Network Working Group J. Mogul (Stanford)
Request for Comments: 950 J. Postel (ISI)
August 1985

Internet Standard Subnetting Procedure

Status Of This Memo

This RFC specifies a protocol for the ARPA-Internet community. If subnetting is implemented it is strongly recommended that these procedures be followed. Distribution of this memo is unlimited.

Overview

This memo discusses the utility of "subnets" of Internet networks, which are logically visible sub-sections of a single Internet network. For administrative or technical reasons, many organizations have chosen to divide one Internet network into several subnets, instead of acquiring a set of Internet network numbers. This memo specifies procedures for the use of subnets. These procedures are for hosts (e.g., workstations). The procedures used in and between subnet gateways are not fully described. Important motivation and background information for a subnetting standard is provided in RFC-940 [7].

Acknowledgment

This memo is based on RFC-917 [1]. Many people contributed to the development of the concepts described here. J. Noel Chiappa, Chris Kent, and Tim Mann, in particular, provided important suggestions. Additional contributions in shaping this memo were made by Zaw-Sing Su, Mike Karels, and the Gateway Algorithms and Data Structures Task Force (GADS).

1. Motivation

The original view of the Internet universe was a two-level hierarchy: the top level the Internet as a whole, and the level below it individual networks, each with its own network number. The Internet does not have a hierarchical topology, rather the interpretation of addresses is hierarchical. In this two-level model, each host sees its network as a single entity; that is, the network may be treated as a "black box" to which a set of hosts is connected.

While this view has proved simple and powerful, a number of organizations have found it inadequate, and have added a third level to the interpretation of Internet addresses. In this view, a given Internet network is divided into a collection of subnets.

The three-level model is useful in networks belonging to moderately large organizations (e.g., Universities or companies with more than one building), where it is often necessary to use more than one LAN cable to cover a "local area". Each LAN may then be treated as a subnet.

There are several reasons why an organization might use more than one cable to cover a campus:

- Different technologies: Especially in a research environment, there may be more than one kind of LAN in use; e.g., an organization may have some equipment that supports Ethernet, and some that supports a ring network.

- Limits of technologies: Most LAN technologies impose limits, based on electrical parameters, on the number of hosts connected, and on the total length of the cable. It is easy to exceed these limits, especially those on cable length.

- Network congestion: It is possible for a small subset of the hosts on a LAN to monopolize most of the bandwidth. A common solution to this problem is to divide the hosts into cliques of high mutual communication, and put these cliques on separate cables.

- Point-to-Point links: Sometimes a "local area", such as a university campus, is split into two locations too far apart to connect using the preferred LAN technology. In this case, high-speed point-to-point links might connect several LANs.

An organization that has been forced to use more than one LAN has three choices for assigning Internet addresses:

1. Acquire a distinct Internet network number for each cable; subnets are not used at all.

2. Use a single network number for the entire organization, but assign host numbers without regard to which LAN a host is on ("transparent subnets").

3. Use a single network number, and partition the host address space by assigning subnet numbers to the LANs ("explicit subnets").

Each of these approaches has disadvantages. The first, although not requiring any new or modified protocols, results in an explosion in the size of Internet routing tables. Information about the internal details of local connectivity is propagated everywhere, although it is of little or no use outside the local organization. Especially as some current gateway implementations do not have much space for routing tables, it would be good to avoid this problem.

The second approach requires some convention or protocol that makes the collection of LANs appear to be a single Internet network. For example, this can be done on LANs where each Internet address is translated to a hardware address using an Address Resolution Protocol (ARP), by having the bridges between the LANs intercept ARP requests for non-local targets, see RFC-925 [2]. However, it is not possible to do this for all LAN technologies, especially those where ARP protocols are not currently used, or if the LAN does not support broadcasts. A more fundamental problem is that bridges must discover which LAN a host is on, perhaps by using a broadcast algorithm. As the number of LANs grows, the cost of broadcasting grows as well; also, the size of translation caches required in the bridges grows with the total number of hosts in the network.

