

CATNIP: Common Architecture for the Internet

[<draft-ietf-catnip-common-arch-00.txt>](#)

Status of this Memo

This document was submitted to the IETF IPng area in response to [RFC 1550](#). Publication of this document does not imply acceptance by the IPng area of any ideas expressed within. Comments should be submitted to the big-internet@muninari.oz.au mailing list.

Distribution of this memo is unlimited.

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a ``working draft'' or ``work in progress.''

Please check the `1id-abstracts.txt` listing contained in the internet-drafts Shadow Directories on nic.ddn.mil, nsc.nsf.net, nic.nordu.net, ftp.nisc.sri.com, or muninari.oz.au to learn the current status of any Internet Draft.

Executive Summary

This paper describes a common architecture for the network layer protocol. The Common Architecture for Next Generation Internet Protocol (CATNIP) provides a compressed form of the existing network layer protocols. Each compression is defined so that the resulting network protocol data units are identical in format. The fixed part of the compressed format is 16 bytes in length, and may often be the only part transmitted on the subnetwork.

With some attention paid to details, it is possible for a transport layer protocol (such as TCP) to operate properly with one end system using one network layer (e.g. IP version 4) and the other using some other network protocol, such as CLNP. Using the CATNIP definitions, all the existing transport layer protocols used on connectionless network services will operate over any existing network layer protocol.

The CATNIP uses cache handles to provide both rapid identification of the next hop in high performance routing as well as abbreviation of the network header by permitting the addresses to be omitted when a valid cache handle is available. The fixed part of the network layer header carries the cache handles.

The cache handles are either provided by feedback from the downstream router in response to offered traffic, or explicitly provided as part of the establishment of a circuit or flow through the network. When used for flows, the handle is the locally significant flow identifier.

When used for circuits, the handle is the layer 3 peer-to-peer logical channel identifier, and permits a full implementation of network-layer connection-oriented service if the routers along the path provide sufficient features. At the same time, the packet format of the connectionless service is retained, and hop by hop fully addressed datagrams can be used at the same time. Any intermediate model between the connection oriented and the connectionless service can thus be provided over cooperating routers.

CATNIP Objectives

The first objective of the CATNIP is a practical recognition of the existing state of internetworking, and an understanding that any approach must encompass the entire problem. While it is common in the IP Internet to dismiss CLNP with various amusing phrases, it is hardly realistic.

The CATNIP integrates CLNP, IP, and IPX. The CATNIP design provides for any of the transport layer protocols in use, for example TP4, CLTP, TCP, UDP, IPX and SPX to run over any of the network layer protocol formats: CLNP, IP (version 4), IPX, and the CATNIP.

Incremental Infrastructure Deployment

The best use of the CATNIP is to begin to build a common Internet infrastructure. The routers and other components of the common system are able to use a single consistent addressing method, and common terms of reference for other aspects of the system.

The CATNIP is designed to be incrementally deployable in the strong sense: you can plop a CATNIP system down in place of any existing network component and continue to operate normally with no reconfiguration. (Note: not "just a little". None at all. The number of "little changes" suggested by some proposals, and the utterly enormous amount of documentation, training, and administrative effort then required, astounds the present authors.) The vendors do all of the work.

There are also no external requirements; no "border routers", no

requirement that administrators apply specific restrictions to their network designs, define special tables, or add things to the DNS. When the end users and administrators fully understand the combined system, they will want to operate differently, but in no case will they be forced. Not even in small ways. Networks and end user organizations operate under sufficient constraints on deployment of systems anyway; they do not need a new network architecture adding to the difficulty.

Typically deployment will occur as part of normal upgrade revisions of software, and due to the "swamping" of the existing base as the network grows. (When the Internet grows by a factor of 5, at least 80% will then be "new" systems.) The users of the network may then take advantage of the new capabilities. Some of the performance improvements will be automatic, others may require some administrative understanding to get to the best performance level.

The CATNIP definitions provide stateless translation of network datagrams to and from CATNIP and, by implication, directly between the other network layer protocols. A CATNIP-capable system implementing the full set of definitions can interoperate with any existing protocol. Various subsets of the full capability may be provided by some vendors.

No Address Translation

Note that there is no "address translation" in the CATNIP specification. (While it may seem odd to state a negative objective, this is worth saying as people seem to assume the opposite.) There are no "mapping tables", no magic ways of digging translations out of the DNS or X.500, no routers looking up translations or asking other systems for them.

