecture and recommendations in this paper are oriented primarily toward the large-scale division of IP address allocation in the Internet.

IP Addresses and Routing

Efficiency versus Decentralized Control

IP Address Administration and Routing in the Internet

Administration of IP addresses within a domain

Indirect Providers (Backbones)*

Continental aggregation*

Argument in Opposition (Justification of the New 'CIDR' Architecture):

While there is a lot that can be said regarding RFC 1518, especially since this is a proposal which advocates a great deal of dependency upon ISP's, whose entire existence is based upon the Economy, the Consumer, and a Volatile Market. This actually means, an ISP has no guaranteed Future, regarding either the use of the IP Address Base, or their Routers for a thoroughfare. In other words, while this RFC did mention some good points, these arguments are supported in the IPtX Specification. It nevertheless, maintained more the soundings of a White Paper Solicitation for a New System Overall, than an actual presentation representing 'CIDR' Architecture. Needless to say, some of the problems discussed, and emphasized repeatedly, addressed the need for a Internet Hierarchy, while dismissing the need to expand the number of Backbone connections, this is the main point of consideration when addressing the concept of an Internet Hierarchy.

RFC 1519 (While this RFC should be the replacement for RFC 1517, because it is clearly derived from RFC 1517, it claims to Obsoletes RFC 1338, which I have not read. And while this paper also disputes some of the proposals outlined in RFC 1518 {Noting Specifically the causes for a loss of aggregation efficiency; Organizations, which are multi-homed, and Organizations, which change, service provider but do not renumber.}. Nevertheless, one thing this RFC does, that the others so far do not, is that, it Mathematically Introduces the beginnings of Foundation for the 'CIDR' Architecture.)

Argument in Opposition (Justification of the New 'CIDR' Architecture):

Nonetheless, while this RFC introduces the basic Mathematical Foundation for the 'CIDR' Architecture, and sets the fundamentals for the hardware and software specifications for Networking in a Supernetted Environment, it actually does nothing to prevent IP Address wasted on Host Assignments. This is because the foundation of the 'CIDR' Architecture was derived from the IPv4 specification, which means there was no way, short of a New IP Addressing System, could this waste be avoided. However, this is not the problem with the IPt1 specification, because it De-Emphasizes the HOST IP Address, and gives it secondary functional value, which defines a dummy Host. Needless to say, this was the foundation that was needed to get the 'Ball Rolling'. Nevertheless, while RFC 1519 developed the Mathematical foundation for the 'CIDR' Architecture, it never fully Exploited the benefits this Architecture maintains.

Definitions

CIDR: Classless Inter-Domain Routing is an IP Address Resolution Technique that provides a way to resolve any Binary Number(s) into it's the Integer Translation to verify an IP Address, which is written in 'Dotted Notation', and defined by ' 4 ' 8 Bit Octets.

CIDR Network Descriptor: It is a 2 Part Number used to resolve, or discover the Integer representing the Binary Number defining the Network Address IP Bit-Mapped.

[Where by, the Digits to the Right of the Colon Represents the Starting Point for the IP Address Class Range in Binary Notation, or Network IP Address assigned to the Specified IP Address Class Range. And the Digits to the Left of the Colon represent the Count of Bit Mapped Displacement, or the Number of Binary Digits the Network IP Address uses. In which case, the '/XXXX:XX' notation would be used to Identify the Bit-Mapped Address Class and the Address Class Range of a Network IP Address.]

Nevertheless, the definitions noted above, concludes the argument by providing logical support for Re-Defining the 'CIDR' notation as the "Network Descriptor", which comprises a Switch, and a 2-part Number that defines the entire Range of every IP Address, including the Address Class Range mapping every Octet for the Network portion of the IP Address defines.

TABLE II

IPtX - Internet Protocol t1 Address Space INDEX

IPaddNum	=	Network IP Address (XXX.XXX.XXX.XXX)
CIDRNetDescrip	=	CIDR Network Descriptor (/XXXX:XX)
CIDRNetDesSwitch	=	CIDR Network Descriptor Switch (/)
CIDRNetDesClassID	=	A "4" place Binary Number (XXXX) Identifying the 'Front-End' of the Bit-Mapped Space of the Network Address in the Range of the Address Class.

CIDRNetDesDivider	=	Statement End-Start New Statement (:)
CIDROctDesNetID	=	A "2" place, or variable Number (XX) Identifying the enrire Bit-Mapped Range of an IP Address. (IPv6 has "3" places)
ZONEIPaddNum	=	Zone IP designates the Continent's location, and it is the First of a 2 Octet configuration defining the Prefix of a 32 Bit IP Address, which is 8 Bit Number Terminated by a Colon (XXX:)
IPAreaCodeaddNum	=	IP Area Code designates the second level of the Continent's Sub-Region, and defines the Second of the 2 8 Bit Octet configuration Prefixing a 32 Bit IP Address that is also Terminated by a Colon (XXX:)
GlobalIPaddNum	=	GlobalNet IP Address (XXX:XXX:XXX.XXX.XXX.XXX)
CIDRTrunkNetID	=	The combined use of the 'Zone IP'and the 'IP Area Code' to identify the "Trunk-Identifier", which is assigned to the 'TelCo-Xchge or Backbone Routers'

3. The IPtX and IPv4 IP Addressing Schemes –100% Compatibility

The IPtX IP Addressing Scheme is a logically derived 'Internet Protocol Addressing Family' that is founded upon the IPv4 IP Addressing Specification. In other words, because IPtX utilizes the same Bit-Mapped Binary Addressing Format, which does not require any deviation from the operational infrastructure of the IPv4 Address Space. It maintains a 100% Backward Compatibility with IPv4, which is retained throughout an expansion capability defining an Infinitely Bit-Mapped IP Address Space using only a 64 Bit Header. There is however, the one noted difference in the IPv4 and the IPtX Specifications, which defines a distinction between the respective "Subnet-Mask" and the "Subnet-Identifier" that allows the IPtX Address Pool to be greater than IPv4 and IPv6, when using the same Bit-Mapped IP Address Space. Needless to say, while this distinction might at first, appear to be a strong departure from the IPv4 Format. It's only a 'Binary Switch' that uses the elements from the Set, {X,Y} to provide the ability the Change ((between) or (Extend)) the 'Programmed Functions' or 'Operations' defining the "Subnet-Mask" and the "Subnet-Identifier". In any case, it should be clearly understood, the definition of the "Subnet-Identifier" extends the definition of the "Subnet-Mask" by providing the ability to Change the Range of the Subnet, which specifies the range of the Host IP Address. And this, it should be noted, is the Distinguishing Mathematical Hallmark that assigns every IP Address in the IPv4 Specification to the IP Address Pool, with the added benefit of the creation of a Host IP Address Pool. Now, if the Address Spaces were equal, the number of available IP Address in the IP Address Pool assigned to the IPv4 Specification, given that IPtX is defined as the extension of IPv4, exceeds the number of available IP Address in the IP Address Pool assigned to the IPv6 Specification. Nevertheless, the validity of the forgoing is supported by the conclusions from the mathematical analysis and comparison of Tables III, IV, and V.

TABLE III

Decimal Structure of the IPv4 Representation IP Class System

IPv4 IP Address Pool = 4.145 x 10^{9} Addresses

1. **Class A: 1 - 126, Default Subnet Mask 255.X.X.X:**
   126 Networks and 16,387,064 Hosts: 0000

2. **Class B: 128- 191, Default Subnet Mask 255.255.X.X:**
   16,256 Networks and 64,516 Hosts: 1000

3. **Class C: 192 223, Default Subnet Mask 255.255.255.X:**
   2,064,512 Networks and 254 Hosts: 1100

4. **Class D: 224 - 239; Used for Multicasting, No Host: 1110**
   16 x 254^{3} = 262,192,024 IP Addresses available

5. **Class E: 240 - 254; Denoting Experimental, No Host: 1111**
   15 x 254^{3} = 245,805,960 IP Addresses available

TABLE IV

IPtX - 'Subnet-Identifier' = "Subnet-Mask" - 100% Backward Compatibility with IPv4
"IPtX Addressing System Using the Current Binary System."

[Note: The Law of the Octet defines the value, relative to Class, of the 'Y' variable in the IP Address Pool containing; 4,145,927,192 = 4.145×10^{9} Addresses, which should be: 2^{32} = 4,294,967,296]

1. Total IP Addresses for Class A = 126×254^{3} = 2,064.770,064
   Total available IP Host Addresses Equals 126×254^{N}
   [Where N = Number of Octet(s), and 'Y' equals the Address Range '128 - 254', 1 - 126 is not included in the Address Range Represented by the equation 'Y = 254 - 126'.]

Class A-1, 1 - 126, Default Subnet-Mask 255.Y.X.X:
126 Networks and 8,129,016 Hosts: /0000:08

Class A-2, 1 - 126, Default Subnet-Mask 255.255.Y.X:
15,876 Networks and 32,004 Hosts: /0000:16

Class A-3, 1 - 126, Default Subnet-Mask 255.255.255.Y:
2,000,376 Networks and 126 Hosts: /0000:24

Class A-4, 1 - 126, Default Subnet-Mask 255.255.255.255:
252,047,376 Network / MultiCast IP Addresses / AnyCast: /0000:32

TABLE IV - Continued

2. Total IP Addresses for Class B = 64 x 254^3 = 1,048,772,096
Total available IP Host Address es Equals 64 x 254^N
[Where N = Number of Octet, and 'Y' equals the Address Range '254 - Q'; 128 - 191 is not included in the Address Range Represented by the equation 'Y = 254 - 64'.]

Class B-1, 128 - 191, Default Subnet Mask 255.Y.X.X:
64 Networks and 4,129,024 Hosts: /1000:08

Class B-2, 128 - 191, Default Subnet Mask 255.255.Y.X:
4,096 Networks and 48,260 Hosts: /1000:16

Class B-3, 128 - 191, Default Subnet Mask 255.255.255.Y:
262,144 Networks and 64 Hosts: /1000:24

Class B-4, 128 - 191, Default Subnet Mask 255.255.255.255:
16,777,216 Network / MultiCast IP Addresses / AnyCast: /1000:32

3. Total IP Addresses for Class C = 32 x 254^3 = 524,386,048
Total available IP Host Addresses Equals 32 x 254^N
[Where N = Number of Octet, and 'Y' equals the Address Range '254 - Q'; 192 - 223 is not included in the Address Range Represented by the equation 'Y = 254 - 32.]

Class C-1, 192 - 223, Default Subnet Mask 255.Y.X.X:
32 Networks and 2,065,512 Hosts: /1100:08

Class C-2, 192 - 223, Default Subnet Mask 255.255.Y.X:
1,024 Networks and 8,128 Hosts: /1100:16

Class C-3, 192 - 223, Default Subnet Mask 255.255.255.Y:
32,768 Networks and 32 Hosts: /1100:24

Class C-4, 192 - 223, Default Subnet Mask 255.255.255.255:
1,048,576 Network / MultiCast IP Addresses / AnyCast: /1100:32

TABLE IV - Continued

4. Total IP Addresses for Class D = 16 x 254^{3} = 262.193,024
Total available IP Host Addresses Equals 16 x 254^{N}
[Where N = Number of Octet, and 'Y' equals the Address Range '254 - Q'; 224 - 239 is not included in the Address Range Represented by the equation 'Y = 254 - 16'.]

Class D-1, 224 - 239, Default Subnet Mask 255.Y.X.X:
16 Networks and 1,032,256 Hosts: /1110:08

Class D-2, 224 - 239, Default Subnet Mask 255.255.Y.X:
256 Networks and 3,048 Hosts: /1110:16

Class D-3, 224 - 239, Default Subnet Mask 255.255.255.Y:
4,096 Networks and 16 Hosts: /1110:24

Class D-4, 224 - 239, Default Subnet Mask 255.255.255.255:
65,536 Network / MultiCast IP Addresses / AnyCast: /1110:32

5. Total IP Addresses for Class E = 15 x 254^{3} = 245,805,960
Total available IP Host Addresses Equals 15 x 254^{N}
[Where N = Number of Octet, and 'Y' equals the Address Range '254 - Q'; 240 - 254 is not included in the Address Range Represented by the equation 'Y = 254 - 15'.]

Class E-1, 240 - 254, Default Subnet Mask 255.Y.X.X:
15 Networks and 967,740 Hosts: /1111:08

Class E-2, 240 - 254, Default Subnet Mask 255.255.Y.X:
225 Networks and 3,810 Hosts: /1111:16

Class E-3, 240 - 254, Default Subnet Mask 255.255.255.Y:
3,375 Networks and 15 Hosts: /1111:24

Class E-4, 240 - 254, Default Subnet Mask 255.255.255.255:
50,625 Network / MultiCast IP Addresses / AnyCast: /1111:32

The mathematical analysis of the result from Table III and Table IV reveals that when the "Subnet-Mask" is equal to the "Subnet-Identifier". The division of the Address Classes in the IPv4 Specification, is defined by the "Network Descriptor" as; the Logical Expansion of each of the Address Classes in the IPv4 Specification, through the use of the Network ID designation, "255", in remaining Octets. And because the requirement of the "Subnet-Mask" mandates that 'only a Number from the Range of the Address Class' can be assigned a Network ID, then every Octet specifies one of the '4 Sub-Division' created in the Range of every Address Class. In other words, because the "Subnet-Mask" specifies the uses of the "255" designation in conjunction with the 'Range of the Address Class', to identify the Address Class associated with the Network ID. The logical use of the Octets remaining in the Range of every Address Class in conjunction with the "Subnet-Mask", is the logical consequence of the conclusion derived from the expansion, or sub-division of the 'Address Class Range' that was outlined in RFC 1517, 1518, and 1519. Needless to say, since 'the Quantified Sum of the 'Product of the Network and Host IP Addresses' for every Address Class in Table IV, is equal to the Total Number of IP Address assigned to the Range of the Address Class defined by Table III, then the Addressing Specifications shown by these Tables are mathematically equal. In which case, mathematically speaking, is should be concluded that the IPtX Specification, by RFC 1517, 1518, and 1519, is the interpretation of the IPv4 Specification derived from the use of the "Subnet-Mask".

4. The Structural Comparison - IPv6 vs. IPtX

The handicaps from using an askew Binary System, the Loop-Back Address (127), and the Subnet-Mask (255), makes it impossible for the IPv4 Specification, even through the progressive expansion using 32 Bit additions for an equal Address Space, to match the IP Addresses available in the IP Address Pool of the IPv6 Specification. And clearly, the same fate, according to Figures 1 and 2, will befall the IPtX Specification. However, because the IPtX Specification maintains an expansion capability that defines an Infinitely Bit-Mapped IP Address Space, using only a 64 Bit Header. In which, the 'TelCo-Xchge or Backbone Routers' would be assigned a Network ID, which would absorb the Bit-String of the Address Space beyond the 32 Bit-Mapped IP Address Space defined by the IPt1 Specification. And since, the losses represented by Figures 1 and 2, at infinity, are not discernable. The IPtX Specification maintains an IP Address Pool Capacity, which is theoretically, infinitely larger than the IP Address Pool availability in the IPv6 Specification (See Figure 3).

IPv4 Specification

IPv4 = 32 Bit Address Space

IPv4 IP Address Pool = $253(254^{3})$ = 4,145,927,192 = 4.145×10^{9} Addresses
IPv4 IP Address Pool Specification = 4,294,967,296 = $X(2^{32})$
This represents a loss: 4,294,967,296 - 4,145,927,192 = 149,040,104 IP Addresses

IPv6 Specification

IPv6 = 128 Bit Address Space

IPv6 IP Address Pool = $(2^{32}) \times (2^{32}) \times (2^{32}) \times (2^{32}) = 2^{128} = X(2^{32})$
IPv6 IP Address Pool Specification =
340,282,366,920,938,463,463,374,607,431,768,211,456 IP Addresses = $X(2^{32})$

IPtX Specification

IPt1 = 32 Bit Address Space

IPt1 IP Address Pool = (255^{4}) = 4,228,250,625 + 16,500,000 IP Addresses
IPt1 IP Address Pool Specification = $X(2^{32})$ + 16,500,000 IP Addresses
= 4,294,967,296 + 16,500,000 IP Addresses
This represents a loss: 4,294,967,296 - 4,228,250,625 = 66,716,671 IP Addresses

Figure 1

IPtX Specification

IPt2 = 64 Bit Address Space

IPt2 = 48 Bit IP Address = $(255^{2})(255^{4}) = X(2^{32}) + 16{,}500{,}000$ IP Addresses
IPt2 IP Address Pool $= (255^{2})(255^{4}) = 65{,}025\ X(2^{32}) + 16{,}500{,}000$ IP Addresses
$= 65{,}025(4{,}228{,}250{,}625) + 16{,}500{,}000$
$= 2.7494200 \times 10^{14} + 16{,}500{,}000$ IP Addresses

IPt2 IP Address Pool Specification $= X(2^{32}) + 16{,}500{,}000$ IP Addresses
$= (256^{2})(2^{32}) = 2.8147498 \times 10^{14}$
This represents a loss: $2.8147498 \times 10^{14} - 2.7494200 \times 10^{14}$
$= 6.5329799 \times 10^{12}$ IP Addresses

IPt2 = 64 Bit IP Address $= (255^{4})(255^{4}) = X(2^{32}) + 16{,}500{,}000$ IP Addresses
IPt2 IP Address Pool $= (255^{8}) = 1.7878103 \times 10^{19} + 16{,}500{,}000$ IP Addresses
$= 1.7878103 \times 10^{19} + 16{,}500{,}000$
$= 1.7878103 \times 10^{19} + 16{,}500{,}000$ IP Addresses

IPt2 IP Address Pool Specification $= (256^{4})(2^{32}) = 1.8446744 \times 10^{19}$
$= X(2^{32}) + 16{,}500{,}000$ IP Addresses
This represents a loss: $1.8446744 \times 10^{19} - 1.7878103 \times 10^{19}$
$= 5.6864072 \times 10^{17}$ IP Addresses

Figure 2

Note: The IP Addressing Design Specification implements the IP Addressing Operational Design Procedures for the "Front-End" and the "Back-End"... Specifically noting that; IPv6 is a "Back-End" only, Addressing Specification: IPv4 is a "Front-End" and "Back-End" IP Addressing Specification: The IPtX Design however, is a Dual Addressing Specification that offers a choice between a design specification that is 100% backward compatible with the "Front-End" and the "Back-End" of IPv4, or a design that to mimics the "Front-End" of IPv4, with a new "Back-End" design specification that defines an Alternate "Electro-Magnetic Spectrum", which is a (New) "Binary" Interpretation" of the "Electro-Magnetic Spectrum" {"Quantum Theoretical Physics"; the Logical Foundation resulting from the discovery of the "Quantum Scale"}.

'IPtX IP Specification'
(Topology of the Internet Backbone Hierarchy)

Trunk-Id Address Size Spec.		Zone IP Area Code Address Spec.		IP Address Spec.		IP Header BITS Size Spec.		Bit-Mapped IP Address Space Size Spec.		Network IP Address Size Spec.
None	=	None	=	IPt1	=	32 Bit	=	32 Bit	=	32 Bit
32 Bit	=	16 Bit	=	IPt2	=	64 Bit	=	64 Bit	=	32 Bit
64 Bit	=	16 Bit	=	IPt3	=	64 Bit	=	96 Bit	=	32 Bit
96 Bit	=	16 Bit	=	IPt4	=	64 Bit	=	128 Bit	=	32 Bit
128 Bit	=	16 Bit	=	IPt5	=	64 Bit	=	160 Bit	=	32 Bit
:	:	:	:	:	:	:	:	:	:	:
3.168k Bit	=	16 Bit	=	IPt100	=	64 Bit	=	3,200 Bit	=	32 Bit
:	:	:	:	:	:	:	:	:	:	:
159,968 Bit	=	16 Bit	=	IPt5000	=	64 Bit	=	160,000 Bit	=	32 Bit
:	:	:	:	:	:	:	:	:	:	:
XXXXX Bit	=	IPt 1.00×10^{11}			=	:		3.200×10^{12} Bits		:
:	:	:	:	:	:	:	:	:	:	:
XXXXX Bit	=	IPt 1.00×10^{14}			=	:		3.200×10^{18} Bits		:
:	:	:	:	:	:	:	:	:	:	:
Infinity	=	16 Bit	=	IPtX	=	64 Bit	=	Infinity	=	32 Bit

The "Trunk-Identifier" Specification (Spec.) contains the Zone IP and IP Area Code. And given that Software can assign the Zone IP and IP Area Code to any location/destination. Eventually it will be necessary to know the 18 Digit IP Address: ZoneIP, IP Area Code, Network IP Address. And noting that any Trunk-Id beyond 96 Bits might seem ridiculous, because of the inherent limitations of the design specifications for the "Network Descriptor", (+/0000:00). However, assigning a 'Router-ID' that requires Special Authentication between communicating Routers could easily absorb any Number of Bits beyond the IPt4 Specification.

