

组织：中国互动出版网 (<http://www.china-pub.com/>)

RFC 文档中文翻译计划 (<http://www.china-pub.com/compters/emook/aboutemook.htm>)

E-mail: ouyang@china-pub.com

译者：傅小均 (michael_fu_fuxj@21cn.com)

译文发布时间：2001-5-9

版权：本中文翻译文档版权归中国互动出版网所有。可以用于非商业用途自由转载，但必须保留本文档的翻译及版权信息。

Network Working Group
Request for Comments: 917
Category: Standards Track

Jeffrey Mogul
Computer Science Department
Stanford University
October 1984

因特网子网

(RFC917 ——Internet Subnets)

本备忘录的状态

本文档是有关 **Internet** 的协议的提案，有待讨论。本备忘录的发布不受任何限制。

摘要

本文档讨论因特网中“子网”的效用。“子网”是整个因特网中的一部分。由于管理和技术的原因，许多机构选择把一个网络分成几个子网，而不是单纯的使用一系列的因特网地址。

本文档提出使用子网的程序和过程，并讨论解决由此而产生的问题的方法，特别是路由问题。

目录

1. 介绍	2
1. 1. 术语.....	3
2. 子网地址分配标准	3
2. 1 <i>Internet</i> 地址的解析.....	3
2. 2 为支持子网，软件所需的改动.....	4
2. 3 子网和广播.....	5
2. 4 决定子网字段的长度.....	6
3. 子网路由方法.....	6
4. 例子.....	7

4. 1 斯坦福大学.....	7
4. 2 麻省理工学院 (MIT)	8
4. 3 卡内基-梅隆大学 (CMU)	9
5. 地址格式因特网信报控制协议 (ICMP)	9
5. 1 描述.....	9
5. 2 例子.....	10
参考	12

1. 介绍

Internet 在开始时被视为两层结构，高层是作为一个整体的链式网，其下是一系列“网络”的集合，每一个网络都有各自的网络号。（虽然 **Internet** 的拓扑结构其实是不分层的，但 **Internet** 的地址解析是分层的。）

这种做法曾一度被证明是简单而有效的，但许多机构发现并不过充分。因此，在对 **Internet** 地址的解析中加入了第三层。从这个观点出发，某一特定的网络就需要（也可能不需要）分层一系列的子网。

将 **Internet** 视为两层的观点是建立在这样一个假设之上的，即：对一台处于某网络中的主机而言，它所处的网络只有一个边界，也就是说，这个网络可以被视为一个有许多主机相连接的黑盒。这对 **Internet** 早期的 **ARPA** 网来说是对的。因为 **IMPs** 屏蔽了网络中的特殊连接的使用。对大多数局域网技术来说也是这样，比如以太网和环网。

但这种假设在许多实践中却是不对的。在一个中等大小的机构中，比如有好几个建筑物的大学和公司，常常需要多条局域网网线将“局部地区”相连。在写这篇文档是，斯坦福大学就有 18 条这样的网线，而且更多的还在计划中。

要用多条网线连接几个区域的原因有几个：

- 不同的技术的网络：特别是在研究环境中，可能会有几个不同的局域网，例如，某个机构有一些设备支持以太网，而另一些则支持环网。
- 技术的限制：多数技术由于起电气参数的限制，而对连接的主机数和网线的总长度有限制。这些限制，特别是网线长度很容易达到。
- 网络拥塞：在一个局域网中，一小部分的主机很可能独占大部分的带宽。通常解决这个问题的方法是把主机根据相互间通信的多少分成几部分，各部分使用不同的网线。
- 点对点的连接：有时一个“局部区域”被分成几个部分，而个部分之间的距离对上述局域网技术来说太远了。在这种情况下，高速的点对点连接可以用来连接这些局域网。

对不得不使用多个局域网的机构来说，分配 **Internet** 地址有三种选择：

1. 为每一条网线分配一个网络号。
2. 为整个机构分配一个网络号，并给主机分配地址，而不理会主机在哪个局域网中。
3. 使用一个网络地址，并分成几个地址空间，从中给每个局域网分配一个子网地址（显式子网）。