The third approach is to explicitly support subnets. This does have a disadvantage, in that it is a modification of the Internet Protocol, and thus requires changes to IP implementations already in use (if these implementations are to be used on a subnetted network). However, these changes are relatively minor, and once made, yield a simple and efficient solution to the problem. Also, the approach avoids any changes that would be incompatible with existing hosts on non-subnetted networks.

Further, when appropriate design choices are made, it is possible for hosts which believe they are on a non-subnetted network to be used on a subnetted one, as explained in RFC-917 [1]. This is useful when it is not possible to modify some of the hosts to support subnets explicitly, or when a gradual transition is preferred.

2. Standards for Subnet Addressing

This section first describes a proposal for interpretation of Internet addresses to support subnets. Next it discusses changes to host software to support subnets. Finally, it presents a procedures for discovering what address interpretation is in use on a given network (i.e., what address mask is in use).

2.1. Interpretation of Internet Addresses

Suppose that an organization has been assigned an Internet network number, has further divided that network into a set of subnets, and wants to assign host addresses: how should this be done? Since there are minimal restrictions on the assignment of the "local address" part of the Internet address, several approaches have been proposed for representing the subnet number:

1. Variable-width field: Any number of the bits of the local address part are used for the subnet number; the size of this field, although constant for a given network, varies from network to network. If the field width is zero, then subnets are not in use.

2. Fixed-width field: A specific number of bits (e.g., eight) is used for the subnet number, if subnets are in use.

3. Self-encoding variable-width field: Just as the width (i.e., class) of the network number field is encoded by its high-order bits, the width of the subnet field is similarly encoded.

4. Self-encoding fixed-width field: A specific number of bits is used for the subnet number.

5. Masked bits: Use a bit mask ("address mask") to identify which bits of the local address field indicate the subnet number.

What criteria can be used to choose one of these five schemes? First, should we use a self-encoding scheme? And, should it be possible to tell from examining an Internet address if it refers to a subnetted network, without reference to any other information?

An interesting feature of self-encoding is that it allows the address space of a network to be divided into subnets of different sizes, typically one subnet of half the address space and a set of small subnets.

For example, consider a class C network that uses a self-encoding scheme with one bit to indicate if it is the large subnet or not and an additional three bits to identify the small subnet. If the first bit is zero then this is the large subnet, if the first bit is one then the following bits (3 in this example) give the subnet number. There is one subnet with 128 host addresses, and eight subnets with 16 hosts each.

To establish a subnetting standard the parameters and interpretation of the self-encoding scheme must be fixed and consistent throughout the Internet.

It could be assumed that all networks are subnetted. This would allow addresses to be interpreted without reference to any other information.

This is a significant advantage, that given the Internet address no additional information is needed for an implementation to determine if two addresses are on the same subnet. However, this can also be viewed as a disadvantage: it may cause problems for networks which have existing host numbers that use arbitrary bits in the local address part. In other words, it is useful to be able to control whether a network is subnetted independently from the assignment of host addresses.

The alternative is to have the fact that a network is subnetted kept separate from the address. If one finds, somehow, that the network is subnetted then the standard self-encoded subnetted network address rules are followed, otherwise the non-subnetted network addressing rules are followed.

If a self-encoding scheme is not used, there is no reason to use a fixed-width field scheme: since there must in any case be some per-network "flag" to indicate if subnets are in use, the additional cost of using an integer (a subnet field width or address mask) instead of a boolean is negligible. The advantage of using the address mask scheme is that it allows each organization to choose the best way to allocate relatively scarce bits of local address to subnet and host numbers. Therefore, we choose the address-mask scheme: it is the most flexible scheme, yet costs no more to implement than any other.

For example, the Internet address might be interpreted as:

<network-number><subnet-number><host-number>

where the <network-number> field is as defined by IP [3], the <host-number> field is at least 1-bit wide, and the width of the <subnet-number> field is constant for a given network. No further structure is required for the <subnet-number> or <host-number> fields. If the width of the <subnet-number> field is zero, then the network is not subnetted (i.e., the interpretation of [3] is used).