Figure 3

Figure 3

INTERNET PROTOCOL t2 (64 Bit) ADDRESS SPACE

Reserved BITS		CIDR Network Descriptor	Zone IP	IP Area Code	IPt1 Address Schematic IP Address Assignment	Distribution Purpose	Date	
8	8	None	000:	000:	000.000.000.000	None	4/2002	
8	8	All	001:	256:	XXX.XXX.XXX.XXX	NA	4/2002	
8	8	All	002:	256:	XXX.XXX.XXX.XXX	SA	4/2002	
8	8	All	003:	256:	XXX.XXX.XXX.XXX	EU	4/2002	
8	8	All	004:	256:	XXX.XXX.XXX.XXX	OS	4/2002	
8	8	All	005:	256:	XXX.XXX.XXX.XXX	AU	4/2002	
8	8	All	006:	256:	XXX.XXX.XXX.XXX	AF	4/2002	
8	8	All	007-256:	256:	XXX.XXX.XXX.XXX	IANA/RSRVD	4/2002	
8	8	NO-IDs	001-256:	000-256:	000.XXX.XXX.XXX	IANA/EMGNCY	4/2006	*
8	8	±/0000:00	256:	256:	127.000.000.000	IANA/LopBck	4/2002	

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

Figure 4

Note: While noting that any Trunk-Id beyond 96 Bits might seem ridiculous, because of the inherent limitations of the design specifications for the "Network Descriptor", (+/0000:00). However, assigning a 'Router-ID' that requires Special Authentication between communicating Routers could easily absorb any Number of Bits beyond the IPt4 Specification.

Nevertheless, it should be clearly understood, that if the 'TelCo-Xchge or Backbone Routers' were assigned the "Trunk-Identifier", then there must be a way to verify, resolve, or confirm the "Zone IP and the IP Area Code" Address defining the "Trunk-Identifier". In other words, the IPtX Specification can maintain the "Subnet-Identifier", use the "Network Descriptor" to verify the "Trunk-Identifier", and exhaust, without loss, every available IP Address in the Address Pool defined by the equation 1 (eq-1):

$$\text{eq-1:} \quad \text{IPtX} = X(2^{32}) + 16{,}900{,}000 = \infty$$

$$\text{eq-2:} \quad \text{IPv6} = X(2^{32}) = (2^{128}) = (256^{16}) = 3.4028237 \times 10^{38}$$

The process of maximizing the Address Pool defined by equation 1, is a '3' Step procedure, in which the first step exhaust all of the available IP Addresses in the IPv4 Specification. The second step requires changing the Symbol used as the "Network Designator" for the "Subnet-Identifier" (255 or 256), and the third step requires expanding the function of the Switch for the "Network Descriptor". To achieve the first goal, multiply the Network and Host IP Addresses from Table IV, and use the product to represent the total number of Networks, leaving the Host total unchanged. And after the "Network Descriptor" and the "Subnet-Identifier" has been redefined, as provided in Table V. The adjusted results now reveals that the total number of available IP Addresses in the Address Pool for the IPtX Specification is equal the equation 1, and it exceeds the amount assigned to the IPv6 Specification, when the size of the Bit-Mapped Address Space is equal to both Addressing Specifications (See Table VI).

TABLE V - Definitions

"Network Descriptor": A method derived from the CIDR notation, which is used to resolve and identify every part of a Network IP Address. And when it defines the use of '2' State Switch, comprising a 'Statement 'End-Start' New Statement Command symbol' and a 2 Part numeral, it separates an IP Address into 2 components, which identifies the Network IP Address and the Trunk-Identifier.

CIDR Network Descriptor ' ±/0000:00 '

1) '2' State Switch: " ±/ " - CIDRNetDesSwitch
   a) "Network-Identifier" Switch: ' -/ ' - CIDRGetIPaddNum
   b) "Trunk-Identifier" Switch: ' ± ' - CIDRTrunkIDNum

2) 'End-Start', Flip/Flop Command symbol: '(:)' - CIDRGetTrunkIDNum

3) 'Network ID': ' 0000 ' - CID RNetDesClassID

4) 'Octet Bit-Map': ' 00 ' - CIDROctDesNetID - Address Length

"Subnet-Identifier": Defines the Overllay that is used in conjunction with a "Network Designator" and the 'X' and 'Y' variables to resolve a Networks IP Address.

1) "Network Designator": The symbol for the Null Set, ' Ø ', which is used as the 'Place Holder', identifying the OCTET(s) that define the Network ID, or the Network portion of the 32 Bit IP Address.|

2) "Octet using eitire Address Range: ' X '

3) "Octet not using "Network Designator" Addresses: ' Y '

Note: From Table V, the "Network Designator", which is symbolized as; ' Ø ', could just as easily, been assigned to defined any symbol, (e.g.; '257') except for ' 0 '(representing ' 00 ', the Binary equivalent of ' 1 ') and any assignment represented in the Range of the Addressing Specification. In any case, it should be clearly understood that when every available IP Address is used for IP Addressing. The IPtX Addressing Schematic, (where fig. 5 and fig. 6 respectively denotes IPt1 and IPt2 Specifications) becomes nothing more than an OVERLAY, which is used to facilitate the visualization of the Topology for the Structure of the Network Design. And this is an extremely important advantage when designing a Network containing hundreds (or thousands) of Servers and several thousand (or Million) Hosts assigned to Subnets. [It is important to note, ' Ø ' represents the 'NULL SET' or TRUE ZERO {the Traditional European Representation for True Zero}, and ' 0 ' is EQUAL to ' 00 ', which defines the Binary equivalence of ' 1 ': ' 00 ' = ' 0 ' = ' 1 '.]

TABLE VI : ' IPt4 ' 128 Bit-Mapped Space

"The Logically derived Structure of the 'Synthetic' Decimal Representation of the IPt4 IP Addressing Specification"

IPt4 Address Pool Size = 2^{32} (256^{12}) = 256^{16}

= 4,294,967,296(256^{12}) = 3.4028237 x 10^{38}

NOTE: IPt4, by FIGURE '5" contains;

'7.9228163 x 10^{28} ' COPIES of the 'IPt1' Specification

CLASS A - 'Address Range 1 - 128'

A-1; Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.Y.X.X ±/0000:08
- **Networks: 128 x (128 x 256^{2})(256^{12})**
- **Host: 128 x 256^{2} = 8,388,608**

A-2; Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.Y.X ±/0000:16
- **Networks:(128^{2})(128 x 256)(256^{12})**
- **Host: 128 x 256 = 32, 768**

A-3; Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.ØØØ.Y ±/0000:24
- **Networks:(128^{3}) x 128(256^{12})**
- **Host: 128 = 128**

A-4; Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.ØØØ.ØØØ ±/0000:32
- **Networks: 128^{4} (256^{12})**
- **Host: 0**

Total Number of Available Network and Host Addresses:

Networks ID - 2^{7} = (256^ 12)(128 x 16,777,216)

= 1.7014118 x 10^{38}

Hosts = (128 x 256^{2}) + (128 x 256) + 128 = 8,421,504

TABLE VI : ' IPt4 ' 128 Bit-Mapped Space - Continued

CLASS B - 'Address Range 129 - 192'

B-1, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.Y.X.X ±/1000:08
- **Networks: 64(256 - 64)(256^{2})(256^{12})**
- **Host: 64 x 256^{2} = 4,194,304**

B-2, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.Y.X ±/1000:16
- **Networks: (64^{2})(256 - 64) x 256(256^{12})**
- **Host: 64 x 256 = 16, 384**

B-3, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.ØØØ.Y ±/1000:24
- **Networks: 64^{4} (256^{12})**
- **Host: 64 = 64**

B-4, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.ØØØ.ØØØ ±/1000:32
- **Networks: (256^{12})**
- **Host: 0**

Total Number of Available Network and Host Addresses:

Networks ID - 2^{6} = (256^{12})(64 x 16,777,216)

= 8.5070592 x 10^{37}

Hosts = (64 x 256^{2}) + (64 x 256) + 64 = 4,210,752

TABLE VI : ' IPt4 ' 128 Bit-Mapped Space - Continued

CLASS C - 'Address Range 193 - 224'

C-1, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.Y.X.X ±/1100:08
- Networks: $32(256 - 32)(256^{2})(256^{12})$
- Host: $32 \times 256^{2} = 2{,}097{,}152$

C-2, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.Y.X ± /1100:16
- Networks: $32^{2}(256 - 32) \times 256(256^{12})$
- Host: $32 \times 256 = 5{,}888$

C-3, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.ØØØ.Y ± /1100:24
- Networks: $32^{3}(256 - 32)(256^{12})$
- Host: 32 = 32

C-4, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.ØØØ.ØØØ ± /1100:32
- Networks: $32^{4}(256^{12})$
- Host: 0

Total Number of Available Network and Host Addresses:
Networks ID - $2^{5} = (256^{12})(32 \times 16{,}777{,}216)$
$= 4.2535296 \times 10^{37}$
Hosts $= (32 \times 256^{2}) + (32 \times 256) + 32 = 2{,}105{,}376$

TABLE VI : ' IPt4 ' 128 Bit-Mapped Space - Continued

CLASS D - 'Address Range 225 - 240'

D-1, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.Y.X.X ±/1110: 08
- Networks: $16(256 - 16)(256^{2})(256^{12})$
- Host: $16 \times 256^{2} = 1{,}048{,}576$

D-2, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.Y.X ± /1110:16
- Networks: $16^{2}(256 - 16) \times 256(256^{12})$
- Host: $16 \times 256 = 4{,}096$

D-3, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.ØØØ.Y ± /1110:24
- Networks: $16^{3}(256 - 16)(256^{12})$
- Host: 16 = 16

D-4, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:ØØØ.ØØØ.ØØØ.ØØØ ± /1110:32
- Networks: $16^{4}(256^{12})$
- Host: 0

Total Number of Available Network and Host Addresses:
Networks ID - $2^{4} = (256^{12})(16 \times 16{,}777{,}216)$
$= 2.1267648 \times 10^{37}$
Hosts $= (16 \times 256^{2}) + (16 \times 256) + 16 = 1{,}052{,}688$

TABLE VI : ' IPt4 ' 128 Bit-Mapped Space - Continued

CLASS E - 'Address Range 241 - 256'

E-1, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:000.Y.X.X ±/1111:08
- Networks: $16(256 - 16)(256^{2})(256^{12})$
- Host: $16 \times 256^{2} = 1{,}048{,}576$

E-2, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:000.000.Y.X ± /1111:16
- Networks: $16^{2}\ (256 - 16) \times 256(256^{12})$
- Host: $16 \times 256 = 4{,}096$

E-3, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:000.000.000.Y ± /1111:24
- Networks: $16^{3}\ (256 - 16)(256^{12})$
- Host: 16 = 16

E-4, Subnet Id - 256:256:256:256:256:256:256:256:256:256:256:256:000.000.000.000 ±/1111:32
- Networks: $16^{4}\ (256^{12})$
- Host: 0

Total Number of Available Network and Host Addresses:

Networks ID - $2^{4} = (256^{12})(16 \times 16{,}777{,}216)$

$= 2.1267648 \times 10^{37}$

Hosts $= (16 \times 256^{2}) + (16 \times 256) + 16 = 1{,}052{,}688$

Note: Since, the "Subnet Id" = the "Subnet Identifier", the Shared Host IP Address Pool increases by a factor equal the number of available Network IP addresses assigned to every sub-division within the Overlay defining the Address Class Range, given that No Host Address is assigned a direct Internet Connection.

eq-1: IPtX $= X(2^{32}) + 16{,}900{,}000 = \infty$

eq-2: IPv6 $= X(2^{32}) = (2^{128}) = (256^{16}) = 3.4028237 \times 10^{38}$

eq-3: IPt4 $= X(2^{32}) + 16{,}900{,}000$
$= (2^{128}) + 16{,}900{,}000$
$= (256^{16}) + 16{,}900{,}000$
$= 3.4028237 \times 10^{38} + 16{,}900{,}000$ Host Addresses

eq-4: IPv4 $= X(2^{32})$; when X = 1.

FIGURE 5 : 'IPt1'

"The Logically derived Structure of the 'Synthetic' Decimal Representation of the IPt1 IP Addressing Specification"

'IPt1' Address Pool Size = 2^{32} = 256^{4}

= 4,294,967,296 = 4.294967296 x 10^{9}

CLASS A

1. A-1, 1 - 128, Subnet Identifier ØØØ.Y.X.X -/0000:08
   - Networks: 128 x (128 x 256^{2})
   - Host: 128 x 256^{2}

A-2, 1 - 128, Subnet Identifier ØØØ.ØØØ.Y.X -/0000:16
- Networks: (128^{2})(128 x 256)
- Host: 128 x 256

A-3, 1 - 128, Subnet Identifier ØØØ.ØØØ.ØØØ.Y -/0000:24
- Networks: (128^{3}) x 128
- Host: 128^{4}

A-4, 1 - 128, Subnet Identifier ØØØ.ØØØ.ØØØ.ØØØ -/0000:32
- Networks: 128
- Host: 0

Total Number of Available Network and Host Addresses:

2^{7} Networks = 128 x 16,777,216 = 2,147,483,648

Hosts = (128 x 256^{2}) + (128 x 256) + 128 = 8,421,504

FIGURE 5 : 'IPt1' - Continued

CLASS B

2. B-1, 129 - 192, Subnet Identifier ØØØ.Y.X.X -/1000:08
   - Networks: $64(256 - 64)(256^{2})$
   - Host: 64×256^{2}

   B-2, 129 - 192, Subnet Identifier ØØØ.ØØØ.Y.X -/1000:16
   - Networks: $(64^{2})(256 - 64) \times 256$
   - Host: 64×256

   B-3, 129 - 192, Subnet Identifier ØØØ.ØØØ.ØØØ.Y -/1000:24
   - Networks: $64^{3}(256 - 64)$
   - Host: 64^{4}

   B-4, 129 - 192, Subnet Identifier ØØØ.ØØØ.ØØØ.ØØØ -/1000:32
   - Networks: 64
   - Host: 0

Total Number of Available Network and Host Addresses:

2^{6} Networks = $64 \times 16{,}777{,}216 = 1{,}073{,}741{,}824$

Hosts = $(64 \times 256^{2}) + (64 \times 256) + 64 = 4{,}210{,}752$

FIGURE 5 : 'IPt1' - Continued

CLASS C

3. C-1, 193 - 224, Subnet Identifier ØØØ.Y.X.X -/1100:08
   - Networks: $32(256 - 32)(256^{2})$
   - Host: 32×256^{2}

   C-2, 193 - 224, Subnet Identifier ØØØ.ØØØ.Y.X -/1100:16
   - Networks: $32^{2}(256 - 32) \times 256$
   - Host: 32 x 256

   C-3, 193 - 224, Subnet Identifier ØØØ.ØØØ.ØØØ.Y -/1100:24
   - Networks: $32^{3}(256 - 32)$
   - Host: 32

   C-4, 193 - 224, Subnet Identifier ØØØ.ØØØ.ØØØ.ØØØ -/1100:32
   - Networks: 32^{4}
   - Host: 0

Total Number of Available Network and Host Addresses:

2^{5} Networks = 32 x 16,777,216 = 536,870,912

Hosts = $(32 \times 256^{2}) + (32 \times 256) + 32$ = 2,105,376

FIGURE 5 : 'IPt1' – Continued

CLASS D

4. D-1, 225 - 240, Subnet Identifier ØØØ.Y.X.X -/1110:08
   - Networks: $16(256 - 16)(256^{2})$
   - Host: 16×256^{2}

   D-2, 225 - 240, Subnet Identifier ØØØ.ØØØ.Y.X -/1110:16
   - Networks: $16^{2}(256 - 16) \times 256$
   - Host: 16 x 256

   D-3, 225 - 240, Subnet Identifier ØØØ.ØØØ.ØØØ.Y -/1110:24
   - Networks: $16^{3}(256 - 16)$
   - Host: 16

   D-4, 225 - 240, Subnet Identifier ØØØ.ØØØ.ØØØ.ØØØ -/1110:32
   - Networks: 16^{4}
   - Host: 0

Total Number of Available Network and Host Addresses:

2^{4} Networks = 16 x 16,777,216 = 268,435,456

Hosts = $(16 \times 256^{2}) + (16 \times 256) + 16$ = 1,052,688

FIGURE 5 : 'IPt1' - Continued

CLASS E

5. E-1, 241 - 256, Subnet Identifier ØØØ.Y.X.X -/1111:08
   - Networks: $16(256 - 16)(256^{2})$
   - Host: 16×256^{2}

   E-2, 241 - 256, Subnet Identifier ØØØ.ØØØ.Y.X -/1111:16
   - Networks: $16^{2}(256 - 16) \times 256$
   - Host: 16×256

   E-3, 241 - 256, Subnet Identifier ØØØ.ØØØ.ØØØ.Y -/1111:24
   - Networks: $16^{3}(256 - 16)$
   - Host: 16

   E-4, 241 - 256, Subnet Identifier ØØØ.ØØØ.ØØØ.ØØØ -/1111:32
   - Networks: 16^{4}
   - Host: 0

Total Number of Available Network and Host Addresses:

2^{4} Networks = 16 x 16 ,777,216 = 268,435,456

Hosts = $(16 \times 256^{2}) + (16 \times 256) + 16 = 1,052,688$

Special Note: The simplification of the Network IP Addressing format into the 'Zone IP', the 'IP Area Code', and the 'Network IP Address', as provided by the IPtX Specification, could also REPLACE the format currently being used by the Telephone Systems all over the World. In other words, there is an EASY, 'Off-The-Shelf' procedure for DIALING any Telephone Number defined by the 18 Digit String from the 'IPtX' Specification:

1. Dial First - Key in the digits representing the 'ZONE IP'

2. Dial Next - Key in an Asterisk " * " the 'End-Start Statement'(:)'

3. Dial Second - Key in the digits representing the 'IP AREA CODE'

4. Dial Next - Key in an Asterisk " * " the 'End-Start Statement'(:)'

5. Dial Third - Key in the All '12' digits of the 'NETWORK IP ADDRESS'

eg. Direct Trunk-Identifier, and respective Local "ENUM" Dialing:

a. Dial Zone IP: '123:213:121.0.12.3' =
'123 * 213 * 121-000-012-003';

b. Dial IP Area Code: '123:213:121.0.12.3' =
' *213 * 121-000-012-003';

c. Dial Local '123:213:121.0.12.3' = ' ** ' 121-000-012-003';

Where it is not possible to define the Zone IP,
IP Area Code, or Network IP Address with ' 0 '.

