

Network Working Group
Internet Draft

B. Carpenter
CERN
21 March 1994

Expires 26 September 1994

File name [draft-carpenter-aeiou-00.txt](#)

Address Extension by IP Option Usage (AEIOU)

Status of this Memo

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 lid-abstracts.txt listing contained in the internet-drafts Shadow Directories on nic.ddn.mil, nnsf.net, nic.nordu.net, ftp.nisc.sri.com, or munnari.oz.au to learn the current status of any Internet Draft.

Distribution of this memo is unlimited. Comments should be submitted to the big-internet@munari.oz.au mailing list.

Abstract

This document proposes an addressing extension mechanism for IP as a possible life extender while awaiting IPng.

AEIOU proposal

It is possible that no IPng candidate will be considered mature enough in time to be widely deployed before the IPv4 address space is fully allocated. This document suggests a way to cope with this risk. Although it does not propose a new IPng candidate, the proposal does amount to a major change to IP, requiring new versions of host and router software. However, the impact on existing code is much less than that of any IPng proposal. By the same token, the only problem that it tackles is the address space problem.

During the process of collecting requirements for IPng, a number of people have stated the need to fix the IPv4 address space problem by

flexible address extensions beyond 32 bits. However, if this means that IPv4 addresses should be replaced by bigger addresses, it breaks the current Internet routing and addressing architecture.

AEIOU proposes to tackle the problem differently. Firstly, three conservative measures should be taken:

1. Do not extend the global Internet address space or modify the global routing architecture.
2. Do deploy CIDR and BGP4. Be very restrictive on allocations to new sites (i.e. give them the minimum CIDR space that will satisfy them for one or two years).
3. Do not allocate any new network numbers to sites that already have at least one network number of any class (A, B, C). Try to recover unused numbers and CIDRise sparse networks. Encourage use of [RFC 1597](#).

Secondly, provide a simple address extension mechanism as an IP option:

4. Add two new IPv4 options "Source Address Extension" (SAE) and "Destination Address Extension" (DAE).

Type: [copied, class 0, option number TBD]
Length: 6
Data: 4 octet address extension (AE)

The AE options are carried transparently and survive fragmentation. Neither, either, or both may be present.

5. AEs are allocated by a site (not to a site) when it has run out of addresses in its share of the IPv4 32-bit address space. The AE space is structured like Class A, B and C space and all normal subnetting techniques apply. A host in AE space can be considered to have an address in the form of a 64-bit couplet (site address, AE).

Each site has a globally unique 32-bit site address, which can be any IPv4 address from the site's IANA-allocated IPv4 address space. This site address then "hides" all the site's AEs. For example, the site currently allocated the Class B network 128.141 could use 128.141.141.141 as its site address, and this would be the wide-area address used for routing all packets to the site's 32-bit AE space. Wide-area routing does not change. Routing to non-AE hosts does not change.

6. Hosts now fall into three classes:

OLD hosts; still running 32-bit IP only.

NEW hosts; AE options implemented, but no allocated AE (i.e. NEW hosts are still in the global 32-bit space.)
By convention, AE=0.0.0.0 for NEW hosts.

AE hosts; AE options implemented, and an AE allocated.
AE hosts are in the new 64-bit space.

The following table shows which combinations of hosts can talk to each other, using 32 or 64 bit addresses.

	OLD	NEW	AE	
OLD	32	32	no	
NEW	32	32	64	
AE	no	64	64	

(There might be some cases where OLD and AE hosts can talk, if they are on the same subnet.)

The outline of the deployment plan follows.

Stage 1: A site address is allocated for every Internet site.
On-site routers are upgraded to AE.
All servers are upgraded from OLD to NEW.
OLD clients can still access NEW servers.

Stage 2: Ensure that all newly installed clients are AE hosts.
New clients no longer consume 32-bit addresses.

Stage 3: Progressively convert OLD clients to AE. This means upgrading software versions, and renumbering clients.
At the same time, [RFC 1597](#) addresses could be transformed into AEs if required.

Stage 4: Sites will now have many unused 32-bit addresses recovered during Stage 3. A fraction of these can be recovered by further use of CIDR, and can be re-allocated as site addresses for new Internet sites.

Stage 5: Theoretically, all servers can now be converted from NEW to AE. At this stage almost all 32-bit addresses can be recovered and re-used as in Stage 4. We now have a two-tier Internet address scheme (32-bit site address space managed by IANA, and 32-bit AE space managed by individual sites).

Thirdly, some more details:

7. IP packets flow from source to the destination site exactly as

they do today. The SAE and DAE options are copied (even if fragmentation occurs). At the ingress router on the destination site, if the DAE option is present, it is used by local routing protocols to route the packet to the correct physical subnet. These routing protocols can be any of those in use today, except that they look at the DAE field instead of the normal IP destination address (DA).

This is not optimal for router performance. However, the obvious solution of having the ingress router swap the DAE and the DA does not work, since this would disturb the TCP and UDP pseudo-header checksum calculation.

8. On a subnet such as Ethernet, a rather straightforward extension of ARP (EARP, extended ARP) is needed to obtain the physical address corresponding to the (site address, DAE) couplet. For further study, but clearly soluble.

9. DNS RR's will include the AE when it exists. The DNS reply will directly imply whether a host is OLD (no AE), NEW (AE = 0.0.0.0) or AE.

10. I haven't worked out how the AE is handled at the socket API yet. This is the biggest problem if software changes are to be minimised. It's no worse than for SIPP, of course. It can also be argued that forcing applications to use an API with multiple address formats (for example, AF_INET and AF_INET6 in the case of BSD sockets) is good, because this will be needed for the real IPng when it comes.

11. FOOTBAR ([RFC 1545](#)) can handle it for FTP.

12. etc. Lots more to be written, by analogy with the SIPP and TUBA transition documents.

References

TBD

Disclaimer and acknowledgements.

This is a personal view and does not necessarily represent that of my employer.

AEIOU is not the same as EIP ([RFC 1385](#)) but it is in some sense a simple inversion of the EIP idea, which has been at the back of my mind for two years. It resembles the original (1992) IPAE draft too and has some similarities to TP/IX ([RFC 1475](#)). Other ideas came from the IPng transition discussions in general, and there is a clear

Draft

Address Extension by IP Option Usage (AEIOU) 21 March 1994

resemblance to the DECnet Phase V deployment process.

Various individuals have commented on the earlier versions of this document; a full list will be included in a future version.

Author's Address

Brian E. Carpenter	
Group Leader, Communications Systems	Phone: +41 22 767-4967
Computing and Networks Division	Fax: +41 22 767-7155
CERN	Telex: 419000 cer ch
European Laboratory for Particle Physics	E-mail: brian@dxcoms.cern.ch
1211 Geneva 23, Switzerland	

Expires 26 September 1994

File name [draft-carpenter-aeiou-00.txt](#)