For example, on a Class B network with a 6-bit wide subnet field, an address would be broken down like this:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 0|           NETWORK           |  SUBNET   |    Host Number  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Since the bits that identify the subnet are specified by a bitmask, they need not be adjacent in the address. However, we recommend that the subnet bits be contiguous and located as the most significant bits of the local address.

Special Addresses:

From the Assigned Numbers memo [9]:

"In certain contexts, it is useful to have fixed addresses with functional significance rather than as identifiers of specific hosts. When such usage is called for, the address zero is to be interpreted as meaning "this", as in "this network". The address of all ones are to be interpreted as meaning "all", as in "all hosts". For example, the address 128.9.255.255 could be interpreted as meaning all hosts on the network 128.9. Or, the address 0.0.0.37 could be interpreted as meaning host 37 on this network."

It is useful to preserve and extend the interpretation of these special addresses in subnetted networks. This means the values of all zeros and all ones in the subnet field should not be assigned to actual (physical) subnets.

In the example above, the 6-bit wide subnet field may have any value except 0 and 63.

Please note that there is no effect or new restriction on the addresses of hosts on non-subnetted networks.

2.2. Changes to Host Software to Support Subnets

In most implementations of IP, there is code in the module that handles outgoing datagrams to decide if a datagram can be sent directly to the destination on the local network or if it must be sent to a gateway.

Generally the code is something like this:

```
IF ip_net_number(dg.ip_dest) = ip_net_number(my_ip_addr)
    THEN
        send_dg_locally(dg, dg.ip_dest)
    ELSE
        send_dg_locally(dg,
                         gateway_to(ip_net_number(dg.ip_dest)))
```

(If the code supports multiply-connected networks, it will be more complicated, but this is irrelevant to the current discussion.)

To support subnets, it is necessary to store one more 32-bit quantity, called my_ip_mask. This is a bit-mask with bits set in the fields corresponding to the IP network number, and additional bits set corresponding to the subnet number field.

The code then becomes:

```
IF bitwise_and(dg.ip_dest, my_ip_mask)
                          = bitwise_and(my_ip_addr, my_ip_mask)
    THEN
        send_dg_locally(dg, dg.ip_dest)
    ELSE
        send_dg_locally(dg,
               gateway_to(bitwise_and(dg.ip_dest, my_ip_mask)))
```

Of course, part of the expression in the conditional can be pre-computed.

It may or may not be necessary to modify the "gateway_to" function, so that it too takes the subnet field bits into account when performing comparisons.

To support multiply-connected hosts, the code can be changed to keep the "my_ip_addr" and "my_ip_mask" quantities on a per-interface basis; the expression in the conditional must then be evaluated for each interface.

2.3. Finding the Address Mask

How can a host determine what address mask is in use on a subnet to which it is connected? The problem is analogous to several other "bootstrapping" problems for Internet hosts: how a host determines its own address, and how it locates a gateway on its local network. In all three cases, there are two basic solutions: "hardwired" information, and broadcast-based protocols.

Hardwired information is that available to a host in isolation from a network. It may be compiled-in, or (preferably) stored in a disk file. However, for the increasingly common case of a diskless workstation that is bootloaded over a LAN, neither hardwired solution is satisfactory.

Instead, since most LAN technology supports broadcasting, a better method is for the newly-booted host to broadcast a request for the necessary information. For example, for the purpose of determining its Internet address, a host may use the "Reverse Address Resolution Protocol" (RARP) [4].

However, since a newly-booted host usually needs to gather several facts (e.g., its IP address, the hardware address of a gateway, the IP address of a domain name server, the subnet address mask), it would be better to acquire all this information in one request if possible, rather than doing numerous broadcasts on the network. The mechanisms designed to boot diskless workstations can also load per-host specific configuration files that contain the required information (e.g., see RFC-951 [8]). It is possible, and desirable, to obtain all the facts necessary to operate a host from a boot server using only one broadcast message.