{See "TELe-RIP" Protocol}

FIGURE 6 : 'IPt2'

"The Logically derived Structure of the 'Synthetic' Decimal Representation of the IPt2 IP Addressing Specification"

'IPt2' Address Pool Size = 2^{32} (256^{2}) = 256^{6}
= $4,294,967,296(256^{2})$ = 2.814798×10^{14}

NOTE: IPt2, by FIGURE '5' contains;
' 256^{2} ' COPIES of the 'IPt1' Specification

CLASS A

1. A-1, 1 - 128, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.Y.X.X ±/0000:08
- Networks: $128 \times (128 \times 256^{2})(256^{2})$
- Host: 128×256^{2}

A-2, 1 - 128, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.Y.X ±/0000:16
- Networks: $(128^{2})(128 \times 256)(256^{2})$
- Host: 128 x 256

A-3, 1 - 128, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.ØØØ.Y ±/0000:24
- Networks: $(128^{3}) \times 128(256^{2})$
- Host: 128

A-4, 1 - 128, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.ØØØ.ØØØ ±/0000:32
- Networks: 128^{4} (256^{2})
- Host: 0

Total Number of Available Network and Host Addresses:
Networks 2^{7} $(256^{3})(256^{2})$ = $(128 \times 16,777,216)(256^{2})$
= $2,147,483,648(256^{2})$
Hosts = (128×256^{2}) + (128 x 256) + 128 = 8,421,504

FIGURE 6 : 'IPt2' - Continued

CLASS B:

2. B-1, 129 - 192, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.Y.X.X ±/1000:08
- Networks: 64(256 - 64)(256^2)(256^2)
- Host: 64 x 256^2

B-2, 129 - 192, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.Y.X ±/1000:16
- Networks: (64^2)(256 - 64) x 256(256^2)
- Host: 64 x 256

B-3, 129 - 192, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.ØØØ.Y ±/1000:24
- Networks: 64^3 (256 - 64)(256^2)
- Host: 64

B-4, 129 - 192, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.ØØØ.ØØØ ±/1000:32
- Networks: 64^4 (256^2)
- Host: 0

Total Number of Available Network and Host Addresses:
Networks 2^6 (256^3)(256^2) = (64 x 16,777,216)(256^2)
= 1,073,741,824(256^2)
Hosts = (64 x 256^2) + (64 x 256) + 64 = 4,210,752

FIGURE 6 : 'IPt2' - Continued

CLASS C

3. C-1, 193 - 224, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.Y.X.X ± /1100:08
- Networks: $32(256 - 32)(256^{2})(256^{2})$
- Host: 32×256^{2}

C-2, 193 - 224, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.Y.X ± /1100:16
- Networks: $32^{2}(256 - 32) \times 256(256^{2})$
- Host: 32 x 256

C-3, 193 - 224, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.ØØØ.Y ± /1100:24
- Networks: $32^{3}(256 - 32)(256^{2})$
- Host: 32

C-4, 193 - 224, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.ØØØ.ØØØ ± /1100:32
- Networks: $32^{4}(256^{2})$
- Host: 0

Total Number of Available Network and Host Addresses:
Networks $2^{5}(256^{3})(256^{2}) = (32 \times 16{,}777{,}216)(256^{2})$
$= 536{,}870{,}912(256^{2})$
Hosts $= (32 \times 256^{2}) + (32 \times 256) + 32 = 2{,}105{,}376$

FIGURE 6 : 'IPt2' - Continued

CLASS D

4. D-1, 225 - 240, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.Y.X.X ± /1110:08
- Networks: $16(256 - 16)(256^{2})(256^{2})$
- Host: 16×256^{2}

D-2, 225 - 240, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.Y.X ± /1110:16
- Networks: $16^{2}(256 - 16) \times 256(256^{2})$
- Host: 16×256

D-3, 225 - 240, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.ØØØ.Y ± /1110:24
- Networks: $16^{3}(256 - 16)(256^{2})$
- Host: 16

D-4, 225 - 240, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
ØØØ.ØØØ.ØØØ.ØØØ ± /1110:32
- Networks: $16^{4}(256^{2})$
- Host: 0

Total Number of Available Network and Host Addresses:
Networks $2^{4}(256^{3})(256^{2}) = (16 \times 16{,}777{,}216)(256^{2})$
$= 268{,}435{,}456(256^{2})$
Hosts = $(16 \times 256^{2}) + (16 \times 256) + 16 = 1{,}052{,}688$

FIGURE 6 : 'IPt2' - Continued

CLASS E

5. E-1, 241 - 256, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
   ØØØ.Y.X.X ±/1111:08
   - Networks: $16(256 - 16)(256^{2})(256^{2})$
   - Host: 16×256^{2}

   E-2, 241 - 256, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
   ØØØ.ØØØ.Y.X ±/1111:16
   - Networks: $16^{2}(256 - 16) \times 256(256^{2})$
   - Host: 16×256

   E-3, 241 - 256, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
   ØØØ.ØØØ.ØØØ.Y ±/1111:24
   - Networks: $16^{3}(256 - 16)(256^{2})$
   - Host: 16

   E-4, 241 - 256, Subnet Id - 8 Bit Reserved:8 Bit Reserved:256:256:
   ØØØ.ØØØ.ØØØ.ØØØ ±/1111:32
   - Networks: $16^{4}(256^{2})$
   - Host: 0

Total Number of Available Network and Host Addresses:
Networks $2^{4}(256^{3})(256^{2}) = (16 \times 16{,}777{,}216)(256^{2})$
$= 268{,}435{,}456(256^{2})$
Hosts = $(16 \times 256^{2}) + (16 \times 256) + 16 = 1{,}052{,}688$

Special Note:

In the IPtX Specification any Contiguous String of ZERO(s) is not a mathematically viable IP Address, because the IP Address Pool equation, $X(X^{32})$, defines a sequence of counting by successive additions of '1'. That is, using Zero(s), while not a problem for the IPtX Specification, it is however, a problem for IPv6. In other words, the problems inherent in the IPv6 Specification pertain primarily to the difficulties arising from the 48 Digit Number used to represent the Default format of an IP Address. And while there are Rules which define the use of " :: " as the replacement for the Contiguous ZERO(s) String, because they are not mathematically viable, the problem is exacerbated. Furthermore, since I have not found any documentation that specifically defines how to distinguish between 2 or more Addresses defining a different number of ZERO(s) in the Contiguous String, which might occur in the Start, Middle, or Trailing portion of the Address. Clearly, the " :: " cannot be used as the solution for the size reduction of an IPv6 Address, the replacement for a Contiguous String of ZERO(s), at least, not without the necessary Rule(s) explaining its use. Nevertheless, these inadequacies are further compounded by defining the IP Addressing Format as having a variable length, which may be greater than 128 Bits, and the " :: " as a compressor of the leading and/or trailing zeros, or as the replacement for the multiple 16-bit Zero Groups with a specification that limits its use, number of occurrences, in an address. And this, needless to say, defines the gist of the mathematical failings of every supporting document, including RFC 2373, and more importantly, the reasons the Chinese developed the IPv9 Addressing Specification.

It should have been quite clear from documents expounding the development of the design for the 'IPtX' Specification that the "Network Descriptor" provided the means to identify the "Trunk-Identifier", which defines the IP Address assigned to the 'TelCo-Xchge or Backbone Routers'. In other words, from 'Table V - Definitions', the "Network Descriptor" allows for the resolution of the Trunk-Identifier ID and the resolution of the entire Length of the Network IP Address String. And while the "CIDR" notation was exploited even further during the creation of the IPv6 Specification; because the symbol, ' /128 ', specifies the length for the Bit-Mapped IP Address, which defines the size of the 128 Bit-Mapped IPv6 Address Space. The "Network Descriptor" provided the ability to separate, or distinguish between the addresses assigned to the 'TelCo-Xchge or Backbone Routers' and the 'Network IP Address'. And this specified the Length defining the Network IP Address as '32 Bits' in the IPtX Specification, even if the size of its Address Space was equal to or greater than the 128 Bit IPv6 Specification.

Nevertheless, this procedure allows the retention of the more familiar structure defining a Network IP Address, which is similar to the 'Telephone Number' and defined by the IPv4 Specification. And clearly, this mathematically clarifies the interpretation of the IPtX Specification (See Table VII), which is logically derived from IPv4, by proving that the number of available Addresses in its IP Address Pool is greater than the IPv6 Specification. Furthermore, since it was only in the configuration of the Router where "CIDR" was addressed, "CIDR", as with the "Network Descriptor" can be automatic, and remain only as the concerns for the Network Engineer, because its affect lies outside the boundary of the Network Domain. In other words, IPtX is a more powerful and cost effective IP Addressing Specification, which allows the interface of the "Front-End" to mimic or simulate the IPv4 Specification, a 32 Bit-Mapped IP Address, in the Backbone environment of an unlimited size IP Address Space.

Table VII

IPtX - "Network IP Address" Bit-Mapped Length Specification = 32 Bits

IPt2 - "Trunk-Identifier" = 8 Bit Rsvd:8 Bit Rsvd:256:256:
= 16 Bit-Mapped Address String
"Network Descriptor" =+/ 0000:16; Range is '00 thru 16'

IPt4 - "Trunk-Identifier" = 256:256:256:256:256:256:256:256:256:256:256:256:
= 96 Bit-Mapped Address String
"Network Descriptor" =+/ 0000:96; Range is '00 thru 96'

Given that the first '0000' defines the '8 Bit' String that identifies:

"Trunk-Identifier" : Class Range - A, B, C, D, E - using the '+/ ' Switch
"Network ID" : Address Class of Network IP Address - A, B, C, D, E - using the ' -/ ' Switch

5. Security Considerations

This document, whose only objective was the deliberation of the final explanation for the development of the IPtX Specification, which resulted from the Mathematics of Quantification, does not directly raise any security issues. Hence, there are no issues that warrant Security Considerations.

6. IANA Considerations

The complete Anatomy of a IPtX Address String**

1. **'Router-ID + Trunk-Identifier + Network IP Address**':**

Total Address Length - IANA Specifications

2. **'TelCo-Xchge or Backbone Routers' - Router-ID: 'DNS ZONE'**

" ID + Trunk-Identifier + ZONE IP + IP AREA CODE "

Total Segment Address Length - IANA Specifications

`CIDR Network Descriptor = /0000:00`

3. **'Trunk-Identifier' - ID: 'DNS ZONE - Address Block Specification'**

" ID + ZONE IP + IP AREA CODE "

a. **- 'ZONE IP' - ID ' - Segment Address Length - IANA Specifications**

b. **- 'IP AREA CODE - ID ' - Segment Address Length - IANA Specifications**

Total Segment Address Length - IANA Specifications

`CIDR Network Descriptor = +/0000:00`

4. **'Network IP Address' - ID: Segment Address Length - IANA Specifications**

`CIDR Network Descriptor = -/0000:00`

6.1 Special IANA Consideration:

While the use of Router Authentication can include Address String verification, its application may prove even more vital for Security applications. However, Address resolution could be accomplished more easily using a '3 State CIDR Network Descriptor Switch' {0, -1, +1}:

a) **Router-ID: /0000:00 or /0000:00E00; ' 00E 00' = $00^{00 \text{ thru } 99}$**

{e.g. 96E99 = 96^{99} Bits ; or , $X^{99}(2^{32}) \approx (7.9228163 \times 10^{28})^{99}$ copies of 'IPt1' }

b) **Trunk-Identifier: +/0000:00 or +/0000:00E00; ' 00E 00' = $00^{00 \text{ thru } 99}$**

{e.g. 32E10 = 32^{10} Bits; or, $X^{10}(2^{32} \approx (4{,}294{,}967{,}296)^{10}$ copies of 'IPt1' }

c) **Network IP Address: -/0000:00 or -/0000:00E00; ' 00E 00' = $00^{00 \text{ thru } 99}$**

d) **'ANSI Trace – Ping Switch': Multi-Conditional Switch defining Additional Command Statements – ±//0000: |? and //0000:00|? {Where '|?' = HELP!}**

The above considerations represent the conclusions derived from the foundation of 'RFC(s) 1518 and 1519', which actually define 'Address Segment Routing'. Given that in the Binary System of counting, sums by the addition of '1's' are from the Right, and leading Zeros to the Left are insignificant. The conclusions clearly establish the above mathematical expressions as a viable representation of the Router's interpretation of an IPtX 'Bit-Mapped' IP Address. That is, the 'CIDR Network Descriptor' provides the Router's depiction of an IP Address, which only acknowledges the routing of the Network-ID according to the structure of the Network's IP Address defined by the Address Class Range: '±/0000:00E00 and /0000:00E00'. Hence, this is the gist of an extremely powerful method, 'Address Segment Routing', which provides the translation of the Network Address defined by the Schematic representing the IPtX IP Addressing Specification, and any length of one or more additional Address Segments defining the total length of the IPtX Bit-Mapped Address. In other words, the 'CIDR Network Descriptor', in addition to defining the mathematical expression of the routed Address, it also acts as the Address Mask, resolving the Integer, one Bit at a time, in a process that defines every Address defined in the Schematic of the IPtX Addressing Specification. And finally, these conclusions also confirm the necessity of resolving the 48 Bit-Mapped IP Address (18 Digits), as the required identification to specifically determine the exact location/destination of the communicating nodes.

In other words, the 'CIDR Network Descriptor', in addition to defining the mathematical expression of the routed Address, it also acts as the Address Mask, resolving the Integer, one Bit at a time, in a process that defines every Address defined by the Schematic of the IPtX Addressing Specification. And finally, these conclusions also confirm the necessity of the 48 Bit-Mapped IP Address (18 digits), as the required identification to specifically determine the exact location/destination of the communicating nodes.

Nevertheless, the method of Counting that is defined by '±/0000:00E00 and 000:00E00', defines a "One-to-One' Correspondence with the Unary Element, ' 1 ', which defines the Sum of the Elements contained in the Set defining the 'CIDR Network Descriptor, as the Cardinality representing the count of the number of elements the Set Contains. Thus, yielding the exact depiction, from the Sum of the number ' 1's ', this equals the Number representing the IPtX IP Address. In other words, the Sum of the Bit Count defined by the '00E00' String is equal to the Number representing the remaining portion of the IP Address that is not defined by the Address Class Range, which defines the 8 Bit '0000' Prefix of the IP Address defined by the Network Descriptor. However, allowing the '00E00.0000' String to be equal to the method defining 'Exponential Enumeration' would be easier, because an Exponential Equation could Mask and Un-Mask the Integer representing the IP Address(s); '±/0000:00E00.0000 and /0000:00E00.0000'. And this method would also represent an exact count of the Sum of the ' 1's ', because it equals the one-to-one relationship that assigns only one Number from the result of, '2^X', the 'Exponent', to one IP Address. Furthermore, it is also behooving to note, that since (±/)'0000:00E00.0000' and (/)'0000:00E00.0000' is Routable, its Function as the 'Network Descriptor' still remains useful, because it allows for Address Segment Routing, and retains the original definition of its Address String Length; '±/0000:00'. In other words, the expansion of the 'CIDR Network Descriptor' defines a Routable Mathematical Expression, defining a '2^X : 1' 'Exponential Compression Ratio', which can be used in either a Static or Dynamic Mode.

Note: The logical expansion of the 'CIDR Network Descriptor' defines a Routable 'Universal Expression', which is a Mathematical Expression defined by an Exponential Equation', expressing the value of the result that uniquely represents the value of the Quantified Sum of the Component Strings defining the numerical value of every IP Address defined by the IPtX Specification. And this, as it will be clearly realized, changes the calculation for the Bit/Data Transfer Rate.

In any case, it should by quite clear that the foregoing uniquely defines the IPtX Specification, a distinct difference from IPv4, that provides the exact rendering of Binary Enumeration, which is equal to the sequential incrementation resulting from the summation of the progressing using ' 1's '. In other words, since the Binary equivalent of the Unit displacement, given by eq-5, defines the 'Exponential Expression' derived from the Binary Equation defined by the Mathematics of Quantification, which defines an incremental progression using the result from the summation of ' 1's '. It defines the result, 2^Q, from the equation as representing the Quantified Sum of the Binary Expressions defining the Numerical Displacement of the 8 Bit Segment that defines the Octet represented in an IP Address {see eq-6 and eq-7}.

eq-5: 2^X = 1; respectively denoting the Binary and Unary Sets

eq-6: 2^X = 8 Bit String = " 1 thru 256 "

eq-7: $(2^X) + (2^X) + (2^X) + (2^X) = 2^Q$, given that { • } = { + }, then
8 Bit. 8 Bit. 8 Bit. 8 Bit = 32 Bit = $2^Q = 2^N$,
XXX + XXX + XXX + XXX = 2^Q = 00E000.0000...,
2^Q now defines an incremental progression, using the summation from the additions of ' 1's ', which approaches a Bit displacement defining an infinite length; see eq-9.

And clearly, given that 2^Q = 00E000.0000..., where 2 = 00 and Q = E000.0000..., the Masking and Un-Masking procedure for protocol encapsulation is extremely elementary, because it is a comparison defined by the laws of addition and subtraction. In other words, the Sum of any number of the Binary Expressions equaling the value of the 8 Bit Address Segment defining an Octet of an IP Address, {where the 00E00.0000 = IPtX IP Address as the Sum of the Binary Expressions defining the Total Number of Octets the IP Address contains.) is equal to the value of the Integer defining an IPtX IP Address. Hence, if there is a difference between any of the values defining an Octet or their resulting Sum, 2^Q, it would be determined, confirmed, or verified by the Translation / Resolution of an IP Address from the Masking or Un-Masking of the 'Exponential Expression(s)'. And this uniquely defines, by eq-8, the 'IPtX Mathematical Expression' Routing Protocol, and the ability of IPtX Specification to define an Un-Limited IP Address Pool: see 'The Anatomy of the IPtX-MX Routing Protocol'.

eq-8: 0000:00E000.0000... ; $2^X + 2^X + 2^F + 2^F = 2^Q$
2^Q = IP Address value = the Sum of the Octets.

The Anatomy of the IPtX-MX Routing Protocol

The Number of 8 Bit Address Segments - IANA Specifications

0000:00E000.0000... = $2^{999.99987654321}$ IP Addresses = 2^{Q}
or: Theoretical Number of available IP Addresses =
$2^{2(999.99987654321)}$ copies of IPt1 = [('/0000:00') x ('+/0000:00')]

1\. ' 0000: ' ID: 8 Bit Address Segment - Address Class Range

Total Segment IP Address Length - IANA Specifications

2\. ' 00 ' ID: 4 Bit Address Segment - The Base 2, in 2^{Q}

Total Segment Address Length - IANA Specifications

3\. ' E ' ID: 4 Bit Address Segment - Binary Exponential Operator

Total Segment Address Length - IANA Specifications

4\. ' 000.0000...' ID: 48 Bit Address Segment - 2 Part Exponent Q

a.' 000 ' ID: 3-4 Bit Address Segment - Integer Part of Exponent

Segment Length - IANA Specifications

b.'.0000...' ID: 36 Bit Address Segment - Decimal Part of Exponent

The Anatomy of the IPtX-MX Routing Protocol

5. 'Router-ID + Trunk-Identifier + Network IP Address*

0000:00E000.0000... + 0000:00E000.0000... + 0000:00E000.0000...

3(8 + 20 + 36) = 3(64) Bit Strings

a. 0000:00E000.0000... /0000:00 - Masked Routable Router-ID

b. 0000:00E000.0000... +/0000:00 - Masked Routable Trunk-Identifier

c. 0000:00E000.0000... -/0000:00 - Masked Routable Network IP Address

Total Address Length - IANA Specifications

eq-9: 0000:00E000.0000... ; The Reality of "2^{X} : 1"
Compression Ratio of the IPtX-MX Protocol

a. $(2^{X})_1 + (2^{X})_2 + (2^{F})_3 + (2^{X})_4 = 2^{Q}$

b. $(2^{X})_1 + (2^{X})_2 + (2^{F})_3 + (2^{X})_4 + (2^{F})_5 + (2^{F})_6 = 2^{Q}$

c. $(2^{X})_1 + (2^{X})_2 + \ldots + (2^{F})_N = 2^{Q}$

Given that 2^{X} defines Binary Enumeration, as the result from an Exponential Equation that defines the Unary incremental progression of addition, or counting, using ' 1's '.

Note: The comparable analogy the Computing Power defining the payload capacity of the '00.E000.0000... String' is equivalent to having $2^{999.99987654321}$ 3-State Logical Transistor Switches in the Core of '1' CPU. And this, it should be duly noted, is equivalent to the ability of programming every Transistor, or allowing every Transistor to become the CPU. However, since every 4 Bit segment can equal any number in the range of '0 thru 16', the '00.E000.0000... String' Theoretical Number of available IP Addresses now equals:

$$[('/0000:00') \text{ x } ('+/0000:00')] = (16^{161,616})^2 = [(2^4)^{161,616}]^2 =$$

$$2^{2(646,464)} = 2^X = 2^{1,292,928.99987654321} \text{ copies of IPt1}$$

In which case, this means; A 64 Bit Address String can define the Number or Data Stream that represents the Sum of the incremental progressions, using '1's ', which equals the count of a $2^{646,464}$ Bit-Mapped displacement. And if the Address total defined by IPt1; 4,294,967,296, equaled a book containing 67,108,864 64 Bit words, then the '00.E000.0000... ' 64 Bit Address Block defines a Bit-Mapped Address String equaling approximately $2^{646,464}$ Books; when a book contains approximately 50,000 pages.

[The Genetic Equation representing the Formula for Life? ...Yep.]
Encryption and Decryption - The Binary Enumeration Algorithm - 2^N

1) $X(2^X)$ - Encryption and Decryption Key

a. X - Counts the Number of Octets - 00 thru 2^X
b. (2^8) - Encodes and Decodes Bit-Map Octet String

2) 00.E000.0000... - Binary Enmeration Compression Algorithm

a. Encryption of the Integer defines a Unique Bit-Map Length -
e.g. - 123.123.123.123 = 1111011.1111011.1111011.1111011 = 2^Q
b. Every Integer has a Unique '00.E000.0000...' Translation

3) ' + ' and the ' . ' Symbols: "Put it Together & Take it Away"

a. String Assembler - ' + ' "Put it Together" -
e.g. - '123 + 123' = '123.123' = '123123' = 2^Q
b. String Delimiter - ' . ' "Take it Away" -
e.g. - '123123' = '123.123' = '123 + 123' = 2^Q

In other words, Binary Enumeration, the method of incrementation using ' 1 ' that defines the ' 2^X : 1 ' Ratio, sustains a practical limit, which is defined by the current technology. However, with the current technology it might be easier if the Exponent of the ' 2^X : 1 ' Ratio defined a Payload Capacity of 256 Bits, or the equation, 2^{256}, which defines a Payload capacity equaling 256 copies of a 50,000 page Book. And this, it should be reasoned, with the appropriate upgrades would allow the Ratio defining the current Data Transmission Rates, and a host of related technological products, to obtain a performance increase by that might be greater than a ' 4 : 1 ' Ratio.