Addresses are modified with a simple algorithmic mapping, a mapping that is no more than using specific prefixes for IP and IPX addresses. Not a large set of prefixes; one prefix. The entire existing IP version 4 network is mapped with one prefix and the IPX global network with one other prefix. (The IP mapping does provide for future assignment of other IANA/IPv4 domains that are disjoint from the existing one.)

This means that there is no immediate effect on addresses embedded in higher level protocols.

Higher level protocols not using the full form (those native to IP and IPX) will eventually be extended to use the full addressing to extend their usability over all of the network layers.

No Legacy Systems

The CATNIP leaves no systems behind: with no reconfiguration, any system

presently capable of IP, CLNP, or IPX retains at least the connectivity it has now. With some administrative changes (such as assigning IPX domain addresses to some CLNP hosts for example) on other systems, unmodified systems may gain significant connectivity. IPX systems with registered network numbers may gain the most.

Limited Scope

The CATNIP defines a common network layer packet format and basic architecture. It intentionally does not specify ES-IS methods, routing, naming systems, autoconfiguration and other subjects not part of the core Internet wide architecture. The related problems and their (many) solutions are not within the scope of the specification of the basic common network layer.

Existing Addresses and Network Numbers

The Internet's version 4 numbering system has proven to be very flexible, (mostly) expandable, and simple. In short: it works. However, there are two problems. Neither was considered serious when the CATNIP was first developed in 1988 and 1989, but both are now of major concern:

- o The division into network, and then subnet, is insufficient. Almost all sites need a network assignment large enough to subnet. At the top of the hierarchy, there is a need to assign administrative domains.
- o As bit-packing is done to accomplish the desired network structure, the 32-bit limit causes more and more aggravation.

Another major addressing system used in open internetworking is the OSI method of specifying Network Service Access Points (NSAPs). The NSAP consists of an authority and format identifier, a number assigned to that authority, an address assigned by that authority, and a selector identifying the next layer (transport layer) protocol. This is actually a general multi-level hierarchy, often obscured by the details of specific profiles. (For example, CLNP doesn't specify 20 octet NSAPs, it allows any length. But various GOSIPs profile the NSAP as 20 octets, and IS-IS makes specific assumptions about the last 1-8 octets. And so on.)

The NSAP does not directly correspond to an IP address, as the selector in IP is separate from the address. The concept that does correspond is the NSAP less the selector, called the Network Entity Title or NET. (An unfortunate acronym, but one we will use to avoid repeating the full term.) The usual definition of NET is an NSAP with the selector set to 0; the NET used here omits the 0 selector.

There is also a network numbering system used by IPX, a product of Novell, Inc. (referred to from here on as Novell) and other vendors making compatible software. While IPX is not yet well connected into a global network, it has a larger installed base than either of the other network layers.

Network Layer Address

The network layer address looks like:

```
+-----+-----+-----+-----+
| length |  AFI  |  IDI ...  | DSP ...  |
+-----+-----+-----+-----+
```

The fields are named in the usual OSI terminology although that leads to an oversupply of acronyms. Here are more detailed descriptions of each field:

length: the number of bytes (octets) in the remainder of the address.

AFI: the Authority and Format Identifier. A single byte value, from a set of well-known values registered by ISO, that determines the semantics of the IDI field

IDI: the Initial Domain Identifier, a number assigned by the authority named by the AFI, formatted according to the semantics implied by the AFI, that determines the authority for the remainder of the address.

DSP: Domain Specific Part, an address assigned by the authority identified by the value of the IDI.

Note that there are several levels of authority. ISO, for example, identifies (with the AFI) a set of numbering authorities (like X.121, the numbering plan for the PSPDN, or E.164, the numbering plan for the telephone system). Each authority numbers a set of organizations or individuals or other entities. (For example, E.164 assigns [16172477959](#) to me as a telephone subscriber.)