In the case where it is necessary for a host to find the address mask as a separate operation the following mechanism is provided:

> To provide the address mask information the ICMP protocol [5] is extended by adding a new pair of ICMP message types, "Address Mask Request" and "Address Mask Reply", analogous to the "Information Request" and "Information Reply" ICMP messages. These are described in detail in Appendix I.
>
> The intended use of these new ICMP messages is that a host, when booting, broadcast an "Address Mask Request" message. A

gateway (or a host acting in lieu of a gateway) that receives this message responds with an "Address Mask Reply". If there is no indication in the request which host sent it (i.e., the IP Source Address is zero), the reply is broadcast as well. The requesting host will hear the response, and from it determine the address mask.

Since there is only one possible value that can be sent in an "Address Mask Reply" on any given LAN, there is no need for the requesting host to match the responses it hears against the request it sent; similarly, there is no problem if more than one gateway responds. We assume that hosts reboot infrequently, so the broadcast load on a network from use of this protocol should be small.

If a host is connected to more than one LAN, it might have to find the address mask for each.

One potential problem is what a host should do if it can not find out the address mask, even after a reasonable number of tries. Three interpretations can be placed on the situation:

1. The local net exists in (permanent) isolation from all other nets.

2. Subnets are not in use, and no host can supply the address mask.

3. All gateways on the local net are (temporarily) down.

The first and second situations imply that the address mask is identical with the Internet network number mask. In the third situation, there is no way to determine what the proper value is; the safest choice is thus a mask identical with the Internet network number mask. Although this might later turn out to be wrong, it will not prevent transmissions that would otherwise succeed. It is possible for a host to recover from a wrong choice: when a gateway comes up, it should broadcast an "Address Mask Reply"; when a host receives such a message that disagrees with its guess, it should change its mask to conform to the received value. No host or gateway should send an "Address Mask Reply" based on a "guessed" value.

Finally, note that no host is required to use this ICMP protocol to discover the address mask; it is perfectly reasonable for a host with non-volatile storage to use stored information (including a configuration file from a boot server).

Appendix I. Address Mask ICMP

Address Mask Request or Address Mask Reply

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |     Code      |          Checksum             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Identifier          |       Sequence Number         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Address Mask                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IP Fields:

Addresses

The address of the source in an address mask request message will be the destination of the address mask reply message. To form an address mask reply message, the source address of the request becomes the destination address of the reply, the source address of the reply is set to the replier's address, the type code changed to AM2, the address mask value inserted into the Address Mask field, and the checksum recomputed. However, if the source address in the request message is zero, then the destination address for the reply message should denote a broadcast.

ICMP Fields:

Type

AM1 for address mask request message

AM2 for address mask reply message

Code

0 for address mask request message

0 for address mask reply message

Checksum

The checksum is the 16-bit one's complement of the one's complement sum of the ICMP message starting with the ICMP Type. For computing the checksum, the checksum field should be zero. This checksum may be replaced in the future.

Identifier

An identifier to aid in matching requests and replies, may be zero.

Sequence Number

A sequence number to aid in matching requests and replies, may be zero.

Address Mask

A 32-bit mask.

Description

A gateway receiving an address mask request should return it with the address mask field set to the 32-bit mask of the bits identifying the subnet and network, for the subnet on which the request was received.

If the requesting host does not know its own IP address, it may leave the source field zero; the reply should then be broadcast. However, this approach should be avoided if at all possible, since it increases the superfluous broadcast load on the network. Even when the replies are broadcast, since there is only one possible address mask for a subnet, there is no need to match requests with replies. The "Identifier" and "Sequence Number" fields can be ignored.

Type AM1 may be received from a gateway or a host.

Type AM2 may be received from a gateway, or a host acting in lieu of a gateway.

Appendix II. Examples

These examples show how a host can find out the address mask using the ICMP Address Mask Request and Address Mask Reply messages. For the following examples, assume that address 255.255.255.255 denotes "broadcast to this physical medium" [6].

1. A Class A Network Case

For this case, assume that the requesting host is on class A network 36.0.0.0, has address 36.40.0.123, that there is a gateway at 36.40.0.62, and that a 8-bit wide subnet field is in use, that is, the address mask is 255.255.0.0.