6.2 Special IANA Consideration

- 6.2.1 - Closing Argument

The proof of the mathematical validity for the Ratio defining the ' $2^X : 1$ ' relationship between Binary and Unary counting is given by:

- **Since; $2^0 = 1$, and $(2^0) + (2^0) = 1 + 1$ is True,**
- **then; If $1 + 1 = 2$, $1 + 2 = 3$, ..., and $1 + N = I_N$, is True,**
- **there is a relationship; defining every 'X', in '2^X',**
- **such that; for every value that the element 'Q' defines, also defines a unique element in I, the Set of Integers,**
- **then; '2^X' also defines an element of the Set of Integers,**
- **given that; for every '2^X' of the Set I, '$2^X = Q$',**
- **and; If 'X' defines an Element of the Set of Integers and an Element of the Set of Real Number,**
- **then; 'X' is Greater Than, or Equal To 'N_{N+1}' and '$N_N + 1.000..._{N \vee N+1}$'.**
- **Hence: The Binary Set defines an Equation, '2^X, that equals the count defining the Sum of the incremental progression resulting from the Addition of '1's', which is equal to the progressive counting sequence defined by the Unary Set.**

This concludes the design specification(s) for the IPtX Addressing Protocol Family, with the implementation of the logically derived 'Intelligent Quantum Worm', which mimics the 'Payload Carrying Capacity' of the 'Instruction Set' carried by a Cell of DNA. In other words, this discovery represents the capability of compressing

a number of any length (Compression of the Set, Q = {1,2,3,4,...,N}, having a count of 'N' Members in it's Set, into Q = {1,2,3,4,...,N} = 1, by Nesting Encryptions of the Exponential Translation representing the exact Numerical arrangement of the members of the Set, Q. And more importantly, this discovery, when the correct Mathematical interpretation of the encoding for the Binary System is used, defines the 64 Bit Address as the Mask of the 'Intelligent Quantum Worm', the 'IPtX-MX Universal Routing/Routed Protocol' Specification, which has a 'Theoretical Payload Carrying Capacity' (Data Stream 'Bit-Count' Capacity) equal to the Bit size specification defining a Backbone Environment having an unlimited size 'Bit-Mapped' Address Space.

Polymorphing the 'Intelligent Quantum Worm' Protocol

"The Biological Depiction of the 3 Divisions of a Cell and the Exponential Compression Algorithm is the Nucleus"

Calibrating the Unit Bit Size to the measurement of the Displacement Frequency Characterizing any one or more Unit measures from the Frequencies of the 'Electro-Magnetic Spectrum of the Nucleus of an Atom.

The Polymorphing "Metamorphosis'" Encryption Compression Algorithm

1. ' 0000:00E000.0000... ' ;

"Metamorphosis" Encryption Compression Algorithm;

2. ' 0000:00.0 000...E000 '

And this can be programmed to imitate Virus or Bacteria Prorogation, which can be Transformed, or adapt..., because this Algorithm provides the Binary representation, translation, or interpretation of every Element contained in the Set of Real Numbers, including the Rational and the Irrational numbers.

Which means, at some Point:

3. 0000:00.00 00...E000 = 0000:00E000.0000...

In other words, the Binary Equation in the Ratio, '2^X : 1', actually defines the Unary Process of successive Additions of ' 1 ', which the counting by 1, represents every Numeral. However, the benefit of the Exponential Algorithm is that, regardless of the Numeral's length, it requires only 64 Bits to represent the count of the Integer depicting the Numeral.

Nevertheless, the pointed fact is that, the mathematical translation of the foregoing conclusions in terms of the number of available IP Addresses, dramatically increases. In other words, the IP Address Pool representing the 64 Bit IP Address of the IPt2 Address Space, which is defined by the IPtX Specification, more closely approximates:

16 Prefix Addresses

99,987,654,321 Addresses in each Prefix Address Set

And this means, there are;

$2^{646,464}(16 \text{ x } 99,987,654,321)$ Encryption Keys, or;

because this defines the Number of Encryption Keys for each Address in the Address Pool given by;

" 0000:00E000.0000... "

then the Total Number of Addresses in the IPt2 Specification equals:

$$((2^{646,464.99987654321}) \times (2^{646,464}(16 \text{ x } 99,987,654,321))) = \text{IPt2} = 2^N$$

Or, The 'Payload Capacity' of ONE

Intelligent Quantum Worm = $(2^{646,464}(16 \text{ x } 99,987,654,321)$ Bits;

Or $((2^{646,464.99987654321}) \times (2^{646,464}(16 \text{ x } 99,987,654,321)$ Copies of IPt1

Or; each Address Segment of the IP Address,

'Router-ID + Trunk-Identifier + Network IP Address,**

which is defined by;

0000:00E000.0000... + 0000:00E000.0000... + 0000:00E000.0000...,

and equals;

$$2^{2(646,464)} = 2^{X} = 2^{1,292,928.99987654321} \text{ copies of IPt1}$$

the 'Polymorphing Intelligent Quantum Worm Protocol Data Compression Engine', now defines each Address Segment with having either an individual Encryption Algorithm, or a Data Compression Algorithm(s);

e.g. One Address Segment defines;

(0000:00E000.0000...) X (0000:00.0000...E000)

or each Address Segment of the 'Intelligent Quantum Worm' has a Payload Capacity equal to;

$$(2^{646,464}) \text{ X } (16 \text{ x } 99,987,654,321) \text{ Bits.}$$

Given that each Segment's Address Pool Capacity equals;

$$(2^{646,464.99987654321}) = (0000{:}00E000.0000...) = 2^{X} = 2^{N}$$

Addresses.

6.2 Special IANA Consideration

- 6.2.2 - Security

Even still, possible interpretation(s) notwithstanding, mathematically speaking however, the (" 00.0000...E000 ") String can only define the "Payload Carrying Capacity" of a Data Compression Engine. In which case, the Encryption Coding Strings can be included and defined as an integral part of the Data Stream Code, which has a 'Bit-Count Length' determined by the Scaleable 'Payload Capacity' of the 'Polymorphing Intelligent Quantum Worm Protocol Data Compression Engine' (P-IQWP-DCE). Nevertheless, this clarification completes the design specification for the 'Intelligent Quantum Worm Protocol', founded upon the logical derivation of the Binary System of Enumeration from the Mathematics of Quantification; it finalizes the unique design structure of the IPtX Protocol 'Family' Specification.

And this is true since;

$$\text{` } 00.0000...E000 \text{ `} = 2^{X} = 2^{N} = \text{` } 00E000.0000... * \text{`};$$

Noting more specifically that the Set of Points at which these Sets are equal, is Finite. Hence, the number of occurrences in which these Sets are equal is defined by;

$$\text{`}2^{X} = 2^{N}\text{`}:$$

"The Special Case defining the Distributive Laws representing the Binary and the Unary Sets*". Furthermore, there is a broader interpretation of the " P-IQWP-DCE " Binary Algorithm, in which the Ratio, {2^{X} : 1} defining this engine represents the Probability or Likelihood of a 'Security Event' occurrence vs. "Quality of Service/Product" to prevent it, because 'Security Protection Coding' is packaged in the "Payload": with an excellent Virus Protection Software, e.g., -

Consumes Limited or Balanced Security Code to Message Space Ratio – Possible Number of Virus Combinations vs. Anti-Virus Combinations ; This is an Absolute Protocol Protection Environment Ratio defining the " P-IQWP-DCE " Space -

$$‘2^{X} = 2^{N}‘;$$

- Balance is Achievable or the Time required to perform a 'Security Event' could be of such a duration, it would seem infinite! In which case, the

$$\{2^{X} : 1\}$$

Ratio represents the scale of comparison, providing the odds of determining the Probability of preventing a 'Security Event', when using the currently available Security Safe Guards. And more importantly –

This Ratio $\{2^{X} : 1\}$ actually depicts a 'Fully Cognizance Autonomous Artificial Intelligence', a True Binary Coding Environment, defining a logical mathematical foundation derived from the Laws of Set Theory and the Mathematics of Quantification. In other words, the spectrum of possibilities defined by '2^{X}' is unlimited; since '2^{X}' represents the Binary equation defining the incremental progression by successive additions of '1', and if 2^{X} is an Element of the Real Number Set, then

$$‘2^{X} = 3^{X} = 4^{X} = 5^{X} = N^{X+1}’.$$

In other words, since nesting is inherently defined, the 'DCE's' Payload could define a series of Binary Equations representing a Nested or Encapsulated Data Stream Transmission.

6.2 Special IANA Consideration

- 6.2.3 – Summary

In summary, the conclusions from Mathematics of Quantification that established a true Binary Environment, produced a Revolutionary method of Coding, using the Exponential Expression '2^X', correlating the Mathematical Language of the Binary System, defining a Unary progression, resulting in the unique logical design of the IPtX Protocol Family Addressing Specification. Hence, the final imperative is to display"IPt2's" remaining 64 Bit-Mapped Protocol Specifications design for the 'Intelligent Quantum Worm Protocol', all of which measure a different size Address Pool correlating with the Address Class Specification of IPv4 – Furthermore, the additional benefits this expansion yields, magnifies the potential of the IPtX-MX Universal Protocol. In other words, there are 5 Sub-Level Divisions of the ' IPt2 ' Specification, and each contains an 'Address Class Sub-Division'; Set = {A, B, C, D, E}:

– TABLE VIII –

- Class A Address Pool Allocation -

1) (0000:0E0000.0000...), or (0000:0.0000...E0000)

- Class B Address Pool Allocation -

2) 00.0000...E000 ' = 2^X = 2^N = ' 00E000.0000...*

- Class C Address Pool Allocation -

3) (0000:000E.00.0000...), or (0000:000.0000...E.00)

- Class D Address Pool Allocation -

4) (0000:0000E0.0000...) , or (0000:0000.0000...E0)

- Class E Address Pool Allocation -

5) (0000:00000E.0000...), or (0000:.0000...E00000)

Note: **Because the Mathematical relationship between the ' 4 ' Unit structure of an IP Address defined in the IPv4 Specification represents the identical, ' 4 ' Unit Pattern defining the divisional structure for the 'DCE' Unit defining the 'IPtX-MX Universal Protocol' in the IPtX Specification - The Address Pool Allocation table defining the Address Class Distribution in Table VIII actually represents the definition of the '5' Sub-Divisions - 'Address Scaling Division'; a 'Mathematical Scaling Communication Protocol' that defines a procedure matching the specifications related to the parameters defining 'Data Transmission Performance Rates' of the Communicating Nodes – by Table VIX;**

TABLE VIX

1) IPt2-a - Class A Address Pool Allocation -

1.1) (0000:0E0000.0000...), or (0000:0.0000...E0000)

Address Class Sub-division - Set = {A, B, C, D, E}

2) IPt2-b - Class B Address Pool Allocation -

2.1) 00.0000...E000 ‘ = 2^{X} = 2^{N} = ‘ 00E000.0000...*

Address Class Sub-division - Set = {A, B, C, D, E}

3) IPt2-c - Class C Address Pool Allocation -

3.1) (0000:000E00.0000...), or (0000:000.0000...E00)

Address Class Sub-division - Set = {A, B, C, D, E}

4) IPt2-d - Class D Address Pool Allocation -

4.1) (0000:0000E0.0000...) , or (0000:0000.0000...E0)

Address Class Sub-division - Set = {A, B, C, D, E}

5) IPt2-e - Class E Address Pool Allocation -

5.1) (0000:00000E.0000...), or (0000:.0000...E00000)

Address Class Sub-division - Set = {A, B, C, D, E}

[If $X^X = 2^X$, then $X = 2^{1/X}$, when 1/X = ' .00000… ': and this is valid because each Address Class Allocation will contain a different number of Binary { '2^F' } definitions.]

In other words, the IPtX Addressing Protocol "Family" is defined by a mathematical foundation that defines an incremental progression using successive additions of '1', which also defines the sequence of counting defined by the Unary set. In which case, if the mathematical laws defining the relationships for the Address patterns of the IPtX Specification are mathematically consistent, then Table VIX defines a mathematical expansion for every Address Class Specification defined by the IPtX Protocol "Family".

- The Additional benefits from the IPtX-MX DCE Unit Mathematical Foundation -

1) Data Transmission Rates = 'DCE' Unit Classification

2) 'Performance' & 'Dimensional Design' = 'DCE' Unit Classification

3) Data Transmission Communication Rates = 'DCE' Unit Classification

4) Data Transmission Rates Coefficient = ' X '

5) Data Transmission Rate(s) Equation = '$X(2^X) = 2^X = 2^N = X^X$'

6) Control Data Transmission Rates / Matching the Node(s) Rate(s)

7) 'DCE' Unit Classification determines Bit Transmission Scale

8) 'DCE' Unit Classification = 'Storage Capacity Scale' Unit

9) 'DCE' Unit = "Hardware Compatibility Scale" Classification;
- **Matching Data Transmission Communication Rates**
- **'HEX' Numbering System Replacement**
- **Matching Binary Numbering System Scale = $X(2^X)$**

10) 'DCE' Unit Classification = "Calibrating the Encoding of Code'
- **a) Defines New Binary Coding Operator for Software Construction and Design**
- **b) Defines New Binary Definition for Instruction Set and Scale for CPU Design**
- **c) Defines New Data Communication Hardware Orientation Connectivity Association Specification – Eliminating Device Drivers, USB, ATA, SCSI, etc, and a host of additional Hardware Control Protocols.**
- **d) Scaleable 'DCE' Unit Defines a New Host IP Addressing Specification, which provides a functional use, by TABLE VIX, for the IPtX-MX Universal Protocol that would allow Host IP Address Activation for a Direct Internet**

Connection - The Activation of every Host defined in the Overlay of IPtX Specification {See TABLE VIX} - Adding a Host designation function to the CDIR Network Descriptor, which assigns the DCE Unit Protocol Specification to the trailing end Switch; 'h': e.g. ' /0000:00-h ' - and this allows every Network and Host IP Address to have a unique Identification in the IPtX Specification.

e) **Binary Encoding Conversion example; '1111' = 'DCE' Unit' = 2^X, or ' 8 ' = 111 = 'DCE' Unit' = $2^X = 2^3$ '; the Benefits are easily realized when the conversion represents a large Number.**

Note: Assigning a DCE Unit Protocol(s) -
Host Address Activation - IANA Specifications
CIDR Network Descriptor Host Activation Switch –
Host Activation Switch 'h' – ' /0000:00-h ' - IANA Specifications

e.g.: The respective reduction and expansion of the 'DCE' Unit:

a) **IPt1 Specification - 4 Bit Address Class ID: 2 Bits Base in 2^X. - 4 Bit 'E'- Exponent - Exponent Parts: - 2 Bit Integer. - 20 Bit Decimal String - 32 Bit Address – ' 0000:00E00.000… 32Bit Intelligent Quantum Worm '**

b) **IPt4 Specification - 16 Bit Address Class ID: 8 Bit Base in 2^X. - 4 Bit 'E' Exponent - Exponent Parts: - 28 Bit Integer. - 72 Bit Decimal String - 128 Bit Address– ' 00.00:0000E0000$_{28}$.000…$_{72}$ - 128Bit Intelligent Quantum Worm '**

Nevertheless, the Intelligent Quantum Worm Technology maintains a scaleable range that defines 5 Address Class Specifications, which represent a different and independent copy of the IPt1 Specification. And this defines the Address Pool of the 64 Bit IPt2 Specification, with a capacity equal to 5 copies of IPt1;

or $5(2^{32}) = 5(4,294,967,296) = 2.147836 \times 10^{10}$;

or "2.147836×10^{10} 'Intelligent Quantum Worms'" that can all share or be assigned to service 4,294,967,296 IP Addresses.

That is, this defines a { 5 : 1 } Ratio, a mathematical relationship sustained throughout the IPtX Specification, which defines approximately 5 different Worm Configurations for every IP Address Class defined by the IPtX Addressing Protocol Family Specification.

Note: This defines the Theoretical Address Pool Capacity of the IPt2 Specification as being equal to:

(0000:000E00.0000...), or (0000:000.0000...E00)

'Router-ID + Trunk-Identifier + Network IP Address*

0000:00E000.0000... + 0000:00E000.0000... + 0000:00E000.0000...

$$\text{Router-ID} = 2(2.147836 \times 10^{10})(2^{646,464}) \times (4,294,967,296)$$

$$+$$

$$\text{Trunk-Identifier} = 2(2.147836 \times 10^{10})(2^{646,464}) \times (4,294,967,296)$$

$$+$$

$$\text{Network IP Address} = 2(2.147836 \times 10^{10})(2^{646,464}) \times (4,294,967,296)$$

$$= 12[(2.147836 \times 10^{10})(2^{646,464})] \times (1.2884902 \times 10^{10})$$

approximately available IP Addresses.

This is a good approximation, because this value (2^646,464) changes, Higher to Lower, values defining 5 Categories of the IPt2 Specification.

In other words, this essentially provides the 'Intelligent Quantum Worm' Protocol with a Scalable "Payload Capacity", which defines a 3-D Spatial Distortion that mimics a Tunneling Effect. In which case, perhaps the "IPtX-MX Universal Protocol" should rightfully be called:

An "Intelligent Quantum Tunneling Worm".

Nevertheless, the ability to program or construct a 3-D Space, implies the possibility that today's 'Bit-Mapped' Specification associates a Bit with a displacement equal to an extremely large Electron Surface Area, one that encompasses the measurement of an extremely large Pool (approximately $2^{646,464}$) or Group of Electrons. In other words, the "Intelligent Quantum Tunneling Worm" Protocol, in such an environment, would define the Unit Bit as the minimum excitation energy required to displace an Electron One Electron-Mass Displacement Unit [7]. And then, if it's A.I., there is the possibility that it could morph itself, acquiring the specifications to accommodate a 3-D Spatial Environment. - The Illusion by Shading, rendering any picture or video with a 3-D effect when displayed.

Nevertheless, while the foregoing proofs are confirmed by the conclusions from the proof of “Fermat’s Last Theorem”: However, a more thorough understanding of the results from the successive additions of '1', and the Basic Theory of Mathematics, which define Sequential Counting is required to understand the error in the current definition of Infinity; where 1/0 = INFINITY. And how is it possible to define an Infinitely Large Address Pool to the IPtX Specification.

The Proof;

First, accepting that 1/0 = INFINITY, is not true, since Division is defined [3] as Subtraction, then the results from the continuous operation defined by eq-a, is pointless, because the result from this equation defines the REMAINER, and if the difference between the Remainder and the Quotient equals Zero, then the solution, by eq-b, of 1/0, is equal to Zero.

eq-a. $1 - 0 = 1 - 0 = \ldots = (1 - 0)_{N+1} = 1$

eq-b. $1 / 0 = 1 - 1 = 0$

And if it can be said that the relationship between the Unary Set and the Binary defines a { 1 : 11 } Ratio, where Cardinality representing this Ratio, given by eq-c, is True.

eq-c. ' 1 = 2 ' - Unary Set = {1} - Binary Set = {11}

"Given that: 11 = 1 + 1 = 2"

Then the Definition of a Prime Number is Given By:‘

eq-d. '1 = 2’ - 'Prime Numbers'

> **A 'Prime Number' or 'Prime Integer', is a positive integer, ‘ p | p ≥ 1 ', that has no positive integer divisors other than itself, ' p ', and ' 1 '.**

In which case, from the definition of a Prime Number, it can be concluded that if every Number except '1', is defined as the Progressive Additions of '1', then "1" defines the Absolute Number, which is equal to the Cardinal Number that defines the Universal Multiplicative Identity Element, representing every Item defining itself [2].

Clearly, the Modern Numbering System evolve defining the Zero distinction as the Multiplicative Coefficient defining the Count of the Members in the Set of ' 1's ' containing 10 Elements. Nevertheless, Nature's Mathematical System defines the Binary pair {0, 1}, as the symbols to be used for numerology, which ultimately refined into the more mathematical symbolic numerical representation used today.

In other words, since '1' is the absolute Numeral, and the Binary and Unary Sets must, if they are equal and enumerate to infinity, map equally, in a One-to-One Correspondence with the Real Number Set, which is counted using successive additions of '1'.