每一种方法都有缺点。第一种方法虽然不需要修改和增加现有协议，但会导致路由

表的急剧增大，整个网络的内部连通性信息传播于整个 Internet，而这些信息对这个机构以外的世界没有用处。特别是现在有些网关没有很大的路由表空间。所以这样的问题应该避免。

第二种方法需要一定的协议把某些局域网的整合成一个单一的网络。例如，在使用地址解析协议（ARP）的局域网中，Internet 地址被解析成为硬件地址，局域网间的网桥会拦截 ARP 对非本地目标的请求。但不是所有的局域网技术都可以做到这一点，特别是没有使用 ARP 或不支持广播协议的。一个更基本的问题是，网桥要知道每台主机在哪个局域网中（这些信息可以用广播算法获得），随着主机的增多，广播的代价也随之增大，转换所需的缓冲也随之增大。

第三种方法的关键问题是：校友的标准认为所有同一局域网上的主机都是用同一网线相连的。解决方法是显式的支持子网。这就需要改变现有的 Internet 协议，改变现在正在使用的 IP 的实现方法。但我们认为，这样的改动不是很大，而且只需修改一次，就能得到一个简单有效的解决方法。我们在本文档中使用的方法会避免导致和现有的非子网上的主机不兼容的修改。

当找到合适的方法，就有可能是子网内的主机并不知道自己处于子网中。这点在后面会解释。当不能修改主机以使其支持“显式子网”时，这样做是非常有用的。

1. 1. 术语

为了讲述的清楚和简洁，这里定义一些术语，并在以后的文中使用：

链式网：连接在一起的网络的集合

网络：Internet 中的一个网络（可以分成子网，也可以不分）

子网：网络中的一部分

网络号：见参考[8]

本地地址：Internet 地址中没有分配给网络号使用的位，也叫“剩余位”

子网号：网络中标识子网的号码

子网位：Internet 地址中分配给子网号使用的位

主机位：Internet 地址中用于指明特定主机使用的位

网关：连接两个或更多不同网络或子网，传递数据的节点

网桥：连接两个或更多物理上可分，但管理上不可分的子网，在必要使传递数据包

的节点，主机不知道其存在。

2 子网地址分配标准

根据参考[2]中的描述，划分子网也就是地址的分配问题。在这部分中，我们首先提出一个支持子网的地址解析方案，然后讨论这种地址格式和广播之间的关系，最后给出一个地址解析协议。

2. 1 Internet 地址的解析

假设某机构分配到一个网络号，并将之分成一系列子网，再分配给主机。如何进行呢？因为对于 Internet 地址中本地地址部分的分配限制很少，因此对子网号的分配主要

有以下几种方法：

- a) 变长字段：本地地址部分任意位都可以给子网号使用，虽然这部分长度对某一特定网络是一定的，但各网络间可以不同。如果长度是 0，则说明没有使用子网。
- b) 定长字段：指定长度的字段（比如 8 位）用语子网号（在使用子网的情况下）。
- c) 自编码变长字段：网络好的字段长度是由其高位决定，相似的，子网号的字段长度也由其高位决定。
- d) 自编码定长字段：一定长度的字段给子网使用。如果最高位是 1，则使用子网，否则没有使用。

用什么标准从这四个方案中选择一个呢？首先，确定是否要选用自编码方案，也就是说能否通过检测一个因特网地址就能得知这个地址是否用道子网？

自编码的一个优点是，人们能知道一个非本地的网络是否被划分成子网。这是否有用还不是很清楚。但主要的好处是不需要额外的信息来说明两个地址是否在同一子网上。然而，从另一个角度看，这也会是个缺点：对于非子网网络，如果有主机在其地址的本地地址字段中任意使用，则会导致问题（1）。也就是说，如果能够独立于主机地址的分配而控制网络是否子网，这会非常有用。另一个自编码方案的缺点是，给主机使用的地址空间会减少至少 2 位。

如果没有使用自编码方案，很明显，变长子网字段方案是合适的。既然任何情况下每个网络都有“标志”显示是否使用子网，使用整数型标志比使用布尔型标志所多的耗费也就可以忽略。使用变长子网字段的好处是允许每个机构选择最好的分配方案，以应付给子网和主机使用的地址位数的相对不足。