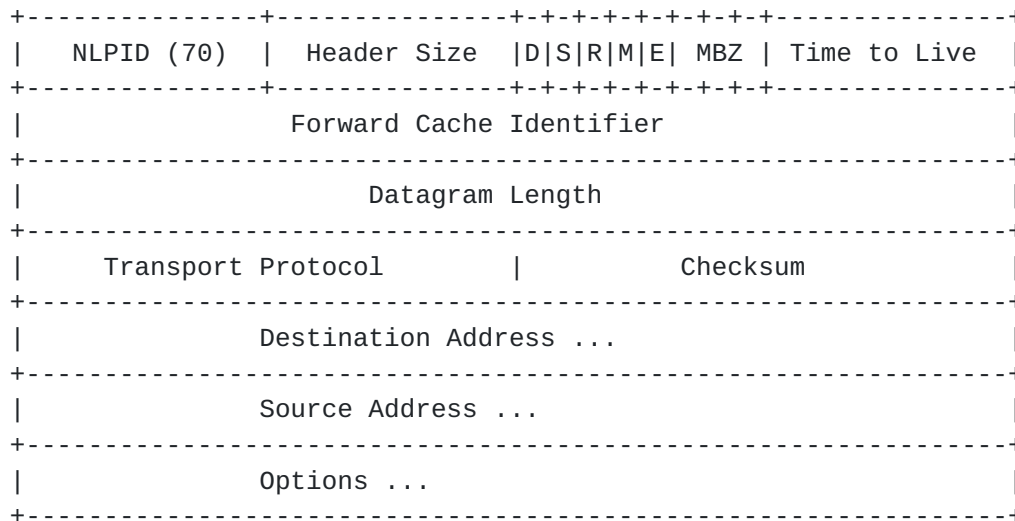
The entity then is the authority for the remainder of the address. I can do what I please with the addresses starting with (AFI=E.164) (IDI=16172477959). Note that this is a delegation of authority, and not an embedding of a data-link address (the telephone number) in a network layer address. The actual routing of the network layer address has nothing to do with the authority numbering.

The domain-specific part is variable length, and can be allocated in whatever way the authority identified by the AFI+IDI desires.

Network Layer Datagram

The common architecture format for network layer datagrams is described below. The design is a balance between use on high performance networks and routers, and a desire to minimize the number of bits in the fixed header. The CATNIP avoids the mistake of making the fixed field too small. Using the current state of processor technology as a reference, the fixed header is all loaded into CPU registers on the first memory cycle, and it all fits within the operation bandwidth. The header leaves the remaining data aligned on the header size (128 bits); with 64 bit addresses present and no options it leaves the transport header 256 bit aligned.

On very slow and low performance networks, the fixed header is still fairly small, and could be further compressed by methods similar to those used with IP version 4 on links that consider every bit precious. In between, it fits nicely into ATM cells and radio packets, leaving sufficient space for the transport header and application data.



NLPID: The first byte (the network layer protocol identifier in OSI) is an 8 bit constant 70 (hex). This corresponds to Internet Version 7.

Header Length: The header length is a 8-bit count of the number of 32-bit words in the header. This allows the header to be up to 1020 bytes in length.

Flags: This byte is a small set of flags determining the datagram header format and the processing semantics. The last three bits are reserved, and must be set to zero. (Note that the corresponding bits in CLNP version 1 are 001, since this byte is the version field. This may be useful.)

Destination Address Omitted: When the destination address omitted (DA0) flag

is zero, the destination address is present as shown in the datagram format diagram. When a datagram is sent with an FCI that identifies the destination and the DAO flag is set, the address does not appear in the datagram.

Source Address Omitted: The source address omitted (SAO) flag is zero when the source address is present in the datagram. When datagram is sent with an FCI that identifies the source and the SAO flag is set, the source address is omitted from the datagram.

Report Fragmentation Done: When this bit (RFD) is set, an intermediate router that fragments the datagram (because it is larger than the next subnetwork MTU) should report the event with an ICMP Datagram Too Big message. (Unlike IP version 4, which uses DF for MTU discovery, the RFD flag allows the fragmented datagram to be delivered.)

Mandatory Router Option: The mandatory router option (MRO) flag indicates that routers forwarding the datagram must look at the network header options.

If not set, an intermediate router should not look at the header options.

(But it may anyway; this is a necessary consequence of transparent network layer translation, which may occur anywhere.)

The destination host, or an intermediate router doing translation, must look at the header options regardless of the setting of the MRO flag.

A router doing fragmentation will normally only use the RFD flag in options to determine whether options should be copied within the fragmentation code path. (It might also recognize and elide null options.) If the MRO flag is not set, the router may not act on an option even though it copies it properly during fragmentation.

If there are no options present, MRO should always be zero, so that routers can follow the no-option profile path in their implementation. (Remember that the presence of options cannot be divined from the header length, since the addresses are variable length.)

Error Report Suppression: The ERS flag is set to suppress the sending of error reports by any system (whether host or router) receiving or forwarding the datagram. The system may log the error, increment network management counters, and take any similar action, but ICMP error messages or CNLP error reports must not be sent.