Then the Ratio defining the relationship between the Binary and the Unary Sets, and represented in the mathematical expression defining eq-c, since any One-to-One defines a sequential count defined by the Unary Set as the successive additions using 1's, also defines 2 Infinities.

eq-c. ' 1 = 2 ' - Unary Set = {1} = ∞ = Binary Set = {11} = ∞

Furthermore, since Infinity is not enumerable, then Infinity must define a Prime Number, because only the mathematical operations involving 'ITSELF' and '1' are defined.

Hence, "Given that: 11 = 1 + 1 = 2", and since eq-c is true, then by eq-e, we have;

eq-e. ' ∞ ' = (∞ + ∞) = ' ∞ '

And from eq-d, since multiplication is equal to the quantified sum of addition, we have by eq-f;

eq-f. ' ∞ ' = (∞ + ∞) = Infinities, (∞ + ∞) = ' ∞ ' = Infinity

Therefore, since eq-d, " 1 = 2 - Prime Numbers ", is true, then;

(∞ + ∞) = Infinity = Infinity = (∞ + ∞)

And this, since division is the quantified difference of the repeated subtractions performed on a constant, further implies;

Infinity/ Infinity = ' 1 ' and (Infinity - Infinity) = 0

Nevertheless, since the foregoing equations defines the mathematical operations involving Infinity, and since these operations also defines the enumeration of the Elements defining the respective members contained in the Binary and Unary Sets. Then the Elements these Set respectively contain, each enumerate an individual count using successive additions of '1', in a One-to-One Correspondence, to reach an Infinite Count of the Members each Set contains.

In other words, by definition [3], since the

Unary Set = {1},

and the

Binary Set = {0, 1}

Then Infinity is the definition of 2 Sets, which defines;

The Binary Set defines; 1 either followed by an Infinite Zero String, or a Decimal Point followed by an Infinite Zero String terminated with a '1'.

1) **A '1' followed by an Infinite String of Zeros**
2) **$1 + 0_{INFINITY}$ … 000000000000000000000000000000000000.0**
3) **A '1' proceeded by an Infinite String of Zeros**
4) **0.000000000000000000000000000000000000 … $0_{INFINITY} + 1$**

The Unary Set defines an Infinite String of ' 1's ', either before or after the Decimal Point.

5) **A '1' followed by an Infinite String of '1's'**
6) **$1 + 1_{INFINITY}$ … 111111111111111111111111111111111111.0**
7) **A '1' proceeded by an Infinite String of '1's'**
8) **0.111111111111111111111111111111111111 … $1_{INFINITY} + 1$**

Note: Both examples define every element contained in the Set of Real Numbers. And more importantly, if the Infinite Zero String is countable, then succeeding or following the Zero String is also countable... Also, this proof can be construed as equating the Count of the '1's' in the Unary Set with the Zeros, and the 1 (occurring at a Count one unit beyond the Infinite Zero String) in the Binary Set – where Infinity defines; "Forever Plus '1' " - Forever Plus '1' = ' $1 + \infty$ ' [3].

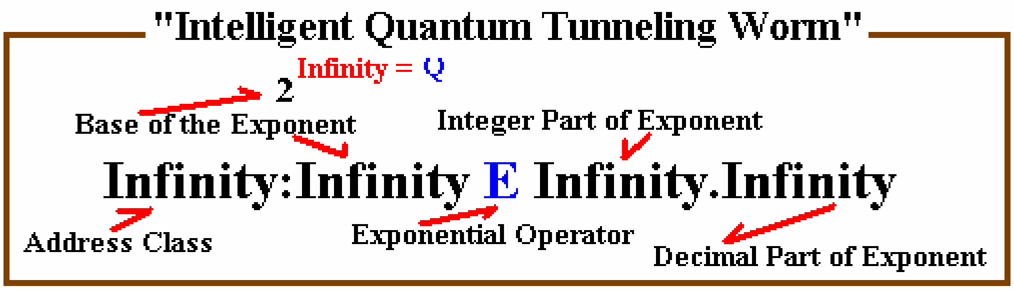


FIGURE 7

"Using Current Binary Translation"

'Showing examples of the Concept of "Footprint" Size Reduction'

$32 + 32 = 64 = 2^{X} + 2^{X} = 2^{5} + 2^{5} = 8^{2} = 2^{6}$

32 Bit String + 32 Bit String = 64 Bit String

which means; - 32 Bit String + 32 Bit String = 2^{6},

and - 32 Bit String = 2^{5}

' e.g. ' a 32 Bit String =

" 10111001100001001111111100010101011000111000111000111000111000111 "

= 32 Bit String = ' 11111111111111111111111111111111 ' = 2^{5}

32 Bit String = 2^{5} = '2E5' = 111111111 '9 Bit String Displacement'

= 32 Bit String = ' INTEGER ' = 2^{5}

[Special NOTE: Possible Concerns of IANA, IESG, and IETF - 'How Does the IPtX / IPtX-MX Protocol comply with the Octet Rule? Noting specifically that, IPv4 Bit-Maps to the current Backbone, and the proposed structure for the IPv6 Backbone adds another difference.]

The Difference between Binary & Unary Systems of Counting
"Multiplication is the Quantified Sum of Addition"

	1. Binary Representation for Counting is Multiplication	2. Binary & Unary Counting	3. Unary Representation for Counting is Addition
	$0^{X} = 0$	0	0+0 = 0
1.	$2^{0} = 1$	00 = 1	1 = 1
2.	$2^{1} = 2$	01 = 11	1+1 = 2
3.	$2^{F} = 3$	10 = 111	1+1+1 = 3
4.	$2^{2} = 4$	11 = 1111	1+1+1+1 = 4
5.	$2^{F} = 5$	100 = 11111	1+1+1+1+1 = 5
6.	$2^{F} = 6$	101 = 111111	1+1+1+1+1+1 = 6
7.	$2^{F} = 7$	110 = 1111111	1+1+1+1+1+1+1 = 7
8.	$2^{3} = 8$	111 = 11111111	1+1+1+1+1+1+1+1 = 8
	:	:	:
	:	:	:
	:	:	:
16.	$2^{4} = 16$	1111 = 1111111111111111	16 = 1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1

{See - Work(s) in Progress [11] "The Mathematics of Quantification, and the Rudiments Of the Ternary Logical States of the Binary Systems"}

Clearly, since Nature's method of Counting, just as in Particle Physics, and Electronics, uses the Base 2 in Exponential Enumeration - I wonder; 'How else could a Binary Pair be Counted?'

In other words, only SOFTWARE (Station or Node Location) can represent or define the ZERO Concept, because a Zero Signal cannot [true for the Binary as well as the Unary Systems] be Transmitted Electronically [0 = EX = 0EX; Or if,

'X = 0', then 0 = 0E0 = E, as in; 0000:00E0000.0000…

= 256:00E0000.0000… = '256: E . (+ Padding to the Bit-Mapped Specification)…']

Hence, when an Electronic Signal represents Binary Zero, '00' (where 2^{0} = 2E0 = ' 1 ' = '00'); it has a value equal to the Unary Set, or '1'.

	Binary System	Zero	Exponential System of Counting
	No Definition	0	0^{X} = 0 = 0EX
1.	00 = aa	No Definition	2^{0} = 1 = 2E0
2.	01 = ab	No Definition	2^{1} = 2 = 2E1
3.	10 = ba	No Definition	2^{F} = 3 = 2EF
4.	11 = bb	No Definition	2^{2} = 4 = 2E2
	:		:
	:		:
	:		:
8.	111 = bbb	No Definition	2^{3} = 8 = 2E3
9.	1000 = baaa	No Definition	2^{F} = 9 = 2EF
10.	1001 = baab	No Definition	2^{F} = 10 = 2EF

[Given that: E = Exponential Operator; F = Variable Irrational Number; and X = Any Variable defined as a Member of the Real Number Set]

IPt2 vs IPv6 Protocol

IPv6 IP Addressing Speecification;

= 128 Bit-Mapped Displacement

= [11111111.11111111.11111111.11111111.
11111111.11111111.11111111.11111111.
11111111.11111111.11111111.11111111.
11111111.11111111.11111111.11111111]

= 2^{128} ~ 3.403×10^{38}

= 2^{128}

~ 340,282,366,920,938,463,463,374,607,431,768,211,456

= Total Number of Available Addresses in the IPv6 Specification

= A 48 Digit Number = 128-bits = 16 Octets

- More typically, the IPv6 addresses are written as eight groups of four hexadecimal digits; e.g.:

= 2001:0db8:85a3:08d3:1319:8a2e:0370:7334 - 'valid IPv6 address'

= [11111010001:110110111000:1000010110100011:100011010011:
10100100111:1000101000101110:101110010:1110010100110]

[Note: Just Imagine the difficulty trying to Configure, or Troubleshoot a 7,000 Node Network using 'valid IPv6 addresses']

Now, retaining the Octet Rules, IPt2 / IPt2-MX represents;

IPt2 IP Addressing Specification;

64-bit length = 64 Bit-Mapped Displacement

= [11111111.11111111.11111111.11111111.
11111111.11111111.11111111.11111111.]

= [11111111 (8 Bit Prefix): = 2^{8}
11 (2 Bit Base) = 2^{2}
11111111 (E = Exponential Operator - 8 Bits) = 2^{8}
111111111111111111111111111111 (Exponent - 30 Bits) = 2^{30}
.1111111111111111 (Decimal String Accuracy - 16 Bits) = 2^{16}]

= 0000:2EX.000 = XXX:2EX.000 = 256:$4^{1,073,741,824.6553}$

= XXX:2EX.000 = 256($4^{1,073,741,824}$).65536

= XXX:2EX.000 = 256($2^{2,147,483,648}$).65536

= Total Number of Available Addresses in the IPt2 Specification

= XXX:2EX = 256($2^{2,147,483,648}$) = $2^{2,147,483,656}$ IP Addresses

- Or - This translates into a Number representing;

2,147,483,656 Bit-Mapped Length

- Or - A Number containing;

268,435,457 Octets

805,306,371 Digits

And if you will take note this represents a Number so Large, No Computer TODAY, can determine or calculate its actual value -

```
64 Bit-Mapped Displacement  =  0000:00E0000.0000...

                                    28 "Or" 30
        8 Bit Prefix                Bit Exponent
              \         2 "Or" 4         /
               \        Bit Base        /
                \           |          /
                0000  :  00    E  0000  .  0000...
                               |              \
              8 Bit Exponential Operator        \
                                                  \
                       16 Bit Exponential Decimal String
```

256:16 E 28 Bits .XXX... = 0000:00E0000.0000...

256:16 E 268,435,456 . XXX... = $256(16^{268,435,456})$. XXX...

$256(16^{268,435,456})$.XXX... = $256(2^{4(268,435,456)})$.XXX...

$256(2^{4(268,435,456)})$.XXX... = $256(2^{1,073,741,824})$ XXX...

And a Bit-Mapped Displacement of 1,073,741,824 Bits,

contains 134,217,728 Octets -

134,217,728 Octets represents a Number

Approximately (3 x 134,217,728)

402,653,184 Digit Number

256(402,653,184) = 103,079,215,104 Digit Number

- Or -

256:4 E 30 Bits.XXX... = 0000:00E0000.0000...

$256(4^{1,073,741,824}).XXX... = 256(2^{2(1,073,741,824)}).XXX...$

$256(2^{2,147,483,648}).XXX... = 0000:00E0000.0000...$

And a Bit-Mapped Displacement of 2,147,483,648 Bits,

contains 268,435,456 Octets -

268,435,456 Octets represents a Number

Approximately (3 x 268,435,456)

805,306,368 Digit Number

256(805,306,368) = 206,158,430,208 Digit Number

Total Number of Available IP Addresses in the Address Pool for the IPt2 Specification.

Yep! This number only has a 64 Bit Foot-Print - vs 128 Bit Length

There is no doubt that the profoundness of Figure 7 seems to overshadow the simplicity and flexibility of the design of the IPtX Addressing Protocol Family Specification. In which case, and rightfully so, I must conclude the Summary of the IPtX design specification with the explanation of the description defining the 32 Bit version of IPtX-MX Protocol, and how it applies to the IPt1 Specification. And this, it shall be seen, will provide a more gradual interpretation, elaborating the procedural change defining the steps of the expansive effects describing how "The CIDR Network Descriptor expands the size of the IPtX Address Space beyond the IPv6 IP Addressing Specification" – concluding more importantly, with the correct mathematical expression reflecting these changes, which is supported by a logical argument derived from the proof of "Fermat's Last Theorem", and defined by the Mathematics of Quantification.

+--------TABLE X--------+
IPt1; The preferred IPtX
Open - ClassLess - Bit-Mapped Routing Architecture

- IPt1 Address Class -
'Network Connection - Addressing "Location/Destination" Protocol'

- IPtX-MX 32 Bit {Masked} Universal Protocol -
'Masked Address Protocol -
Data Streaming Packet -
Variable "Payload" Encapsulation Capacity -
Encapsulation of Linear Programmed
Information/Instruction Code -
(e.g. IP Header, TCP/IP Encapsulation Protocols,
Else Encapsulation Protocols, Info etc.)'

- CIDR Network Descriptor "RIP" "TELe-RIP" "AS-RIP" Protocol -
"RIP" - Routing Information Protocol
"AS-RIP" - Address Segment Routing Information Protocol
"TELe-RIP" - Internet Telephone "ENUM" Routing Information Protocol
"TELe-ARP" - eTelephone "DCE-Unit" Address Recognition Protocol
"MUM-ARP" - Masking/Un-Masking "DCE" Address Recognition Protocol
'Initiating the Binary Algorithm Choosing and Encoding
the "P-IQWP-DCE" "DCE" Unit for "MUM-ARP" Identification'

- Distribution Overlay -

TABLE XI
CDIR Network Descriptor - Scaling IPt1 Address Space
'Scaling the Anatomy of a IPtX Address String'

Router - ID + Trunk-Identifier - ID + Network - ID

1) /0000:00 = 0 + +/0000:00 = 0 + -/0000:00 = 1
Copies of 'IPt1' = 1 'Total Address Length 32 Bits'

2) /0000:00 = 0 + +/0000:00 = 1 + -/0000:00 = 1
Copies of 'IPt1' = 256^4 'Total Address Length 64 Bits'

3) /0000:00 = 1 + +/0000:00 = 1 + -/0000:00 = 1
Copies of 'IPt1' = (256^4) x 256 'Total Address Length 96 Bits'

TABLE XI

CDIR Network Descriptor - Scaling IPt1 Address Space

'Scaling the Anatomy of a IPtX Address String'

IPt1

IPtX-MX Universal Protocol Allocation

- Class A Address Po ol Allocation -

1) (0000:0E000.0000...) = 0^{X} ., or (0000:0.0000...E000)

- Class B Address Pool Allocation -

2) 00.0000...E00 ' = 2^{X} = 2^{N} = ' 00E00.0000...

- Class C Address Pool Allocation -

3) (0000:000E0.0000...), or (0000:000.0000...E0)

- Class D Address Pool Allocation -

4) (0000:0000E.0000...) , or (000.000.000.000 - IPt1/IPv4)

- Class E Address Pool Allocation -

5) (0000:0000E.0000...), or (0000:.0000...E0000)

Note: The conjecture leading to the possibility of a 'Quanta Electron', or 'Electron' Particle Size Variation is clearly a rational assumption, defined by the "Rudiments of Finite Physics", as defining the Pattern of the Accelerated Particle as a function of its "Mass-Displacement" Unit. [7]

TABLE XII
IPtX Addressing Specification - '1)'
'CDIR Network Descriptor - Scaling IPt1 Address Space'
- IPtX-MX 32 Bit "Masked" Protocol Specification -

1) /0000:00 = 0 + +/0000:00 = 0 + -/0000:00 = 1

'Total Address Length 32 Bits - '1' Copy of the IPt1 Specification'
Network - ID: Address String Length - IANA/FCC Specifications

'Network ID Address Mask Bit-Mapped Specification'
- IPtX-MX 32 Bit "Masked" Bit-Mapped Specification -
- "Front-End" - "Back-End" Address Masking Specification -
- IANA/FCC Specifications -

'(0000:0E000.0000...) = 0^{X} , or (0000:0.0000...E000)'
- IANA/FCC Specifications -

'OverLay Design/Address Bit-Mapped Specification'
- See FIGURE 5 -

Note: The actual difference the Theory of the IPtX Specification presents, is defined by the difference between Linear and Non-Linear Coding, or the choice between Encoding the Bit-Map of the Data Stream or Encoding the Bit-Map of the Equation for the Data Stream.

TABLE XIII
IPtX Addressing Specification - '2)'
'CDIR Network Descriptor - Scaling IPt1 Address Space'
- IPtX-MX 32 Bit "Masked" Protocol Specification -

1) /0000:00 = 0 + +/0000:00 = 0 + -/0000:00 = 1

'Total Address Length 32 Bits - '1' Copy of the IPt1 Specification'
Network - ID: Address String Length - IANA/FCC Specifications

'Network ID Address Mask Bit-Mapped Specification'
- IPtX-MX 32 Bit "Masked" Bit-Mapped Specification -
- "Front-End" - "Back-End" Address Masking Specification -
- IANA/FCC Specifications -

' 00.0000...E00 ' = 2^{X} = 2^{N} = ' 00E00.0000... '
- IANA/FCC Specifications -

'OverLay Design/Address Bit-Mapped Specification'
- See FIGURE 5 -

TABLE XIV
IPtX Addressing Specification - '3)'
'CDIR Network Descriptor - Scaling IPt1 Address Space'
- IPtX-MX 32 Bit "Masked" Protocol Specification -

1) /0000:00 = 0 + +/0000:00 = 0 + -/0000:00 = 1

'Total Address Length 32 Bits - '1' Copy of the IPt1 Specification'
Network - ID: Address String Length - IANA/FCC Specifications

'Network ID Address Mask Bit-Mapped Specification'
- IPtX-MX 32 Bit "Masked" Bit-Mapped Specification -
- "Front-End" - "Back-End" Address Masking Specification -
- IANA/FCC Specifications -

' (0000:000E0.0000...), or (0000:000.0000...E0)'
- IANA/FCC Specifications -

'OverLay Design/Address Bit-Mapped Specification'
- See FIGURE 5 -

TABLE XIV
IPtX Addressing Specification - '4)'
'CDIR Network Descriptor - Scaling IPt1 Address Space'
- IPtX-MX 32 Bit "Masked" Protocol Specification -

1) /0000:00 = 0 + +/0000:00 = 0 + -/0000:00 = 1

'Total Address Length 32 Bits - '1' Copy of the IPt1 Specification'
Network - ID: Address String Length - IANA/FCC Specifications

'Network ID Address Mask Bit-Mapped Specification'
- IPtX-MX 32 Bit "Masked" Bit-Mapped Specification -
- "Front-End" - "Back-End" Address Masking Specification -
- IANA/FCC Specifications -

' (0000:0000E.0000...) , or (000.000.000.000 - IPt1/IPv4)'
- IANA/FCC Specifications -

'OverLay Design/Address Bit-Mapped Specification'
- See FIGURE 5 -

TABLE XIV
IPtX Addressing Specification - '5)'
'CDIR Network Descriptor - Scaling IPt1 Address Space'
- IPtX-MX 32 Bit "Masked" Protocol Specification -

1) /0000:00 = 0 + +/0000:00 = 0 + -/0000:00 = 1

'Total Address Length 32 Bits - '1' Copy of the IPt1 Specification'
Network - ID: Address String Length - IANA/FCC Specifications

'Network ID Address Mask Bit-Mapped Specification'
- IPtX-MX 32 Bit "Masked" Bit-Mapped Specification -
- "Front-End" - "Back-End" Address Masking Specification -
- IANA/FCC Specifications -

'(0000:0000E.0000...), or (0000:.0000...E0000)'
- IANA/FCC Specifications -

'OverLay Design/Address Bit-Mapped Specification'
- See FIGURE 5 -

Now, take another Look at the IPt1/IPt1-MX Protocol Specification, and then compare it to the IPv6 Specification:

```
                     Back-End 'Only' Protocol Format
                                  /    /    /
32 Bit-Mapped Displacement  = 0000:00E0000.0000... = IPt1/IPt1-MX

32 Bit Length = 11111111.11111111.11111111.11111111 = IPv4 / IPt1
                      /         /         /
                 Back-End 'Only' Protocol Format
```

```
                               2^6  = 6 Bit Exponent = 64 Bits
 2^8  = 8 Bit Prefix                     /
             \    2^2 = 2 Bit Base     /
              \         |             /
               0000  :  00   E  0000  .  0000...
                             |              \
             8 Bit Exponential Operator      \
                                              \
                     2^8  = 8 Bit Exponential Decimal String
```

```
256:4 E 64.XXXX = 0000:00E0000.0000...

256:4 E 64.XXXX = 256:4E64.XXX... = 256(4^64 ).XXX...

256:4 E 64.XXXX = 256(4^64 ).XXX... = 256(2^128 )

                = 256(3.4028236692093846346337460743177e+38)

                = 8.7112285931760246646623899502533e+40


 {Maximum Number of Available IP Addresses Contained in a 32
  Bit Foot Print... which is Greater than the IPv6 Specification.}
```

```
[Note: Regardless of the Backbone Configuration, the IPtX
       Specification eliminates IP Address loss by; using only
       the Base 2 Exponential methods for Enumeration, then
       using: The IPtX-MX Protocol; A 'Non-ZERO Prefixed'
       Addressing System that counts sequentially starting with
       '1', and when using the current Backbone Configuration,
       the lost IP Addresses are converted to Host Addresses.]
```

TABLE XV

```
                  |- Omni Directional Linear Flat Space
                 /
                /- Omni Directional Linear Layered Flat Space
'DNS ZONE' --
                \- Omni Directional Non Linear Layered Flat Space
                 \
                  |- Omni Directional Non Linear 3-D Space Cloud;
                   "The Mathematical proof establishing equality
                    between the measured dimensions associated
                    with the 'Circle and the Square' and the
                    'Sphere and the Cube' [6]."

'DNS ZONE - Address Block Specification'

 - ZONE IP - Primary Node of the 'DNS ZONE' Bit-Mapped Space
 - IP AREA CODE - IPt! Address Block Specification - X(2^32 )


IPtX-MX UNIVERSAL PROTOCOL - Specification Bit-Map

    - Identifiable Bit-Mapped Address Classification
    - Range Scaling Data Stream Classification
    - Ratio Matching Copies of IPtX Specification {5 :1}
```

Clearly, while the Address Pool equation, $X(2^{32})$, which provides the design of the IPtX Addressing Protocol Family Specification with an impressive IP Address Pool address count availability.

eq-1: IPtX $= X(2^{32}) + 16{,}900{,}000 = \text{Infinity}$

eq-2: IPv6 $= X(2^{32}) = (2^{128}) = (256^{16}) = 3.4028237 \times 10^{38}$

eq-3: IPt4 $= X(2^{32}) + 16{,}900{,}000$
$= (2^{128}) + 16{,}900{,}000$
$= (256^{16}) + 16{,}900{,}000$
$= 3.4028237 \times 10^{38} + 16{,}900{,}000$ **Host Addresses**

eq-4: IPv4 $= X(2^{32})$; when $X = 1$.