因此，我们提议的因特网地址的解析是：

<网络号><子网号><主机号>

网络号使用的位在参考[8]中有述。主机号字段至少长 1 位。子网字段的长度在一个网络中是固定的。子网字段和主机字段不需要其他的数据。如果子网字段的长度是 0，则说明没有使用子网。

例如，在一个 A 类网络中的有 8 位长的子网字段，则它的地址如下：

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|0|   网络       |   子网       |           主机号           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

为了实现的简单和有效，我们希望所有的机构都使用 8 位或者 8 的倍数的子网字段长度。但作为一个统一的实现方法，必须能够其他可能的长度。

我们反对“递归子网”的使用，就是将主机号字段再分成子网和主机两部分。因为：

- 没有对四层结构的明显的需求。
- IP 地址中没有足够的位使这种方法有实用价值。
- 需要复杂的而外机制

2. 2 为支持子网，软件所需的改动

在大多数 IP 的实现中，处理向外发送数据包的模块里常有类似下面的代码：

```

IF ip_net_number(packet.ip_dest) = ip_net_number(my_ip_addr)
  THEN
    send_packet_locally(packet, packet.ip_dest)

```

```

ELSE
    send_packet_locally(packet,
        gateway_to(ip_net_number(packet.ip_dest)))
IF 因特网网络号（数据包的目标地址） = 自己的网络号
THEN
    发送本地数据包
ELSE
    发送本地数据包到网关

```

为了支持子网，需要另一个 32 位的值，成为网络掩码。这是一个位掩码，各个位的设置和 IP 网络号以及子网号相对应。例如，一个 A 类网络使用 8 位子网字段，则其掩码为 255.255.0.0。

则上述的程序代码变为：

```

IF bitwise_and(packet.ip_dest, my_ip_mask)
    = bitwise_and(my_ip_addr, my_ip_mask)
THEN
    send_packet_locally(packet, packet.ip_dest)
ELSE
    send_packet_locally(packet,
        gateway_to(bitwise_and(packet.ip_dest, my_ip_mask)))

```

当然，部分条件的表达式可以预先计算好。

函数"gateway_to"可能需要修改，以做类似的比较和判断。

为支持连接在多个网络上的主机，程序可以给每个网络接口设置各自的"my_ip_addr"和"my_ip_mask"，上述代码中的比较和判断也要对每个网络接口进行。

2.3 子网和广播

在没有子网的情况下，因特网协议中只可能有两种广播：广播给指定网络中的所有主机，或者是广播给本身网络的所有主机。后一种方法在主机不知到自己在哪个网络中是很有用。

当使用了子网后，情况就变的复杂了。首先，产生了广播给特定子网的可能性。第二，广播给子网中的所有主机需要附加的机制。最后，“广播给本身网络”的解释变成“广播给本身子网”

这中的实现中必须认识 3 中广播地址以及自己的主机地址：

本身的物理网络

所有位都是 1 的目标地址（255.255.255.255）将使数据包在本地的物理网络中进行广播，网关并不传递这些数据包。

指定的网络

目标地址中有有效的网络地址，而本地地址部分都是 1（例如：36.255.255.255）。

指定的子网

目标地址中的网落地址和子网地址有效，而主机号字段都是 1（比如：36.40.255.255）。

因特网广播的更深入的讨论参看[6]。

一个有助于决定是否使用子网的因素是：某台主机是否需要用一步操作就能给所有主机广播。如果两台主机不在同一网络中，就不可能用一个步骤就给它们广播。

2. 4 决定子网字段的长度

一台主机怎么知道该使用多长的子网字段呢？这个问题和几个“引导程序”的问题很相似：一台主机怎么知道自己的地址以及怎么知道网关的地址。对这三个问题，有两个基本的解决办法：“硬编码”的信息和基于广播的协议。