The ERS flag is normally set on ICMP messages and other network layer error reports. It does not suppress the normal response to ICMP queries or similar network layer queries (CNLP echo request).

If both the RFD and ERS flags are set, the fragmentation report is

sent.

(This definition allows a larger range of possibilities than simply over-riding the RFD flag would; a sender not desiring this behavior can see to it that RFD is clear.)

Time To Live: The time to live is a 8-bit count, nominally in seconds. Each hop is required to decrement TTL by at least one. A hop that holds a datagram for an unusual amount of time (more than 2 seconds, a typical example being a wait for a subnetwork connection establishment) should subtract the entire waiting time in seconds (rounded upward) from the TTL.

Forward Cache Identifier: Each datagram carries a 32 bit field, called "forward cache identifier", that is updated (if the information is available) at each hop. This field's value is derived from ICMP messages sent back by the next hop router, a routing protocol (e.g. RAP), or some other method. The FCI is used to expedite routing decisions by preserving knowledge where possible between consecutive routers. It can also be used to make datagrams stay within reserved flows, circuits, and mobile host tunnels. If an FCI is not available, this field must be zero, the SA0 and DA0 flags must be clear, and both destination and source addresses must appear in the datagram.

Datagram Length: The 32-bit length of the entire datagram in octets. A datagram can therefore be up to 4294967295 bytes in overall length. Particular networks normally impose lower limits.

Transport Protocol: The transport layer protocol. For example, TCP is 6.

Checksum: The checksum is a 16-bit checksum of the entire header, using the familiar algorithm used in IP version 4.

Destination: The destination address, a count byte followed by the destination NSAP with the zero selector omitted. This field is present only if the DA0 flag is zero. If the count field is not 3 modulo 4 (the destination is not an integral multiple of 32-bit words) zero bytes are added to pad to the next multiple of 32 bits. These pad bytes are not required to be ignored: routers may rely on them being zero.

Source: The source address, in the same format as the destination. Present only if the SA0 flag is zero. The source is padded in the same way as destination to arrive at a 32-bit boundary.

Options: Options may follow. They are variable length, and always 32-bit aligned. If the MRO flag in the header is not set, routers will usually not look at or take action on any option, regardless of the setting of the class field.

Multicasting

The multicast-enable option permits multicast forwarding of the CATNIP datagram on subnetworks that directly support media layer multicasting. This is a vanishing species, even in 10 Mbps Ethernet, given the increasing prevalence of switching hubs. It also (perhaps more usefully) permits a router to forward the datagram on multiple paths when a multicast routing algorithm has established such paths. There is no option data.

Note that there is no special address space for multicasting in the CATNIP. Multicast destination addresses can be allocated anywhere by any administration or authority. This supports a number of differing models of addressing. It does require that the transport layer protocol know that the destination is multicast; this is desirable in any case. (For example, the transport will probably want to set the ERS flag.)

On an IEEE 802.x (ISO 8802.x) type media, the last 23 bits of the address (not including the 0 selector) are used in combination with the multicast group address assigned to the Internet to form the media address when forwarding a datagram with the multicast enable option from a router to an attached network provided that the datagram was not received on that network with either multicast or broadcast media addressing. A host may send a multicast datagram either to the media multicast address (the IP catenet model,) or media unicast to a router which is expected to repeat it to the multicast address within the entire level I area or to repeat copies to the appropriate end systems within the area on non-broadcast media (the more general CLNP model.)

Network Layer Translation

The translating host or router must reassemble datagrams that have been fragmented before translation. Where the translation is being done by the destination host (for example, the case of a native CATNIP host receiving IP version 4 datagrams), this is similar to the present fragmentation model.

When it is being done by an intermediate router (acting as an internetwork layer gateway) the router should use all of source, destination, and datagram ID for identification of fragments. Note that destination is used implicitly in the usual reassembly at the destination. If the fragments take different paths through the net, and arrive at different translation points, the datagram is lost.

The objective of translation is to be able to upgrade systems, both hosts and routers, in whatever order desired by their owners. Organizations must be able to upgrade any given system without reconfiguration or modification of any other, and existing hosts must be able to interoperate essentially forever. (Non-CATNIP routers will probably be effectively eliminated at some point, except where they

exist in their own remote or isolated corners.)

Each CATNIP system, whether host or router, must be able to recognize adjacent systems in the topology that are (only) IP version 4, CLNP, or IPX and call the appropriate translation routine just before sending the datagram.

