

An IP Address Extension Proposal

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard. Distribution of this memo is unlimited.

Abstract

This RFC suggests an extension to the IP protocol to solve the shortage of IP address problem, and requests discussion and suggestions for improvements.

1. Introduction and Background

The Internet community has a well-developed, mature set of protocols that have been quite successful in providing network and transport services to users. However, because of the spectacular success of the TCP/IP protocols and the number of networks that desire connection to the Internet, there is a shortage of network numbers that can be assigned.

The current network addressing scheme uses a 32-bit IP address that has a network part and a local address part. The division between the network part and the local address part has been defined in terms of 5 address classes: class A, B, C, D, E. Of these, only class A, B, C addresses are assigned to hosts. Class D is used for multicasting and class E is reserved.

Class A has the highest order bit set to 0, a 7 bit network number and a 24 bit host address.

Class B has the two higher order bits set to 10, a 14 bit network number and a 16 bit host address.

Class C has the three higher order bits set to 110, a 21 bit network number and a 8 bit host address.

Class D has the four higher order bits set to 1110.

Class E has four higher address bits set to 1111.

Increasing the size of the IP address field to more than 32 bits would solve the problem, but at the expense of making a new IP header definition that would be incompatible with the existing base of IP implementations. OSI based solutions such as using CLNP have been proposed but would take time to implement.

2. Proposal for IP extension

The IP header format should not be modified to minimize the changes necessary for supporting the address extensions that are proposed in this RFC. Instead an "escape" mechanism can be used to specify larger address. The IP header length field is 4 bits and this allows a maximum of fifteen 32-bit words where each word is 4 octets. The minimum size of the IP header without options is 5 words, which leaves 10 words for options. One can reserve 6 words (24 octets) for the normal options and leave the remaining (4 words or 16 octets) for a new option type that specifies an extended address. The details of this mechanism are discussed below.

Class E should be defined with the its five high order bits set to 11110. Its current definition is that four 1's in the most significant bits represent a class E address.

A new class F is proposed with its six high order bits set to 111110. The new class F address would be placed in the same locations that are used for source and destination IP address in the IP header, but would specify that part of the addressing information is in the options part of the IP header. This is illustrated in the figure below:

```

+++++
| Ver. | IHL | TOS | Total Length |
+++++
| Identification | Flags | Fragment Offset |
+++++
| 1|1|1|1|1|0| Offset | Reserved | Source IP address part 1 |
+++++
| 1|1|1|1|1|0| Offset | Reserved | Destination IP address part 1 |
+++++
: Options :
+++++
: SADDR Code | Len adr. part 2 | Source IP address part 2 :
+++++
: DADDR Code | Len adr. part 2 | Destination IP address part 2 :
+++++
: Data :
+++++

```

The "Offset" field specifies the offset in words from the beginning of the IP header where the second part of the IP address is located. Its purpose is to avoid searching the options part for addressing information. The address in the options part is in the Type-Length-Value form for consistency with other IP options that are found in this part. The "Len adr. part" indicates the length of the second IP address part in octets. The lengths should be defined so that the second part of the IP address ends on a word boundary. For example, the possible length values are 4, 8 octets. It is proposed that new IP option codes be used for the SADDR and DADDR codes respectively.

The IP address is the 2 bytes in the fixed IP header part plus the address field defined in the options part.

If the "Len adr. part" field has a value of 4, the new class is designated as the F-4 class (Class F with IP address length of 4 octets).

If the "Len adr. part" field has a value of 8, the new class is designated as the F-8 class (Class F with IP address length of 8 octets).

Each of the F-4 and F-8 IP address class can be further subdivided into a network number and a host number field in a manner that is similar to the current IP addressing scheme.

The sub-class definitions for F-4 class are shown below. Though the 4 octets are drawn contiguously, the first 2 octets and the last 2 octets are not contiguous in the IP header.

Class F-4A has the highest order bit set to 0, a 7 bit network number and a 24 bit host address.

```

+++++
|0| net number | local part |
+++++

```

Class F-4B has the two higher order bits set to 10, a 14 bit network number and a 16 bit host address.

```

+++++
|1|0| net number | local part |
+++++

```

Class F-4C has the three higher order bit set to 110, a 21 bit network number and a 8 bit host address.

```

+-----+
|1|1|0|          net number          | local part |
+-----+

```

Class F-4D has the four higher order bits set to 1110. Class F-4D is reserved for multicasting.

```

+-----+
|1|1|1|0|          multicast          |
+-----+

```

The sub-class definitions for F-8 class are shown below. Though the 8 octets are drawn contiguously, the first 2 octets and the last 6 octets are not contiguous in the IP header.

Class F-8A has the highest order bit set to 0, a 7 bit network number and a 56 bit host address.

```

+-----+
|0| net number |          local part          |
+-----+
|          local part          |
+-----+

```

Class F-8B has the two higher order bits set to 10, a 14 bit network number and a 48 bit host address.

```

+-----+
|1|0|          net number          | local part |
+-----+
|          local part          |
+-----+

```

Class F-8C has the three higher order bit set to 110, a 21 bit network number and a 40 bit host address.

```

+-----+
|1|1|0|          net number          | local part |
+-----+
|          local part          |
+-----+

```

Class F-8D has the four higher order bits set to 1110, a 28 bit network number and a 32 bit host address.

```

+++++
|1|1|1|0|          net number          |
+++++
|          local part          |
+++++

```

Class F-8E has the five higher order bits set to 11110, a 35 bit network number and a 24 bit host address.

```

+++++
|1|1|1|1|0|        net number          |
+++++
| net number |        local part        |
+++++

```

Class F-8F has the six higher order bits set to 111110, a 44 bit network number and a 16 bit host address.

```

+++++
|1|1|1|1|1|0|      net number          |
+++++
| net number |      local part          |
+++++

```

Class F-8G has the seven higher order bits set to 1111110, a 49 bit network number and a 8 bit host address.

```

+++++
|1|1|1|1|1|1|0|    net number          |
+++++
| net number |    local part          |
+++++

```

3. Interoperability Issues

If the new class F address is seen by a host that does not support it the IP datagram will be ignored. So communication will not be possible with existing hosts, but the amount of modification for existing hosts is much less than implementing an entirely different IP header structure or a different protocol.

The receiving host must be modified to contain the following code sketched below:

```
if (Destination_IP_address & 0xFC000000 == 0xF8000000)
{
    /* New extended class F address */
    Class_F_Processing(Destination_IP_address);
}
```

The Class_F_Processing() procedure can be defined in a separate module. There will be other changes required to communicate the results of processing the class F address to the main IP processing module but they should not be so extensive.

Security Considerations

Security issues are not discussed in this memo.

Author's Address

Karanjit Siyan
Siyan Consulting Services
49 Taurus Road, Box 960
North Glastonbury
Emigrant, Montana 59027

Phone: 406-333-4491

EMail: 72550.1634@compuserve.com