“硬编码”信息是指主机不通过网络就能获得的信息。可以是编译好的，或者更好的方法是存放在磁盘文件中。但对于不断增加的无盘工作站来说，由于它是从网络启动的，所以两种“硬编码”的方法都不适用。而大多数的局域网技术都支持广播，因此另一个较好的方法是启动的主机广播所需要信息的要求。比如，为了知道自己的因特网地址，可以使用“逆向地址解析协议”[4]。

我们建议将 ICMP[9]协议（因特网信报控制协议）进行扩展，加入一对新的 ICMP 消息类型：“地址格式请求”和“地址格式回复”，和“信息请求”和“信息恢复”消息很相似。细节参看附录 1。

当一台主机启动时，广播新加入的 ICMP 消息“地址格式请求”<3>.网关（或相当于网关的主机）接收到后，回复以“地址格式回复”。如果请求中没有说明是哪台主机发送的（源 IP 地址是 0），则回复消息以广播形式发出。发出请求的主机就能接收到这个消息，从而知道自己的子网字段长度。

在“地址格式请求”中只可能有一个值，所以发出请求的主机就没有必要去匹配请求和回复：就是有多个网关回复也没有关系。我们认为主机不会经常从新启动，所以网络上这两个消息的广播负载是很小的。

如果主机连在好几个局域网上，它需要对每个局域网使用这个协议，除非它能确定（从其中一个网络的回复）几个局域网是在同一个网络中的。在这种情况下，其地址会有相同的子网字段长度。

一个潜在的问题是如果主机重复好多次都没有收到对“地址格式请求”的响应时该怎么办。有三种原因可能导致这种情况：

1. 局域网没有和其他的网络相连（永久的）。
2. 没有使用子网，而且没有主机支持这两个 ICMP 请求。
3. 所有的网关都没有正常工作（暂时的）

第一、二种情况意味着子网字段长度是 0。第三种情况下没法知道其值会是什么：最安全的选择是 0。虽然很可能是错的，但这样不会阻止原来可以成功的数据传送。当网关恢复工作以后，当它收到“地址格式请求”时，就会回复，主机就可以获得正确的信息，并将自身的数据相应的调整。主机和网关不应该发送基于“猜”出的“地址格式回复”。

最后，要注意并不要求主机使用这两个 ICMP 协议消息来获得子网字段长度，特别是对于有稳定存储介质的主机。

3. 子网路由方法

一个因特网所有主机都要面对的是怎么决定到另一台主机的路由。在有子网的情况下，这个问题只需要很小的改变。

使用子网后，路由过程就要处理两个层次。如果目标主机和源主机在同一个网络中，只需要子网间的网关来决定路由。而如果目标主机和源主机在不同的网络中，则需要网络间的网关和子网间的网关共同来决定路由。

幸运的是，许多主机可以使用“缺省”的网关作为所有路由的第一个目标，在回复 ICMP 主机重定向消息时再定义更多的合适的路由。但这种方法对于网关和连在多个网络中的主机来说效率太低，而应该使用路由信息交换协议。这超出了本文档的讨论范围，在没有子网的情况下也存在这个问题。

对于只连在一个网络上的主机，需要找到至少一个邻接的网关。同样，也有两个解决办法：硬编码和广播。邻接网关的问题在不使用子网时也存在，用不用子网对次问题没有影响。

但还存在另一个问题：源主机必须知道数据包是直接发送给目标主机还是要通过网关发送？也就是要知道目标主机和源主机是否在同一个物理网络上。这是路由过程中唯一一处需要知道子网的地方。事实上，如果不使用广播，这也是因特网的实现中唯一需要修改的地方。

所以，有可能不用修改就可以使用现有的方法使之支持子网<4>。这样的实现方法必须具备如下条件：

- 只给连接一个网络的主机使用，也不给网关使用。
- 使用在有广播的局域网中。
- 使用地址解析协议 ARP，如 [7]。
- 不需要维护和网关的连接。

在这种情况下，可以修改子网网关上的地址解析 ARP 服务模块，当它接收到地址解析请求时，检查数据包是否正由最佳路由传递。如果是，则将自己的硬件地址回复给源主机。源主机认为网关的地址就是目标地址，并将数据包发送给这个地址。实际上，网关将接收到这些数据包，并将之传递到目标地址。