OSI CNLP

The translation between CLNP and the CATNIP compressed form of the datagrams is the simplest case for CATNIP, since the addresses are the same and need not be extended. The resulting CATNIP datagrams may omit the source and destination addresses as explained previously, and may be mixed with uncompressed datagrams on the same subnetwork link. Alternatively, a subnetwork may operate entirely in the CATNIP, converting all transit traffic to CATNIP datagrams, even if FCIs that would make the compression effective are not available.

Similarly, all network datagram formats with CATNIP mappings may be compressed into the common form, providing a uniform transit network service, with common routing protocols (such as IS-IS).

Internet Protocol

All existing version 4 numbers are defined as belonging to the Internet by using a new AFI, to be assigned to IANA by the ISO. This document uses 192 at present for clarity in examples; it is to be replaced with the assigned AFI. The AFI specifies that the IDI is two bytes long, containing an administrative domain number.

The AD (Administrative Domain), identifies an administration which may be an international authority (such as the existing InterNIC), a national administration, or a large multi-organization (e.g., a government). The idea is that there should not be more than a few hundred of these at first, and eventually thousands or tens of thousands at most.

AD numbers are assigned by IANA. Initially, the only assignment is the number 0.0, assigned to the InterNIC, encompassing the entire existing version 4 Internet.

The mapping from/to version 4 IP addresses:

length	AFI	IDI ...	DSP ...
7	192	AD number	version 4 address

While the address (DSP) is initially always the 4 byte, version 4 address, it can be extended to arbitrary levels of subnetting within the existing Internet numbering plan. Hosts with DSPs longer than 4 bytes will not be able to interoperate with version 4 hosts.

Novell IPX

The Internetwork Packet Exchange protocol, developed by Novell based on the XNS protocol (Xerox Network System) has many of the same capabilities as the Internet and OSI protocols. At first look, it appears to confuse the network and transport layers, as IPX includes both the network layer service and the user datagram service of the transport layer, while SPX (sequenced packet exchange) includes the IPX network layer and provides service similar to TCP or TP4. This turns out to be mostly a matter of the naming and ordering of fields in the packets, rather than any architectural difference.

IPX uses a 32-bit LAN network number, implicitly concatenated with the 48-bit MAC layer address to form an Internet address. Initially, the network numbers were not assigned by any central authority, and thus were not useful for inter-organizational traffic without substantial prior arrangement. There is now an authority established by Novell to assign unique 32-bit numbers and blocks of numbers to organizations that desire inter-organization networking with the IPX protocol.

The Novell/IPX numbering plan uses an ICD, to be assigned, to designate an address as an IPX address. This means Novell uses the authority (AFI=47)(ICD=Novell) and delegates assignments of the following 32 bits.

An IPX address in the common form looks like:

```
+-----+-----+-----+-----+
| length |  AFI  |  IDI ...  | DSP ...  |
+-----+-----+-----+-----+
|   13   | 47 (hex) | Novell ICD | network+MAC address |
+-----+-----+-----+-----+
```

This will always be followed by two bytes of zero padding when it appears in a common network layer datagram. Note that the socket numbers included in the native form IPX address are part of the transport layer.

SIPP

It may seem a little odd to describe the interaction with SIPP (version **6 of IP**) which is now only another experimental candidate for the next generation of network layer protocols. However, if SIPP is deployed, whether or not as the protocol of choice for replacement of IP version 4, there will then be four network protocols to accommodate. It is

prudent to investigate how SIPP could then be integrated into the common addressing plan and datagram format.

SIPP defines 64 bit addresses, which are included in the NSAP addressing plan under the Internet AFI as AD number 0.1. It is not clear at this time what administration will hold the authority for the SIPP numbering plan.

length	AFI	IDI ...	DSP ...
11	192	AD (0.1)	SIPP 64 bit address

The SIPP addressing method (the definition of the 64 bits) will not be described here. We note only that in the cases in which SIPP is intended to interoperate directly with IP version 4, the last 32 bits of the address is the IP version 4 address. This permits a convenient set of translations without disturbing the transport protocols.

The SIPP proposal also includes an encapsulated-tunnel proposal called IPAE, to address some of the issues that are designed into CATNIP. The CATNIP direct translation does not use the SIPP-IPAE packet formats. IPAE also specifies a "mapping table" for prefixes. This table is kept up-to-date by periodic FTP transfers from a "central site." The CATNIP definitions leave the problem of prefix selection when converting into SIPP firmly within the scope of the SIPP-IPAE proposal, and possible methods are not described here.