However, it is from the revised definition of the role of the "CIDR Network Descriptor", by TABLE XVI, which actually expands the Address Space of the IPtX Specification beyond IPv6.

Table of Scaleable/Expandable Switch Function defined for the "CIDR Network Descriptor"

TABLE XVI

"CIDR Network Descriptor" - /0000:00

Scaleable/Expandable Switch Function - {'|', '?', '+', '-', h, s, etc}

"CIDR Network Descriptor" - /0000:00 - {'|', '?', '+', '-', h, s, etc}

Current Definitions of the "/0000:00" {'Switch'(s)}

- **'s' - Subnet Network IP Address Encapsulated by Network-ID**
- **Pointer for Encapsulation Network -ID --> Subnet-ID**
- **Pointer for Encapsulation Network -ID --> Host-ID**
- **Switch - Network-ID /0000:00 s - Network Subnet Address Pool**
- **Switch - Subnet-ID = Network-ID /0000:00 -s**
- **Switch - Network-ID /0000:00 +s - Subnet Host Address Pool**
- **Switch - Host-ID = Network-ID /0000:00 -h**
- **Switch - Network-ID /0000:00 +h - Network Host Address Pool**

In other words, Activation of the Host Address Pool by Encapsulating the Host IP Address (or Subnet IP Address) with the Network-ID (Network IP Address), designs a method for every Address Class Network IP Address with the ability to uniquely identify and Activate every Host-ID from a Shared Host IP Address Pool.

Nevertheless, from the Mathematics of Quantification, in which derivation of the "Distributive Law of the Binary System" (The Distributive Law for Exponential Functions) resulted from the use of "Pythagoras Theorem" to proved the conjecture involving "Fermat's Last Theorem" is true, also changes, and redefines, the resulting IPtX Address Pool equation; in a table of comparisons given below;

eq-1.1: IPtX $= X[(2^{32}) + (2^{32})16{,}900{,}000] =$ Infinity

eq-2.1: IPv6 $= X(2^{32}) = (2^{128}) = (256^{16}) = 3.4028237 \times 10^{38}$ – Address Pool Total

eq-3.1: IPt4 $= X[(2^{32}) + (2^{32})16{,}900{,}000$ Host Addresses]

$= (2^{96})(2^{32}) + (2^{96})(2^{32})16{,}900{,}000$ Host-ID]

$= (2^{96})(2^{32}) + (2^{96})(2^{32})16{,}900{,}000$ Host-ID]

$= (2^{96})(2^{32}) + (2^{96})(2^{32})16{,}900{,}000$ Host-ID]

$= 3.4028237 \times 10^{38} + 5.7507720 \times 10^{45}$ Host Addresses

$= (2^{96})(2^{32}) + (2^{96})(2^{32})16{,}900{,}000$ Host-ID]

$= 5.7507723 \times 10^{45}$ IP Address Pool Total

$= X(Y + Y) = XY + XY$ - The Distributive Law

eq-4.1: IPv4 $= X(2^{32})$; and since $X = 1$, then;

eq-4.2: IPt1 $= [(2^{32}) + (2^{32})16{,}900{,}000$ Host-ID]

$= (2^{32}) + 7.2584947 \times 10^{16} = 7.2584952 \times 10^{16}$ – Address Pool Total

eq-5.1: IPt2 $= (2^{32})\ [(2^{32}) + (2^{32})16{,}900{,}000$ Host Addresses]

$= 3.1174999 \times 10^{26}$ – Address Pool Total

And more importantly, the flexibility of the Switch implemented for the "CIDR Network Descriptor" allows the possibility of incorporating the definitions of every 'Router Protocol' into '1' functional Protocol Specification: e.g. - IGMP, ICMP, RARP, TOP, All Query Messages, Redirects, Errors, and Router Solicitation and Queries; etc.

Note: Because the "DCE" Unit of the "IPtX-MX Protocol" can also act as the Bit-Mapped Translation of a 'Carrier Wave', it can also be assigned as the 'Streaming Voice Transport of an Analog Signal' (or a Synchronized Audio Video Wave). And this would provide the means to identify a true Universal Internet-Tel (IP Telephone) Specification, which would, using the 'CIDR Network Descriptor', eliminate the need for Voice and Data Signal Filtering. And more importantly, the Omni Directional Non Linear 3-D Space Cloud, it should be noted, can accommodate Multiple Bit-Mapped Address length Specifications; multiple (or Multiple Bands & Different Band Widths) Bit-Mapped Address length Specifications of the Intelligent Quantum Worm Protocol(s) {See - TABLE XV} :

1) allowing a direct Bit-Map Specification equating "e911" and "911" dialing.

2) Establishes the a 3-D Spatial Grid for the Internet, mimicking a GPS 'like' Mathematical Coordinate System, which provides the ability to Triangulate, using Trigonometry, the Location or Destination of any Internet Connection; Cellar Phone(s) included.

3) Emergence Broadcast Beacons - Seismic Monitoring

4) By Identifying every Node or Internet Connection, every Node becomes a Location Broadcast Beacon; or 'LBGNS' - Land Based Global Navigation System Connection, in the Real Time Environment of a 3-D Space. {This requires Network Synchronization: to Locate and establish Permanent Connections; to Locate and establish Roaming Connections with a 'Location Roaming History', which must also have an 'Established Permanent Connection Location Record' - 'No User Control or Access' with / to; the 'Zone IP', 'IP Area Code', or the 'CIDR Network Descriptor'. [10] - The IPtX DNS Specification)

6.3 Special IANA Consideration

- Current Definition(s) for the Measurement of the Bit

{Information provided as a Courtesy of "Wikipedia, the free Encyclopedia"}

Currently defined mathematical relationship(s) defining the Unit Bit:

Binary Digit –

Claude E. Shannon first used the word bit in a 1948 paper. He attributed its origin to John W. Tukey, who had written a Bell Labs memo in 9 January 1947 in which he contracted "binary digit" to simply "bit", forming a portmanteau. Interestingly, Vannevar Bush had written in 1936 of "bits of information" that could be stored on the punch cards used in the mechanical computers of that time. A bit is like a light switch; it can be either on or off. A single bit is a one or a zero, a true or a false, a "flag" which is "on" or "off", or in general, the quantity of information required to distinguish two mutually exclusive states from each other. The bit is the smallest unit of storage currently used in computing.

Unit Bit –

The bit, as a unit of information, is the amount of information carried by a choice between two equally likely outcomes. It is the capacity of one binary digit. One bit corresponds to about 0.693 nats ($\ln(2)$), or 0.301 hartleys ($\log_{10}(2)$). The name bit is mostly used when discussing data capacity, emphasizing the storage of data as individual binary digits. The name "Shannon", referring to the same unit, is mostly used when discussing information content, emphasizing aggregate information quantity.

A bit refers to a digit in the binary numeral system (base 2). For example, the number 1001011 is 7 bits long. Binary digits are almost always used as the basic unit of information storage and communication in digital computing and digital information theory. Information theory also often uses the natural digit, called either a 'nit' or a 'nat'. Quantum computing also uses 'qubits', a single piece of information with a probability of being true.

The bit is also a unit of measurement, the information capacity of one binary digit. It has the symbol bit, and less formally b (see discussion below). The unit is also known as the 'shannon', with symbol 'Sh'.

Ban –

A ban, sometimes called a ‘hartley’ (symbol Hart), is a logarithmic unit, which measures information or entropy, based on base 10 logarithms and powers of 10, rather than the powers of 2 and base 2 logarithms, which define the bit. Like a bit corresponds to a binary digit, a ban is a decimal digit. A ‘deciban’ is one tenth of a ‘ban’. One ‘ban’ corresponds to about 3.32 bits (log2(10)), or 2.30 ‘nats’ (ln(10)). A deciban is about 0.33 bits.

Nat –

A nat (sometimes also nit or even nepit) is a logarithmic unit of information or entropy, based on natural logarithms and powers of e, rather than the powers of 2 and base 2 logarithms which define the bit. The nat is the natural unit for information entropy, corresponding to Boltzmann's constant for thermodynamic entropy. When the Shannon entropy is written using a natural logarithm,

$$\boxed{H = -\sum_i p_i \ln p_i}$$

it is implicitly giving a number measured in nats. One nat corresponds to about 1.44 bits ($\log_2(e)$), or 0.434 hartleys ($\log_{10}(e)$).

TABLE - XVII

Quantities of Bits

SI prefix and (rare binary meaning)

Name	Symbol	Quantity
kilobit	kb	10^3 (2^{10})
megabit	Mb	10^6 (2^{20})
gigabit	Gb	10^9 (2^{30})
terabit	Tb	10^{12} (2^{40})
petabit	Pb	10^{15} (2^{50})
exabit	Eb	10^{18} (2^{60})
zettabit	Zb	10^{21} (2^{70})
yottabit	Yb	10^{24} (2^{80})

Binary prefix standards from IEC 60027-2

Name	Symbol	Quantity
kibibit	Kibit	(2^{10})
mebibit	Mibit	(2^{20})
gibibit	Gibit	(2^{30})
tebibit	Tibit	(2^{40})
pebibit	Pibit	(2^{50})
exbibit	Eibit	(2^{60})
zebibit	Zibit	(2^{70})
yobibit	Yibit	(2^{80})

IEEE 1541 is a standard issued by the Institute of Electrical and Electronics Engineers (IEEE) concerning the use of prefixes for binary multiples of units of measurement related to digital electronics and computing.

While the International System of Units (SI) defines multiples (and submultiples) based on powers of ten (like 10^3, 10^6, etc.), in computing multiples based on powers of two (like 2^{10}, 2^{20}, etc.) have been usually preferred. In the early times, this choice was made due to the intrinsic binary nature of computers, and often of computer equipment (such as RAM chips), considering that the error between 2^{10} = 1024 and 10^3 = 1000 was small enough to favor binary multiples. Thus, SI prefixes, such as kilo- (k, usually misspelled as K), mega- (M) and so on, have been used to indicate binary multiples in computer-related quantities that are not SI quantities. Moreover, there is not a consistent use of the symbols to indicate quantities such as bits and bytes. IEEE 1541 sets new recommendations to represent those quantities and units unambiguously.

After a trial period of two years, in 2005 IEEE 1541-2002 has been elevated to a full-use standard by the IEEE Standards Association, and it is now scheduled for maintenance in 2007.

Special Note: Clearly, the profound interpretation underpinning the foundational theory for the IPtX Protocol Specification, which transcends the prescribed purpose defining its application, redefines the:

- Electromagnetic Scale – providing Precision Tuning

- Quantum Scale Theory – Changing the Propagation Frequency

- Quantum Scale Theory – Changing the Radiation Frequency

- Quantum Scale Theory – Changing the Energy Mass Relationship

- Electromagnetic Scale – New Physics - Quantum Scale Theory

- Quantum Scale Theory – Work Energy Relationship Redefined

- Electromagnetic Scale – Defining the Frequency of an IP Address

- Quantum Scale Theory – Resolving Issues from Radiation Exposure

- Quantum Scale Theory – A New Energy Cache Defined

- Quantum Scale Theory – Defining the Electron Bit Relationship

- Quantum Scale – Mathematics of the Electromagnetic Spectrum

Furthermore, it is of special importance to mention that IEEE, specification 1541-2002, did not make any reference to the possibility of a Bit-Map association with an Electron - And this it can be said, is probably due, in part(s); to the inability to resolve a mathematical relationship associating the Unit Bit with the measurement of the Frequencies defined by the Electromagnetic Spectrum; the failure to understand the Electronic States of Matter; or the failure to understand the mathematics of Exponential (Binary Base 2) Enumeration and the respective Logarithmic Translations resolving an Irrational Exponent. And Tables XVIII and XIX, which Obsoletes IEEE Specification 1541-2002, substantiates the realization of this fact, by rendering the Binary Equivalent Conversion for the " SI Units " and the 'Electron Bit Association' - given by;

TABLE XVIII

Well - How should the " SI Units " be Defined...?

[Especially since, when using the Binary System, I have to define the Count of the Number of Digits the actual Number contains... just to Define the Number having a Binary Translation, which is equal to the Bit-Mapped Length!]

2^F = 1,000 = 10^3 = 10E3 = 2EF = Kilobit

2^F = 1,000,000 = 10^6 = 10E6 = 2EF = Megabit

2^F = 1,000,000,000 = 10^9 = 10E9 = 2EF = Gigabit

2^F = 1,000,000,000,000 = 10^{12} = 10E12 = 2EF = Terabit

2^F = 1,000,000,000,000,000 = 10^{15} = 10E15 = 2EF = Petabit

2^F = 1,000,000,000,000,000,000 = 10^{18} = 10E18 = 2EF = Exabit

[Given that: E = Exponential Operator; F = Variable Irrational Number; and X = Any Variable defined as a Member of the Real Number Set - And Look... ! IPv6 = 2^{128} ~ 3.40282367 x 10^{38}]

TABLE XIX – The 'Electron Bit Association'

The Binary Translation/Interpretation of the Electromagnetic Spectrum

```
+-----------------------------------------------------------------+
  Note: And now, the Binary Translation, as defined by the Logic from
        the Mathematics of Quantification, maintains that:

                  "If 2 Electrons or 2eV = 1 Byte, then;

                              2 Bits  =  1 Byte

                        2 Electrons = 2eV = 1 Byte

                        2.X  = Analog Signal Frequency

              2X [eV] = One Electromagnetic Spectral Frequency

                         {1eV = 1 Electron Volt}"

                     - IANA/FCC/IEEE Specifications -


        And this is a valid conclusion, especially since using the
        Logarithmic Translations to resolve the respective Irrational
        Exponent; if the value of the Unit Measurement of any
        Frequency is equal to '1 eV', and  2.X = 20  = 1 is true,
        then '  2.X[eV] ' defines {equals the Bit Count} a Neutral
        result, which does not change or effect the current value
        for the measurement of any Frequency defined by the
        Electromagnetic Spectrum. (as defined by the Substitution
        Law for Equality)
```

And more importantly, the greatest advantage of the IPtX-MX Protocol is that, it redefines the Internet Backbone for the IPtX Address Space, which uses the Addressing Overlay from the IPv4 Specification to Map and Connect the Front-End-Nodes distributed throughout the Address Space of the Backbone Environment- Thus allowing for;

1) **Creating a Free Public Internet Access Address Space -**

   a) **32 Bit - Intelligent Quantum Tunneling Worm Protocol**

   b) **256 Copies of the 2^{128} Bit Address Pool in IPt1**
      - Free Access to e911 and emergency eTelephony
      - Free Access to Local eTelephony and Basic Internet Services {The entire World Population)
      - No 32/64 Bit Information Exchange Control

2) **Universally Shared Internet Backbone**

   a) **Distinguishing the Commercial, Public, and Private Sectors Users**

   b) **Defined by the Configuration of the "CIDR Network Descriptor" for either the 32 or 64 Bit Specification for the Network Card (NIC)**

   c) **Specifying the IPtX-MX Protocol for either a 32 or 64 Bit Environment**

   d) **Specifying any one of the IPtX-MX 'Polymorph' Protocol designs for Exclusive and Non-Public use, and defining the Non-Polymorph Specification for Free 32 Bit Services Access**

3) **'Exclusive' Public Address Space - 64-Bit**

   a) **64 Bit - Intelligent Quantum Tunneling Worm Protocol**

   b) **256 Copies of the $2^{2,147,483,648}$ Bit Address Pool in IPt2**
      - Unlimited Access to eTelephony and Internet Services

   c) **64 Bit - 'Semi-Private' Network Backbone Address Space**
      [A 32/64 Bit Information Exchange Backbone Environment]

4) **Commercial Address Space - 64-Bit**

a) **64 Bit - Intelligent Quantum Tunneling Worm Protocol**

b) **256 Copies of the $2^{2,147,483,648}$ Bit Address Pool in IPt2**
- Unlimited Access to eTelephony and Internet Services

c) **64 Bit - 'Semi-Private' Network Backbone Address Space**
[The Contact Point for Customers and/or Clients; 32/64 Bit Xchg]

5) **"Non-Commercial and Non-Public" - Private Address Space**

a) **32 Bit - 'Secure Exclusive' Private Address Space**
- Controlled 32 and Limited-64 Bit Information Exchange with a 32/64 Bit Network
- the continuous change or rotation of the Masking and UN-Masking Equation(s) use with the Polymorph Intelligent Quantum Tunneling Worm Protocol
- Unlimited eTelephony and Internet Access

b) **64 Bit - 'Secure Exclusive' Private Address Space**
- Controlled 32 and Limited-64 Bit Information Exchange with any 32/64 Bit Network
- the continuous change or rotation of the Masking and UN-Masking Equation(s) use with the Polymorph Intelligent Quantum Tunneling Worm Protocol
- Unlimited eTelephony and Internet Access

```
Note: The OCTET, or 8 Bit Binary Numbering Sequence that
      defines the Numbering Format for an IP Address,
      actually, never has to Change from the current
      pattern - given by;
```

```
32 Bit-Mapped Displacement = 0000:00E0000.0000...

0000:00E0000.0000... = 256:00E0000.0000...

256:00E0000.0000... = 256:2E128.0000...
[ Where 2E128 = XXX : XXX : XXX . XXX . XXX . XXX ]

256:2E128.0000... = 256:{XXX:XXX:XXX.XXX.XXX.XXX}.0000...

XXX:XXX:XXX.XXX.XXX.XXX  =  Any IP Address  = XXXXXXXXXXXXXXXXXX

Any Number (IP Addressing Scheme Range) = ' XXXXXXXXXXXXXXXXXX '
```

- OR -

64 Bit-Mapped Displacement = 0000:00E0000.0000...

0000:00E0000.0000... = 256:00E0000.0000...

256:00E0000.0000... = 256:2E2,147,483,648.0000...

256:00E0000.0000... = 256:$2^{2,147,483,648\ .\ 0000...}$

[Where 2E2,147,483,648 = XXX : XXX : XXX . XXX . XXX . XXX]

256:2E2,147,483,648.0000... = 256:{XXX:XXX:XXX.XXX.XXX.XXX}.0000...

2E2,147,483,648 = $2^{2,147,483,648}$ = ' XXXXXXXXXXXXXXXXXX '

XXXXXXXXXXXXXXXXXX = Any IP Address = 123123123123123123 (e.g.)