这种方法使网关中的处理层次不是很清楚，因为通常情况下，地址解析服务器和路由表没有联系。考虑到这点，这种方法不是非常令人满意。但实现起来相当简单，而且没有显著的性能损失。问题是如果原来的网关出问题后，主机没有办法选择另一个网关。这样，一条在其他方法下可以成功的连接就断了。

不要混淆“基于地址解析协议的子网技术”和“基于地址解析协议的网桥”的简单使用。前者是基于网关能检查 IP 地址从而推导出路由的能力，基于显式的子网的拓扑结构。一小部分的路由功能从主机转移到网关上。而基于地址解析协议的网桥则在不知道主机地址和网络拓扑结构的条件下，知道各主机的位置。

注意：基于地址解析协议的子网技术由于广播的使用而变的复杂。地址解析服务器对目标地址是广播地址的请求作出响应。这样的请求只可能来自于不认识广播地址的主机。这将导致数据包的循环传递。如果在一个物理网络中有 N 个不认识广播地址的主机，那么，生存时间是 T 的数据包将会被重复广播 $T \times N$ 的时间。

4. 例子

这部分我们简要的介绍几个机构的子网使用。

4. 1 斯坦福大学

在斯坦福大学，最初的子网是由于历史原因而引入的。自 1979 年以来（当时因特网协议还没有使用），斯坦福在几个实验性的以太网中使用 Pup 协议，使用了很多 Pup

的网关，所有的主机和网关之间使用广播协议相互交换路由表信息。

当引入因特网协议后，决定采用 8 位长的子网，以使因特网子网的数目和各个以太网中的 Pup 网络数目想匹配。Pup 主机的字段长（也是 8 位）也被作为因特网的主机地址字段的长度

只支持 Pup 的网关作了修改，使其可以根据 Pup 的路由表传递因特网数据包。这种网关不对 IP 包中的‘生存时间’（Time-to-live）字段进行操作。当时传递环路的错误没有表现出来。

因特网主机做了修改以理解子网（有好几种方法，效果都相同）。因为所有主机都已经实现了 Pup，所以因特网的路由表的维护过程和维护 Pup 的一样，只要简单的把 Pup 网络号变成因特网子网号就行了。

加入 10M 以太网后，网关修改为使用地址解析协议（ARP）的，这样可以不用修改主机。

因特网子网在 1982 年早期就开始使用了，当时有大约 330 台主机，18 个子网，18 个子网网关。当只支持 Pup 的网关换成真正的因特网网关后，会引入基于因特网的路由信息交换协议，Pup 逐渐停止使用。

4. 2 麻省理工学院（MIT）

麻省理工学院是第一个有大量局域网连接的使用因特网的地方。当时还没有进行网络分类，如果每一条连接都分配一个网络号的话，会用掉大量的可用地址空间。麻省理工学院决定使用一个网络号，并自己管理地址中余下的 24 位。这些位被分成 3 个 8 位字段：子网字段，保留字段（其值为 0）和主机字段。在此之前麻省理工学院使用的 CHAOS 协议中使用 8 位长子网，所以两个协议中可以使用同样的子网号。而使用 8 位主机地址是因为 CHAOS 协议中大多数的硬件都使用 8 位地址。保留的 8 位则是为以后使用。

最初计划在子网网关间使用动态路由协议，并有好几个协议提出，但没有人愿意去实现一个，因此静态路由表依然被使用。

为了解决引入软件需要修改以使其能在子网环境中工作，麻省理工学院想找到一个对 IP 软件做最少修改的模型。这个模型就是 IP 网关发送 ICMP 主机重定向消息，而不是网络重定向消息。所有的麻省理工学院内部 IP 网关都是这样的。因为每一台主机都可以维护非局域网通讯的路由表，这就隐藏了大部分的子网结构。对子网和非子网都适用的主机软件的‘最少调整模型’就是位掩码。