In translating from SIPP (IPv6) to CATNIP (IPv7), the only unusual aspect is that SIPP defines some things that are normally considered options to be "payloads" overloaded onto the transport protocol numbering space. Fortunately, the only one that need be considered is fragmentation; a fragmented SIPP datagram must be reassembled prior to conversion. Other "payloads" such as routing are ignored (translated verbatim) and will normally simply fail to achieve the desired effect.

Translation to SIPP is simple, except for the difficult problem of inventing the "prefix" if an implementation wants to support translating Internet AD 0.0 numbers into the SIPP addressing domain.

Internet DNS

CATNIP addresses are represented in the DNS with the NSAP RR. The data in the resource record is the NSAP, including the zero selector at the end. The zone file syntax for the data is a string of hexadecimal digits, with a period "." inserted between any two octets where desired for readability. For example:

The inverse (PTR) zone is .NSAP, with the CATNIP address (reversed). That is, like .IN-ADDR.ARPA, but with .NSAP instead. The octets are

represented as hexadecimal numbers, with leading 0's. (Zero is always written as ".00.")

This respects the difference in actual authority: the IANA is the authority for the entire space rooted in .IN-ADDR.ARPA. in the version 4 Internet, while in the new Internet it holds the authority only for C0.NSAP. The domain 00.00.C0.NSAP is to be delegated by IANA to the InterNIC. (Understanding that in present practice the InterNIC is the operator of the authoritative root.)

Security

The CATNIP design permits the direct use of the present proposals for network layer security being developed in the IPSEC WG of the IETF. There are a number of detailed requirements; the most relevant being that network layer datagram translation must not affect (cannot affect) the transport layers, since the TPDU is mostly inaccessible to the router. For example, the translation into IPX will only work if the port numbers are shadowed into the plaintext security header.

References

- [Chapin93] A. Lyman Chapin, David M. Piscitello. Open Systems Networking. Addison-Wesley, Reading, Massachusetts, 1993.
- [Perlman92] Radia Perlman. Interconnections: Bridges and Routers. Addison-Wesley. Reading, Massachusetts, 1992.
- [RFC791] Jon Postel, editor. Internet Protocol. DARPA Internet Program Protocol Specification, ISI/USC, September, 1981.
- [RFC792] Jon Postel, editor. Internet Control Message Protocol. DARPA Internet Program Protocol Specification, ISI/USC, September, 1981.
- [RFC793] Jon Postel, editor. Transmission Control Protocol. DARPA Internet Program Protocol Specification, ISI/USC, September, 1981.
- [RFC801] Jon Postel, NCP/TCP transition plan. November, 1981.
- [RFC1191] J. Mogul, S. Deering. Path MTU Discovery. November, 1990.
- [RFC1234] D. Provan. Tunneling IPX Traffic through IP Networks. Novell, Inc., June, 1991.

- [RFC1247] J. Moy. OSPF Version 2. Proteon, Inc., July, 1991.
- [RFC1287] D. Clark, L. Chapin, V. Cerf, R. Braden, R. Hobby. Towards the Future Internet Architecture. December, 1991.
- [RFC1335] Z. Wang, J. Crowcroft, Two-tier address structure for the Internet: A solution to the problem of address space exhaustion. May, 1992.
- [RFC1338] V. Fuller, T. Li, J. Yu, K. Varadhan. Supernetting: an Address Assignment and Aggregation Strategy. June, 1992.
- [RFC1347] R. W. Callon. TCP and UDP with Bigger Addresses (TUBA), A simple proposal for Internet addressing and routing. June, 1992.
- [RFC1466] E. Gerich. Guidelines for Managemnet of IP Address Space. Merit, May, 1993.
- [RFC1475] Robert Ullmann. TP/IX: The Next Internet. Process Software Corporation. June, 1993.
- [RFC1476] Robert Ullmann. RAP: Internet Route Access Protocol. Process Software Corporation. June, 1993.
- [RFC1561] D. Piscitello. Use of ISO CLNP in TUBA Environments. Core Competence. December, 1993.

Author's Addresses

Michael McGovern
Sunspot Graphics

EMAIL: scrivner@world.std.com

Robert Ullmann
Lotus Development Corporation
1 Rogers Street
Cambridge, MA 02142

Phone: +1 617 693 1315
Email: rullmann@crd.lotus.com