The most efficient method, and the one we recommend, is for a host to first discover its own address (perhaps using "RARP" [4]), and then to send the ICMP request to 255.255.255.255:

```
Source address:          36.40.0.123
Destination address:     255.255.255.255
Protocol:                ICMP = 1
Type:                    Address Mask Request = AM1
Code:                    0
Mask:                    0
```

The gateway can then respond directly to the requesting host.

```
Source address:          36.40.0.62
Destination address:     36.40.0.123
Protocol:                ICMP = 1
Type:                    Address Mask Reply = AM2
Code:                    0
Mask:                    255.255.0.0
```

Suppose that 36.40.0.123 is a diskless workstation, and does not know even its own host number. It could send the following datagram:

```
Source address:          0.0.0.0
Destination address:     255.255.255.255
Protocol:                ICMP = 1
Type:                    Address Mask Request = AM1
Code:                    0
Mask:                    0
```

36.40.0.62 will hear the datagram, and should respond with this datagram:

```
Source address:          36.40.0.62
Destination address:     255.255.255.255
Protocol:                ICMP = 1
Type:                    Address Mask Reply = AM2
Code:                    0
Mask:                    255.255.0.0
```

Note that the gateway uses the narrowest possible broadcast to reply. Even so, the over use of broadcasts presents an unnecessary load to all hosts on the subnet, and so the use of the "anonymous" (0.0.0.0) source address must be kept to a minimum.

If broadcasting is not allowed, we assume that hosts have wired-in information about neighbor gateways; thus, 36.40.0.123 might send this datagram:

```
Source address:          36.40.0.123
Destination address:     36.40.0.62
Protocol:                ICMP = 1
Type:                    Address Mask Request = AM1
Code:                    0
Mask:                    0
```

36.40.0.62 should respond exactly as in the previous case.

```
Source address:          36.40.0.62
Destination address:     36.40.0.123
Protocol:                ICMP = 1
Type:                    Address Mask Reply = AM2
Code:                    0
Mask:                    255.255.0.0
```

2. A Class B Network Case

For this case, assume that the requesting host is on class B network 128.99.0.0, has address 128.99.4.123, that there is a gateway at 128.99.4.62, and that a 6-bit wide subnet field is in use, that is, the address mask is 255.255.252.0.

The host sends the ICMP request to 255.255.255.255:

```
Source address:          128.99.4.123
Destination address:     255.255.255.255
Protocol:                ICMP = 1
Type:                    Address Mask Request = AM1
Code:                    0
Mask:                    0
```

The gateway can then respond directly to the requesting host.

```
Source address:           128.99.4.62
Destination address:      128.99.4.123
Protocol:                 ICMP = 1
Type:                     Address Mask Reply = AM2
Code:                     0
Mask:                     255.255.252.0
```

In the diskless workstation case the host sends:

```
Source address:           0.0.0.0
Destination address:      255.255.255.255
Protocol:                 ICMP = 1
Type:                     Address Mask Request = AM1
Code:                     0
Mask:                     0
```

128.99.4.62 will hear the datagram, and should respond with this datagram:

```
Source address:           128.99.4.62
Destination address:      255.255.255.255
Protocol:                 ICMP = 1
Type:                     Address Mask Reply = AM2
Code:                     0
Mask:                     255.255.252.0
```

If broadcasting is not allowed 128.99.4.123 sends:

```
Source address:           128.99.4.123
Destination address:      128.99.4.62
Protocol:                 ICMP = 1
Type:                     Address Mask Request = AM1
Code:                     0
Mask:                     0
```

128.99.4.62 should respond exactly as in the previous case.

```
Source address:           128.99.4.62
Destination address:      128.99.4.123
Protocol:                 ICMP = 1
Type:                     Address Mask Reply = AM2
Code:                     0
Mask:                     255.255.252.0
```

3. A Class C Network Case (illustrating non-contiguous subnet bits)

For this case, assume that the requesting host is on class C network 192.1.127.0, has address 192.1.127.19, that there is a gateway at 192.1.127.50, and that on network an 3-bit subnet field is in use (01011000), that is, the address mask is 255.255.255.88.