Any Number (IP Addressing Scheme Range) = ' XXXXXXXXXXXXXXXXXX '

- **AND (See References; [5], [10])** –

```
[Eample: IPtX IP Address - 64 thru 80 Bits ~ 24 or 30 Digit Number]


                              3 State CIDR Network Descriptor
                              8 Bit -  Switch {'Ø','+','-', '/'}
                             [Where 'Ø' means "No Sign" or '/']
                                              \   /
                                               \ /
                      THE END-NODE OR FRONT-END | Network ID
                           |   Network IP    |    |  8 Bits
                           |     Address     |    |    |   Network
       8 Bit               |                 |    |    |  Octet ID
  ZONE IP ADDRESS          | 4 8-Bit Octets       |    | 5 or 8 Bits
             \             |/    \     /    \|    |    |     /
  '2EX' = [ XXX : XXX : XXX . XXX . XXX . XXX '+' /0000:00 ]
                       \                        /      |    \
       8 Bit - IP AREA CODE ADDRESS            /       |     \
                                              /      / \      \
                     16 thru 32 Bits  -  'CIDR Network Descriptor'


("AS-RIP" Protocol: 8 Bits = '/'; 16 Bits = '+/'; 16 Bits = '-/':
 "IANA/EMGNCY" IP Address Pool Total = 0000:00E0000.0000... =
 256:2EX.0000... ~ 256(1.1038234 x 10E12) ~ 256(2^40 + 2^32 + 2^24 )
 - Where [Figure 4] 'X' equals any variable defined by the Range
   of an 8 Bit Octet.)
```

[So... What would the IPt4/IPt4-MX Protocol look Like...? Well-?]

```
                                       78 Bit = 2^78
       2^8 = 8 Bit Prefix            Exponent
                 \                        /
                  \  2^2 = 2 Bit Base    /
                   \          |         /
                   0000  :  00   E  0000  .  0000...
                                 |                \
         2^8 = 8 Bit Exponential Operator          \
                                                     \
                   2^32  = 32 Bit Decimal String Accuracy
```

```
[Note: The accuracy of the Decimal String does not have to change until the
       displacement between any '2' Consecutive 'Integral Values' for the
       Exponent becomes greater; i.e. more Irrational Exponents between any
       '2' Consecutive 'Integral Values' than the Bit-Mapped Accuracy of the
       Decimal String. However, using the pattern for the "Class System",
       which is a technique defined by the 'CIDR Network Descriptor',
       minimizes this growth by Enhancing the Uniqueness of every Number
       representing an IP Address - And this prevents the Any LOSS in the
       Bit-Mapped Accuracy of the Decimal String [FIGURE 5 : 'IPt1'].
       Nevertheless, to maintain a mathematically consistent growth pattern
       throughout the IPtX Specification, use the IPt1 Specification
       [page 74] as the Baseline Guide. Where by, the 8 Bit Prefix, the 2
       Bit Base, and the 8 Bit Exponential Operator, are the values that are
       consistently maintained throughout the IPtX Specification - Given
       that, for every Sequential change in the IPtX IP Addressing
       Specification; (e.g. IPt2, IPt3, ..., IPt50, etc.) an additional
       24 Bits is added to the Exponent, and an additional 8 Bits is added
       to enhance the accuracy of the Exponential Decimal String.]
```

IPt4/IPt4-MX Protocol - 128 Bit-Mapped Displacement Length

```
128 Bit = 0000:00E0000.0000...

        = 256: 4 E 302,231,454,903,657,293,676,544

        = 256: 2 E 604,462,909,807,314,587,353,088.0000...

        = 604,462,909,807,314,587,353,088 Bit-Mapped Length

        = 75,557,863,725,914,323,419,136 Octets

        = 226,673,591,177,742,970,257,408 Digit Number!
```

- **And with the 8 Bit Prefix -**

['The IPt4/IPt4-MX IP Address Bit-Mapped Displacement is greater than the Bit Length of 100,000,000,000,000,000,000 Bit-Mapped (One Hundred Million Billion BITs; 1.00×10^{20} : [Figure 3] IP Address).']

IPt4/IPt4-MX Protocol - 128 Bit = 0000:00E0000.0000...

0000:00E0000.0000... = 256(226,673,591,177,742,970,257,408)

= 58,028,439,341,502,200,385,896,448 Digit Number!]

[Special Note: The IPv4 Overlay IP Addressing Scheme (Defining the "Front-End 'Only' Protocol Format), when using the appropriate Mathematical Factor for 'Masking and UnMasking' an IP Address, works quite well when using any one of the 'Polymorph Protocols' defined by the IPtX / IPtX-MX Protocol Specification. However, applying the Octet Rules, which does not include the 'Prefix', allows only the Manipulation (Rearrangement) of the '4' Positions defining; the Base, 'E' (the Exponential Operator), the Exponent, and the Exponential Decimal String - limiting the Number of 'Polymorph Protocols' for every Member of the IPtX-MX IP Addressing Protocol Family.]

NOTE: Notwithstanding the comparable difficulty in trying to imagine the Size of a Set, which requires a 1 Billion Digit Number to represent the Count of the Members it Contains, as compared to the Number representing 1 Billion, which defines a 10 Digit Number (1,000,000,000). However, to imagine, or garner a realistic perspective of the Size representing the Displacement of only One IPt4/IPt4-MX Bit-Mapped IP Address. Use the Current Hard Drive Capacity Specification (e.g. Seagate's 750GB Hard Drive) to determine how many 'Hard Drives' would be required to 'Write', or 'Store', 'Only One IP Address' defined by IPt4/IPt4-MX Protocol, when the IP Address Length is '604,462,909,807,314,587,353,088' Bits. In which case, 'IF you Cannot' use the "Intelligent Quantum Tunneling Worm Protocol", you'll need the 'Storage Capacity' of about 805,950,546,410 '750GB' Hard Drives just to 'Write' the Bit-Mapped Displacement, or the Bit Length of one Un-Masked IPt4/IPt4-MX Protocol IP Address.

[Closing Note: It is important to mention that since every Octet represents a (maximum of) 3 Digit Number; where, for example, a 32 Bit IP Address contains 12 Digits. The actual number representing this 12 Digit string exceeds the value of the 'DCE' Unit (2E32, or 2^{32}) in the IPtX/IPtX-MX Specification. In other words, the IP Address must be Converted to the Number equaling the Bit-Mapped String representing the IP Address: where the SUM or Joining of the individual Octets, is equal to the IP Address's Bit-Map Length:

256.256.256.256 = 256^4 = 4,294,967,296 = 32 Bit Length
= (11111111 + 11111111 + 11111111 + 11111111) = 2^{32}
= 2E32 = "DCE" Unit: '<u>Multiplication is the Quantified Sum of Addition</u>' [11]

Note: Joining the Bit-Mapped String of an IP Address requires computing the Number equaling Bit Count defining the IP Address, which also equals the Exponent, 'X', in 2EX; and this procedure sustains uniqueness and prevents duplicating the Octet 'SUM', which represents an IP Address.

Given that, when Sequential Counting represents an IP Address having Leading Zero(s); e.g.:

000.000.000.001 thru 000.256.256.256, or

000.000.000.00X thru 000.XXX.XXX.XXX;

the CIDR Network Descriptor equals;

/ØØØØ: ØØ, or "/E" - An all Zero representation having No Identifying Network ID, or Octet ID.

And when the IP Address is represented by:

256.256.256.256 or XXX.XXX.XXX.XXX,

the CIDR Network Descriptor equals; /XXXX:XX ; and if it is a Transmitted Electronic Signal, then; /0000:XX = /E:XX - when, 0000 = 00 = 1 and if ':XX' = 00 = 1, then 'Octet ID = ClassID' = A = /E:XX = /E:A
- 'Routed or Routing Protocol Specification'
In this case, as shown, the Network ID and the Octet ID are Identified.

In Both Cases, noted above - Giving the Right to Left, 8 Bit, Bit-Mapped Displacement of the IP Address, which can easily be defined by the Masking/Unmasking Software.

Note: Sequential Counting does not exceed the value of the 'DCE' Unit specified by the IPtX Specification. However, since defining the Bit-Map of an IP Address is Required, Bit Counting produces the Decimal String in the Exponent; given that 0E0 = 0^0 = 0 = E: i.e. when '0E0 = E', then ' E ' = 'Empty'; and ' 2E0 = 1 = 2^0 = 2E ': where by, ' E ' has 3 Definitions and Uses; the Binary Set, {0,1}, (Non Zero) 'Prefix' Signal Transmission, or 'E:' = 00 = 1; the 'Exponential Operator' in the "DCE " Unit, or E = 0E0 = 'Empty'; and the 'CIDR Network Descriptor', or /0000:00 = '/E'. {The IP Address Pool Total, as defined by the 'DCE Unit' in the IPtX-MX Specification, Excludes the Prefix and the Exponential Decimal String from the Calculation of the Address Pool Total..}]

```
   -    Example: Masking and Un-Masking Procedure    -
             ( 'ONLY' Network IP Address used )

1)  End-Node - Node Location / Front-End - Network IP Address
    = 211:002 : " 256.256.256.256 "  -/1111:32

2)  Bit-Map Base 2 Exponential Conversion -
    = 256.256.256.256 = 256^4  = 2^32
    = 4 , 294 , 967 , 296
    = 11111111 . 11111111 . 11111111 . 11111111

3)  IPtX-MX DCE Unit Conversion - 'DCE' Unit = 2E128
    = 11111111 . 11111111 . 11111111 . 11111111 = 2E32
    - Optional 'Zone IP' and 'IP Area Code' Bit-Mapped Sum  = 2E48
    - Additional;'CIDR Network Descriptor' Bit-Mapped Sum ~ 2E80

4)  IPtX-MX IP Address - IPt1-MX Protocol
    = 0000: 00 E 0000  . 0000... = 32 Bit-Mapped Length
    = 0001:  2 E   32  . ( Exponential Decimal String and/or + Pad )

5)  Bit-Mapped IPt1-MX IP Address -
    = 0001 :  2         E     32  . ( + Pad )
    = 0000 : 01 01000101 11111    .  ++++...

6)  Binary Transmission Signal (Prefix) Conversion -
    = 0000 : 01 01000101  11111   .  ++++...
    =     E :  2         E      32  . ( + Pad )

7)  Bit-Map Binary Transmission Signal -
    =         E :  2         E     32  .  ( + Pad )
    = 01000101 : 01 01000101 11111  .  ++++...
```

```
- Where; 2 = 01, E = 01000101, 32 = 2^5  = 11111, ++++... = Pad
- Masking Steps; 1) thru 7)  -  Un-Masking Steps; 7) thru 1)
- Noting that the "Mask/UnMask" Software can distinguish the;

   - Prefix; 0000: = 00 = 'E:'
   - DCE Unit; 0E0 = 0^0  = 0 = 'E' = 'Empty'
     where OEX = Emergency Broadcast = BEX -'EBoIP' See 'GWEBS'  [8]
     and -  2E0 = 2^0  = 1 = '2E' : where AM/FM Radio, and Television;
      - AM/FM Radio Broadcast = BRoIP = 0000:BR0000.0000... = 32 Bit
      - AM/FM Radio Broadcast = BRoIP = 0000:BR0000.0000... = 64 Bit
      - Television Broadcast = BToIP = 0000:BT0000.0000... 32/64 Bit
   - CIDR Network Descriptor; /0000:00 = '/E'
   - Seperating their Individual "Mask and UnMask" Procedures
```

```
   e.g., the IPt1 / IPt1-MX Emergency Broadcast Protocol,
  'BEX'= DCE Unit: 0000:BE0000.0000... = 32 Bit, is given by;
                                2^4 = 4 Bit Integer
  2^8  = 8 Bit Prefix                        /
              \   2^8  = 8 Bit Broadcast
               \        |             /
                0000  :  B   E   0000  .  0000...
                             |                \
          2^8  = 8 Bit Emergency Operator       \
                                                 \
                                   2^4 = 4 Bit Decimal String

'Emergency Broadcast Protocol' = e911 = 256:BE9.11 = 32 Bits;
```

```
 Note: The Broadcast Protocol(s) discussed above are Channels
       that can be Transmitted, or 'Pushed' by any 'Carrier
       Wave' Frequency defined by the Electromagnetic Spectrum;
       within any given Hertz Range Specification.
```

- While the 'DCE' MUST Bit-Map the Octet Distribution, Joining the Bit String equaling the IP Address. The 'Exponential Decimal String' however, the 'Fraction or Decimal' part of the Exponent defined by the 'DCE' Unit, DOES NOT Follow the Octet Rules (Per se)- It Bit-Maps the Number(s) Sequentially; the ENTIRE Bit-Mapped Displacement Length that represents the Exponential Decimal String. For example: IPt1-MX - 8 Bit Exponential Decimal (where Significant Figures includes the 'Zeros' to the Right, after the 'Decimal Point'; defining an Accuracy for the Numeral representing the Decimal that depends upon the Bit Length of the Exponential Decimal String.);

  '0.11111111', '0.00000001', e.g.;

  '0.00001000' (Binary) = '0.000090' (Decimal Equivalent); or

  '0.1000' (Binary) = '0.90' (Decimal Equivalent)

  Where; '0.0' = '0' = 'Empty' - Hence; 2E'0.0' = 2E0 = '2E'

  $2^{0.0}$ = 2E0.0 = 2^{0} = 2E = 1

  Given that;

  '0.00000000' (Binary) = '0.00000001' (Decimal Equivalent) -

  Then;

```
   Binary             Transmission            Equivalent
   Bit-Map             Conversion               Decimal
  /   |   \           /     |     \                |
'0.0' = ' . '   =   '0.E' = '.E' = 'E'   =    '0.00000000'
```

Hence, the Bit-Mapped Displacement of the Decimal portion of the Result from a Base 2 Exponential Operation equals the Decimal String of the Decimal portion of the Numeral Equaling the Exponent in the Exponential Equation. Given that;

IF 'Z.Z.Z.Z = $2^{N.XXXX}$ ', then the Bit-Mapped Displacement of 'N.XXXX' is Equal to 'Z.Z.Z.Z' = Z^4 = XXX,XXX,XXX,XXX.

e.g.: Resolving the Bit-Mapped Displacement of the IP Address, '123.123.123.123', is given by;

= 123.123.123.123 = $(123)^4$

= $(123)^4$, = 2EQ = 228,886,641

= 2EQ = 228,886,641 = 2^Q

Q = 1101.10100100.10001000.01110000 ~ 28 Bits
= Bit-Mapped Displacement Length = 28 Bits

Log_2 228,886,641 = Q

Q ~ 27.770058024... ~ 27.77 (2 Digit Accuracy)

2EQ = $2^{27.770058024...}$ - Octet ID = ' :28 '

2 E 27.770058024...~ 228,886,641.4193240053 ~ 228,886,641.42

- Or -

2 E 27.77 ~ 228,886,641.42 ~ 228,886,641

However, if the Decimal String is the Result from a Base 2 Exponential Operation, then the Exponent, Q, is Negative: as given by;

2E-Q = 2$\bar{\text{E}}$Q (READ: Two Bar E Q) = '.xxxx' (Decimal Result).

[Hence; 2$\bar{\text{E}}$Q = 2E-Q = 2^{-Q} = 2$\bar{\text{E}}$Q (READ: Two Bar E Q ;
'Bar E' is used to Denote a Negative Exponent)]

Then the Logarithm of any Decimal Fraction, '.XXXX', to the Base 2, is given by;

Log_2 '.XXXX' = - Q , since Log_2 0.5 = - 1 and 2^{-1} = 0.5

({ Hence, Polymorphing the Basic IPtX-MX Protocol; is the Rearranging of the Components defining the 'DCE Unit': IFF, 'DCE Unit' ≠ 0E0
- IPtX-MX Protocol Specification. })

```
And recall, that if 2^0 = 1 and 2^1 = 2, then any value
assigned to the Exponent that is 'Less Than' 1, yields a
Result defining the Base 2 Exponential Equation, which is
said to Approach the Value of '1', as the value of the
Exponent approaches Zero.


( Hence, Polymorphing the Basic IPtX-MX Protocol; is the
  Rearranging of the Components defining the 'DCE Unit':
  'iff, DCE Not Equal 0E0' - IPtX-MX Protocol Specification. )

- See [page 87 - Eample: IPtX IP Address], [Page 74]
```

Additional Note: To Sustain the Users 32 Bit IP Addressing format definition when the Bit-Mapped Displacement for the actual IP Address defining the 'DCE Unit', exceeds 80 Bits, or Any Numbering Pattern of Choice; 'Think Binary 'DCE Unit' Conversion' - See [Page 86] e.g.:

DCE Unit = 2E500 = 2^{500}

2^{500} = 3.2733906078961418700131896968276 x 10E150.

In other words, while there might be several logically viable mathematical formulations, which can be used to Convert, or resolve the User's Network IP Address. The easiest method however, which follows the Octet Rule(s), requires performing the Operation of Addition and Subtraction in an Equation involving the Converted IP Address's Respective 'DCE Unit' configurations. That is, when the Total Number of Available IP Addresses is Greater, to Exhaust the Entire IP Address Pool Range defined by the 'DCE Unit' of the IPtX / IPtX-MX Specification. It only requires performing the Operation of of Addition and the Operation of Subtraction on the respective DCE Units; the order of the Operation(s) is determined not only by the Result, which Cannot exceed the DCE Unit's Address Pool Total*, but the Transmission Direction as well - e.g.:

11111111 . 11111111 . 11111111 . 11111111

= 32 Bit-Mapped Displacement Length = DCE Unit = 2E32

11111111111111111111111111111111 . 11111111111111111111111111111111
.11111111111111111111111111111111 . 11111111111111111111111111111111

= 128 Bit-Mapped Displacement Length = DCE Unit = 2E128

Where each Octet has 4 Times the Number of Bits, yields;

4 x (11111111) = 11111111 + 11111111 + 11111111 + 11111111

= 11111111111111111111111111111111 = 32 Bits

The Masking / UnMasking Bit-Mapped relationship - given by;

1) Masking; 2EX (DCE Unit) + 2E32 (DCE Unit) = 2E128 ; Union Octets

2) UnMasking; 2E128 (DCE Unit) - 2EX = 2E32 ; Dis-Union Octets

where; 2^{32} = IPv4 Overlay assigned Users Network IP Address

and; 2^{128} = 2E128 = DCE Unit's IP Address Pool Total*;
in which the IP Address specified by the Range of the IPv4
Overlay, or the 48 Bit, 18 Digit specified by the IPtX IP
Address Specification Cannot exceed the Address Range Limits
imposed by the 'DCE Unit' Bit-Mapped Specification.

Note: Given that; "Multiplication is the Quantified Sum of Addition" [2.]
The Rudiments of Finite Algebra, [11] - we have:

1) Joining the Octet String of an IP Address represents Multiplication - the Product of the Numerical values defining every Octet in the IP Address - where by;

$128.002.128.002 = 128 \times 2 \times 128 \times 2 = 2E16 = 2^{16}$

2) Sum of the Octet String of an IP Address represents Addition - the Sum of the Bit-Mapped Lengths defining every Octet in the IP Address - where by;

01111111.00000001.01111111.00000001 =
01111111 + 00000001 + 01111111 + 00000001 =
1111111111111111 = 2E16 = 2^{16} = 65,536

3) Hence; 128.002.128.002 = 1111111111111111 = 65,536

Even still, the foregoing conclusions, while true (The Commutative Law), clearly demonstrates the reality of a Mathematical Anomaly. That is, it emphases that which is clearly evident: "if the Numerical Representation for an IP Address is Unique [page 33] - dialing a IP Telephone Number), then every Digit in the IP Address Must Be Significant (Closure Law(s))" -

128.002.128.002 - Is Not Equal To - 128 x 2 x 128 x 2

128.002.128.002 = 128.002.128.002

Especially since, the Equations below must also be True;

$128 \times 2 \times 128 \times 2 = 128.002.128.002 = 1111111111111111 = 2E16 = 2^{16}$

$2 \times 128 \times 2 \times 128 = 002.128.002.128 = 1111111111111111 = 2E16 = 2^{16}$

$128 \times 2 \times 2 \times 128 = 128.002.002.128 = 1111111111111111 = 2E16 = 2^{16}$

$128 \times 128 \times 2 \times 2 = 128.128.002.002 = 1111111111111111 = 2E16 = 2^{16}$

$2 \times 128 \times 128 \times 2 = 002.128.128.002 = 1111111111111111 = 2E16 = 2^{16}$

$2 \times 2 \times 128 \times 128 = 002.002.128.128 = 1111111111111111 = 2E16 = 2^{16}$

In other words, the procedures noted above could Not serve as a viable means for resolving the Conversion for the 'IP Address - Bit-Map' Translation. However, since the 'DCE Unit" in the IPtX Specification, counts sequentially, beginning with '1'; and given that in IPt1-MX, the -

'DCE Unit' = 2E128 = 2^{128} ~ 3.403 x 10^{38}

= 2^{128} = 3.4028236692093846346337460743177 x 10E38

= 2^{128} ~ 340,282,366,920,938,463,463,374,607,431,768,211,456

(Representing More Than 300 Million Trillion Trillion IP Addresses - when Sequentially Counted)

The "Preferred Method", which is the easiest way to resolve the 'IP Address - Bit-Map' Translation, given that Sequential Counting used in the IPtX Specification complies with the "Octet Rules", equates the Numeral representing the IP Address to the value of the Result from a Base 2 Exponential Operation. Especially since;

128.002.128.002 = 128,002,128,002 = 2EF

002.128.002.128 = 2,128,002,128 = 2EF

128.002.002.128 = 128,002,002,128 = 2EF

128.128.002.002 = 128,128,002,002 = 2EF

002.128.128.002 = 2,128,128,002 = 2EF

002.002.128.128 = 2,002,128,128 = 2EF

- Noting specifically that the Numeral representing the IP Address, hence its Bit-Mapped Configuration, never changes; because this procedure emphasizes using the IP Address Numeral, with the "Dots" in the 'Dotted Notation Removed. ([Page 47 & 64] - IP Address = 2EX; as in Exponential Counting
- e.g. 2E0 = 1, 2E1 = 2)
- and using the Logarithm to Resolve the Exponent for the Equation's Result (the IP Address) completes the process.