The host sends the ICMP request to 255.255.255.255:

```
Source address:          192.1.127.19
Destination address:     255.255.255.255
Protocol:                ICMP = 1
Type:                    Address Mask Request = AM1
Code:                    0
Mask:                    0
```

The gateway can then respond directly to the requesting host.

```
Source address:          192.1.127.50
Destination address:     192.1.127.19
Protocol:                ICMP = 1
Type:                    Address Mask Reply = AM2
Code:                    0
Mask:                    255.255.255.88.
```

In the diskless workstation case the host sends:

```
Source address:          0.0.0.0
Destination address:     255.255.255.255
Protocol:                ICMP = 1
Type:                    Address Mask Request = AM1
Code:                    0
Mask:                    0
```

192.1.127.50 will hear the datagram, and should respond with this datagram:

```
Source address:          192.1.127.50
Destination address:     255.255.255.255
Protocol:                ICMP = 1
Type:                    Address Mask Reply = AM2
Code:                    0
Mask:                    255.255.255.88.
```

If broadcasting is not allowed 192.1.127.19 sends:

```
Source address:          192.1.127.19
Destination address:     192.1.127.50
Protocol:                ICMP = 1
Type:                    Address Mask Request = AM1
Code:                    0
Mask:                    0
```

192.1.127.50 should respond exactly as in the previous case.

```
Source address:          192.1.127.50
Destination address:     192.1.127.19
Protocol:                ICMP = 1
Type:                    Address Mask Reply = AM2
Code:                    0
Mask:                    255.255.255.88
```

Appendix III. Glossary

Bridge

A node connected to two or more administratively indistinguishable but physically distinct subnets, that automatically forwards datagrams when necessary, but whose existence is not known to other hosts. Also called a "software repeater".

Gateway

A node connected to two or more administratively distinct networks and/or subnets, to which hosts send datagrams to be forwarded.

Host Field

The bit field in an Internet address used for denoting a specific host.

Internet

The collection of connected networks using the IP protocol.

Local Address

The rest field of the Internet address (as defined in [3]).

Network

A single Internet network (which may or may not be divided into subnets).

Network Number

The network field of the Internet address.

Subnet

One or more physical networks forming a subset of an Internet network. A subnet is explicitly identified in the Internet address.

Subnet Field

The bit field in an Internet address denoting the subnet number. The bits making up this field are not necessarily contiguous in the address.

Subnet Number

A number identifying a subnet within a network.

Appendix IV. Assigned Numbers

The following assignments are made for protocol parameters used in the support of subnets. The only assignments needed are for the Internet Control Message Protocol (ICMP) [5].

ICMP Message Types

AM1 = 17

AM2 = 18

References

[1] Mogul, J., "Internet Subnets", RFC-917, Stanford University, October 1984.

[2] Postel, J., "Multi-LAN Address Resolution", RFC-925, USC/Information Sciences Institute, October 1984.

[3] Postel, J., "Internet Protocol", RFC-791, USC/Information Sciences Institute, September 1981.

[4] Finlayson, R., T. Mann, J. Mogul, M. Theimer, "A Reverse Address Resolution Protocol", RFC-903, Stanford University, June 1984.

[5] Postel, J., "Internet Control Message Protocol", RFC-792, USC/Information Sciences Institute, September 1981.

[6] Mogul, J., "Broadcasting Internet Datagrams", RFC-919, Stanford University, October 1984.

[7] GADS, "Towards an Internet Standard Scheme for Subnetting", RFC-940, Network Information Center, SRI International, April 1985.

[8] Croft, B., and J. Gilmore, "BOOTP -- UDP Bootstrap Protocol", RFC-951, Stanford University, August 1985.

[9] Reynolds, J., and J. Postel, "Assigned Numbers", RFC-943, USC/Information Sciences Institute, April 1985.