- Clearly, this is an Important Step concluding the process which provides a 100% Backward Compatibility with the IPv4 Specification. In other words, because the current Computer Specification, hence Computer Programming, Bit-Maps using the Only the 'Binary Pair (Set)', {0,1}, 100% Compatibility could only be achieved if all the Procedures involving the current IP Address's associated Applications, Conventions, and Connection remained unchanged at the Operating System level - and this was in fact achieved, because everything can be Converted into the Base 2 Exponential Expression for Transmission on the Backbone (Support CHIPs / Firmware EEprom / NIC Driver Updates, e.g. / etc.) - e.g.;

256.256.256.256 = 2EX ;

End-Node Location
Bit-Mapped Translation

11111111.11111111.11111111.11111111 = 256.256.256.256

Backbone
Bit-Mapped Translation

256.256.256.256 = 00002EX0000...; 2EX = $2^{37.90010...}$

```
01000101 : 01 01000101 100110  .  1000 00   ...++++
```

Bit-Mapped IPt1-MX IP Address -

```
= 0001 :  2        E      37   .      9  00  1  0...+ Pad
= 0000 : 01 01000101 100110  .  1000     00    ...++++
```

Bit-Map Binary Transmission Signal -

```
=         E :  2        E      37   .    9  00  1 0...+ Pad
= 01000101 : 01 01000101 100110  . 1000     00   ...++++
```

Or -

254.254.254.254 = 2EX ; (Could be used in the Example)

The above conclusions are valid, because in both cases when using the IPtX/IPtX-MX Protocol Specification;

256.256.256.256 = 2EX < 2E128 = IPt1/IPt1-MX Protocol

Or -

The IPt1-MX Protocol could, if the Current Computer Specification used the Base 2 Exponential Representation, increase the Number of available IP Addresses in the Overlay for the IPv4 Specification - to;

256.256.256.256 = 2EX =

256 Billion, 256 Million, 256 Thousand, 256 (Hundred) IP Addresses

More than 256 Billion IP Addresses, compared to the 4 Billion IP Addresses in the IPv4 Specification

- and using the Logarithm to Resolve the Exponent for the Equation's Result (the IP Address) completes the process, which requires Minor Back-End Changes to implement, and 'Major' Back-End and Front-End Changes to fully exploit the Addressing capacity of the IPtX/IPtX-MX Specification.

- For Example; IPv4 cannot use the IP Address:

256.999.999.999 =

1110111101011001100100110010100000000 > 32 Bits

Or -

254.999.999.999 =

1110110101111100101111001101100000000 > 32 Bits

the Bit-Mapped Length is Greater Than 32 Bits, which demonstrates the difference between Bit-Mapping the Numeral and Bit-Mapping the Exponent equaling the Binary conversion representing the Numeral. In other words, these are Addressing Formats defining an IPtX-MX IP Address, which are defined as the Result from a Base 2 Exponential Mathematical Operation that uses (works with) the actual Number equaling the value of the Numeral representing an IP Address.

```
      Also - given that, in Reality;

    ftp://256.256.256.256   = 'Is Just a Binary Numeral'

8 Bits 8 Bits 8 Bits : 8 Bits 8 Bits 8 Bits. 8 Bits. 8 Bits. 8 Bits

    ftp://256.256.256.256   = 2EX ~ 72 Bits


      And in;

    http://www.name.com, where

    ' www ' = Zone IP + IP Area Code = '256 : 256' = 16 Bits

      In other words, since every IP, URL, or FTP Address must
      have a Binary Representation, which represents the
      Equivalent Binary Conversion of the Bit-Mapped Length
      for the Exponent in the Base 2 Equation defining the
      Numeral equaling the Address. Then every IP, URL, or FTP
      Address must have an Equivalent Numerical Representation,
      which can be Expressed as the Result from a Binary Base 2
      Exponential Operation. Clearly, changing only the Method
      of Masking, before Bit-Mapping the Data String for
      Transmission - The 'DCE Unit' carries a 'Payload' of
      compressed, or Encoded Data, which can be Un-compressed
      and used without Change - which is to say; Only the
      Method(s) for Data 'Handling' of the Data String has to
      change, and it must occur before Transmission -
```

However, it should be clearly noted, while 100% Backward Compatibility is factual, it also means inheriting all of the Flaws and Errors, which plagued the IPv4 Specification (See; Work(s) in Progress [11.]) - Inherent Foundational Flaws involving the 'Misinterpretation of Zero - Even still, if All of the Masking and Un-Masking Procedures were Preformed in the 'OSI and TCP/IP Layers', which is the Preferred Method (the Temporary Patch until Base 2 Binary System Conversion is Completed), as given below (See; [page 94 and 95] and [The TCP/IP Model for the 'IPtX Specification']); then the entire Address Range would be Available. The resulting exploitation would make it possible to use the entire IP Address Range defined by the 'DCE Unit'; the Hallmark of the IPtX Specification. And more importantly, this process would also provide IP Address and Bit-Mapped Length Control, effectively Hiding all Additional Bits beyond the assigned Network IP Address Space; while preventing User Access and Control. An important functional usage for vendors such as AT&T, and other Telephone Companies, which typically use a Telephone "Account" Numbering System for Billing purposes: A Numbering System Format that includes the User's Assigned Telephone Number in a Digital count that exceeds the Number of Digits assigned to the User's Telephone Number.

Furthermore, it should also be added; this is a benefit that can be adopted throughout the business community. That is, any Business seeking to enhance Control and Security of the User's Billing Account; e.g. the Cable TV and Satellite Broadcast Communication Companies, could use the same, or similar Numbering format to Enhance Security using this type of 'Account Personalization' - Clearly, enhancing the Security Protocols for the identification of the User and End-Node Location, could also provide Users with additional benefits - such as:

Enhanced Personalized Controls -

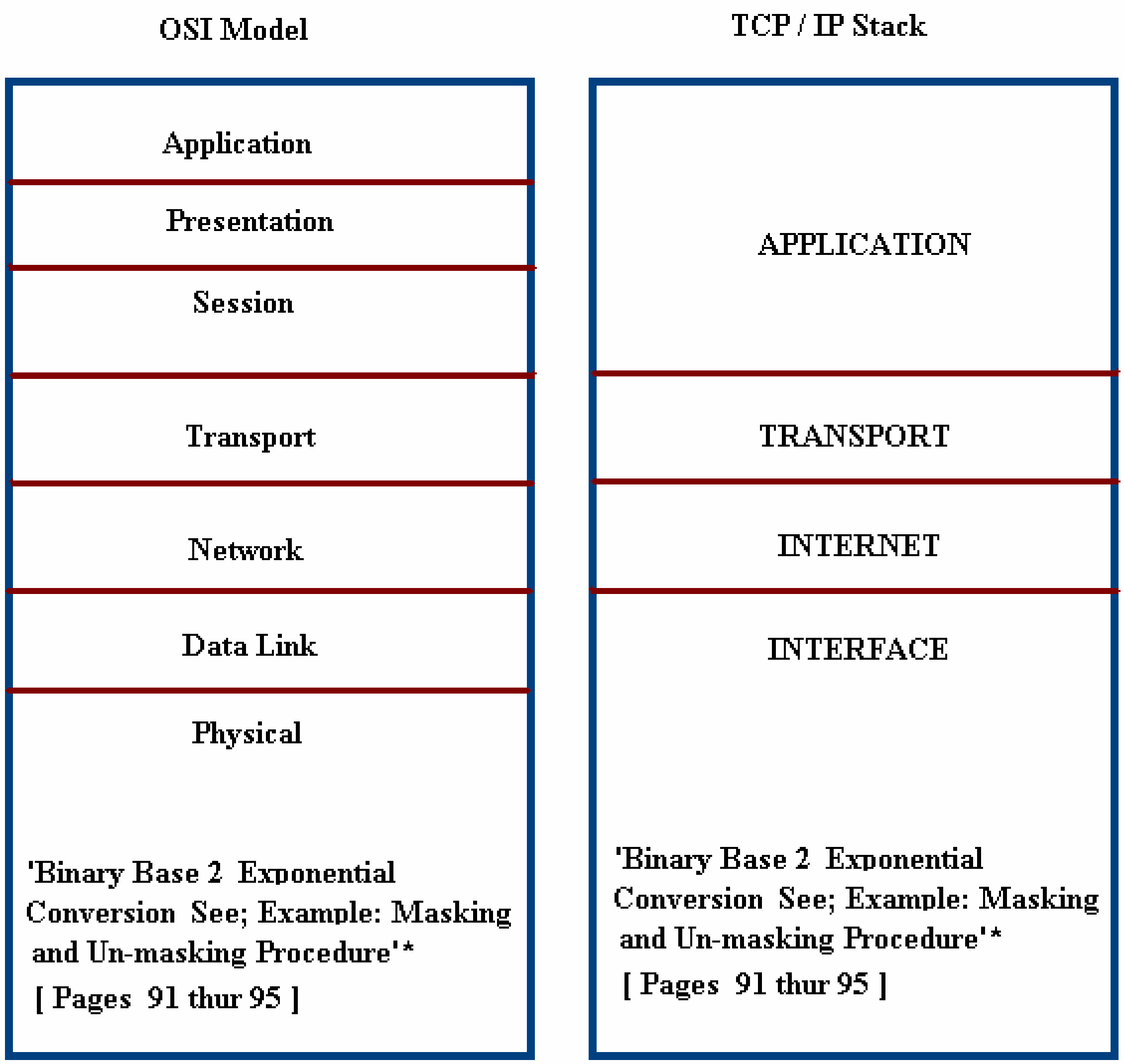
Enhanced Security -

Enhanced Data Transfer Rates over existing Lines, without Expensive Equipment, and / or Upgrade Cost, because this procedure does not impact, or affect any of the current standards involving Data Transfer -

Enhanced Interactive Personalized Entertainment (i.e., 'Broadcast Entertainment Internet Protocol': the 'BEoIP' Protocol) -

TABLE 1.a1

The TCP/IP Model for the 'IPtX Specification' 100% Backward Compatibility with the IPv4 Specification



Cost Effectiveness - Well? Weighting the Benefits of IPtX... It's Less Than IPv6!

Only the 'Physical Layer' in the "OSI Model', and the 'Interface Layer' in the TCP/IP Stack are required to Change - Minor when compared to Changing every Program associated with every Communications 'Layer' – Major Change is required to make everything Mathematically Compatible with Binary Exponential Base 2 System - Eliminating the 'HEX' System and Replacing it with 'The Binary Base 2 Exponential System of Counting'.

Note: Currently the 'Data Link' and 'Transport layers convert the Bits into Frames or Packets, moving the Frames between Connected Hosts;

Minimum packet size - 64 octets

Maximum packet size - 1518 octets

Gigabit Ethernet Jumbo Frames - 9018 Bytes ~ 9018 Octets

However, this Size limitation can be Eliminated, if the Data Stream used the Binary Base 2 Exponential System before the Bit-Mapped conversion is sent to the 'Data Link' and 'Transport layers -

And modification of the 'Session' and 'Transport' Layers would allow the 'Simulcast' of Video and Voice; 'VVoIP' -

- This procedure represents the 'First Stage', defining the Minimum Requirements, which must be Encoded prior to the implementation and / or use of the 'IPtX/IPtX-MX' Protocol Specification - which Complies with the requirement of 'RFC 1550' - "IP: Next Generation (IPng) White Paper Solicitation"; 100% Backward Compatibility with the 'IPv4 IP Addressing Specification'. Noting specifically that, all of the IP Addresses in the 'IPtX Specification' are Now Available. And incrementing using the Addition of "1's", to every IPv4 IP Address, following the Masking / Un-Masking procedures discussed above, yields approximately;

a. IPt1 = 256 x 2^{96} (Copies of 2^{32}); the IPv4 Specification)

b. IPt1 = 256 x 2^{32} (Copies of 2^{96}); the IPt1 Specification)

c. IPt1 = 256 Copies of 2^{128} ; the IPv6 Specification)

Note: It is behooving that consideration is given to Re-defining the Relationship between the 'Bit' and the 'Byte', as presented in; [TABLE XIX] – The 'Electron Bit Association'. Especially since, this would sustain the logical validity and the Mathematical continuity as defined by the 'Binary Base 2 Exponential Mathematical System'.

Especially since [page 97 thru 100];

IPv4 IP Address = 'ØØØ. ØØØ. ØØØ. ØØØ' = 2EX

'ØØØ. ØØØ. ØØØ. ØØØ' = 'XXX,XXX, XXX, XXX ' = 2EX;

and IPt1 = IPv4 = 32 Bit-Mapped Length -

Hence, without using the '3 State CIDR Network Descriptor';

IPt1 a. = IPt1 b. = IPt1 c. = 256 x 2^{128} IP Addresses

256 x 2^{128} = 2^{8} x 2^{128} = 2^{136} IP Addresses

IP Address Pool Total for the IPt1/IPt1-MX Speccification

= 2^{136} = 8.7112285931760246646623899502533 x 10E40 IP Addresses

However, when using the 3 State Logic of the CIDR Network Descriptor (While just a Number; there is a Difference);

a. /0000:00 = 8.7112285931760246646623899502533 x 10E40 Addresses

b. +/0000:00 = 8.7112285931760246646623899502533 x 10E40 Addresses

b. -/0000:00 = 8.7112285931760246646623899502533 x 10E40 Addresses

Note: 'Every Numeral in the Prefix', 256 - (1 thru 256); where '256' = '0000:', defines All 3 State(s), representing the 3 Types Switches defining the 'CIDR Network Descriptor'.

Therefore, the IP Address Pool Total for the IPt1/IPt1-MX Specification increases by a factor of 3 - where by;

'a.' + 'b.' + 'c.' = 2.613368577952807399398716985076 x 10E41

$= 3 \times 2^{136}$ = 3 x 8.7112285931760246646623899502533 x 10E40

= 2.613368577952807399398716985076 x 10E41 IP Addresses

Note: When the IP Address, instead of the Exponent, Equals the Result from a Base 2 Exponential Operation, the Bit length increases by a factor of 2E4; or approximately 16 Bits. [page 89] - [Example: IPtX IP Address - 64 thru 80 Bits ~ 24 or 30 Digit Number] Furthermore, when the IP Address is used in this 'way', the Bit-Mapped Length defining the IP Address, Bit-Maps to the Length corresponding to the value of the Exponent, which defines the actual, or "Numeric Face Value' of the Number equaling the Result from a Base 2 Exponential Operation. (i.e., it increases from '2E80' Bits to approximately '2E96' by representing the actual Number, and not the Bit-Mapped Length of the Octets equaling the Exponent. And this yields an Address Pool Total approximating '$2^{37...}$' = '2E37...' IP Addresses) - e.g.;

IPt1 / IPt1-MX = 256.256.256.256 ~ $2^{37...}$

= 256,256,256,256 ~ $2^{37...}$ IP Addresses

38 Bit Length = 11101110101010000100000010100100000001

Where the Exponent in '$2^{37...}$ ' ~ 38 Bits

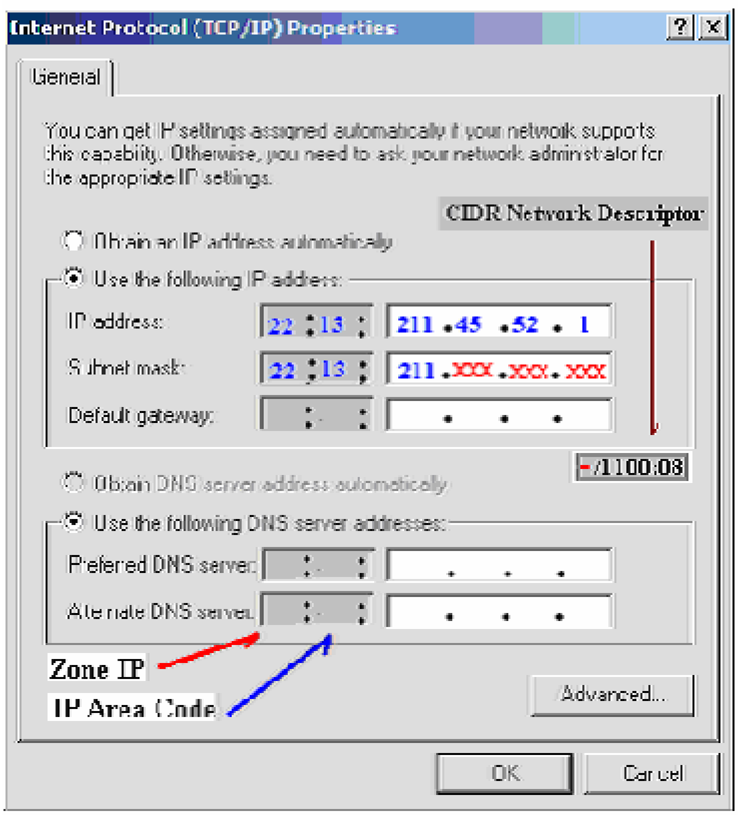
While in the IPv4 Specification - See [page 99];

256.256.256.256 = 1111111111111111111111111111111111111111

32 Bit Length = 1111111111111111111111111111111111111111

256.256.256.256 = 4,294,967,296 = 2^{32} IP Addresses

Example; Network Card Configuration



Note: Complying with the "Octet Rules" changes the results, as presented in Figure 7... Where by, the "IPtX-MX Protocol", at Infinity, as given by Figure 7a, becomes;

```
IPtX / IPtX-MX  Infinite                 IPtX / IPtX-MX Protocol
Bit-Mapped Displacement  =  0000:00E0000.0000...
   [ eq-f.  '∞' = (∞ + ∞)  =  Infinities, (∞ + ∞) = '∞'  =  Infinities ]

      8 Bit Prefix                 Infinite Bit Exponent
             \                            /
              \      2 Bit Base          /
               \          |             /
                0000  :  00    E   0000  .  0000...
                               |               \
            8 Bit Exponential Operator          \
                                                 \
                          Infinite Bit Exponential Decimal String
```

FIGURE 7a

Nevertheless, while providing IANA, FCC, and all other noted Regulatory Agencies [defined with FCC responsibilities within their respective ZONE IP location(s); e.g. IEEE], with an extremely broad range of decision options, concluding the overall general design and operational procedures for the IPtX Addressing Protocol Family Specification. [The Internet Protocol telecommunications Xchange Specification; IPtX, represents a design specification that can contain a Diverse Colony Population of "Intelligent Quantum Tunneling Worm Protocol" Specifications - i.e. any one or more members from the IPtX Addressing Protocol Family Specification can populate the Backbone Environment approaching an unlimited size 'Bit-Map' Address Space.] However, embedded within the context of this document are the Preferred, or Recommended Operational Procedures that mandates the continued existence of the mathematical continuity ascribed by the Hierarchy of the IPtX Specification, which assigns the controlling position of Addressing in the Address Space (containing the "Front-End" and the "Back-End") to the "Front-End".

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'

10. **http://www.ietf.org/internet-drafts/draft-terrell-iptx-dns-req-iptx-ip-add-spec-03.pdf**
(The development of the DNS {Domain Naming Specification} the for IPTX IPSpec)
- "Work(s) in Progress'

11. **http://www.ietf.org/internet-drafts/draft-terrell-math-quant -ternary-logic-of-binary-sys-05.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-15.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.)

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"}.

7. 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.**

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).