由于其自治性和已安装的软件的关系，以及没有一个优秀的工业标准的原因，麻省理工学院不计划马上使用这个协议，而是使用一套单一的物理连接和包交换机制，和在这套机制上的几个虚拟协议网络。麻省理工学院曾经试图在不同的协议间交换路由信息，以及将一个协议包含在另一个协议中，从中得到一些教训。除了基本的硬件，协议因该是严格独立的。使用 ARP 隐藏子网结构不是非常好，在一个复杂的系统（有环路和不同的连接速率）中，ARP 使地址操作过载。网关间需要一个更复杂的信息互换方法。

4. 3 卡内基-梅隆大学 (CMU)

卡内基-梅隆大学使用一个 B 类网络，网络被分为 11 个物理子网，2 个 3M 的实验以太网，7 个 10M 以太网和 2 个 ProNET 环。虽然分配主机地址时，使第三个 8 位字节相同的地址在同一个子网上，但这只是为了管理方便，而不是必须的。软件不知道这个分配机制。

卡内基-梅隆大学使用一个基于 ARP 的网桥方案。当一台主机发出 ARP 请求，收到的网桥将原来的地址映射放入缓存，并将请求传递给其它的电缆。当网桥收到一个有目标地址的 ARP 回复时，从缓存中寻找将这个回复送到哪条电缆上。这样，网桥尝试将 ARP 协议透明的扩展到不同类的多电缆环境中。这就要求网桥将一条电缆上的广播变成所有连接着的电缆上的广播。所以这个算法只在没有连接环路的网络上可行。将这个简单算法替换为支持冗余路径和减低广播负载的算法的工作正在进行中。

卡内基-梅隆大学使用支持 3M 以太网和 10M proNET 环的 RFC-826 地址解析协议。

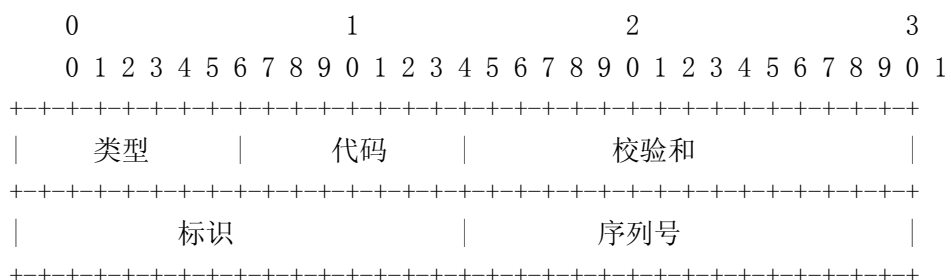
因为卡内基-梅隆大学没有冗余的连接电缆，因此不用关心网桥的崩溃问题。网络上 150 台主机也使网桥有足够的缓存，ARP 广播使用的带宽也不大。

但卡内基-梅隆大学的网络会从单一连接的小网络发展成为有 5000-10000 台主机的连接整个校园的大网络。基于 ARP 的网桥方案不再使用，需要一个有明确子网的系统。中期的目标是建立一个可以引入没有修改的 IP 实现的环境。其目的是尽可能的保持对主机透明的路由机制。

5. 地址格式因特网信报控制协议 (ICMP)

5. 1 描述

地址格式请求或地址格式回复



IP 字段:

地址

地址格式请求消息的源地址就是地址格式回复消息的目的地址。为了构成回复消息，请求的源地址成为回复的目标地址，回复消息的源地址就设成回复者的地址。“类型”设成“A2“，“编码”字段设为子网字段的长度，然后计算校验和。如果请求的源地址是 0，那么回复的目标地址就设成广播地址。

因特网信报控制协议 (ICMP) 字段

类型

A1 表示地址格式请求消息

A2 表示地址格式回复消息

编码

0 代表地址格式请求消息

非 0 代表地址格式回复消息的子网字段长度

校验和

从因特网信报控制协议“类型”字段开始的 16 位的和的余数。计算时，校验和应为 0。其值以后可能被改变。

标识

匹配请求和回复的标识，可以是 0。

序列号

匹配请求和回复的序列号，可以是 0。

收到地址格式请求的网关要回复这个请求。它需要将“编码”字段置为请求的目表地址网络的子网字段的长度。如果请求是广播的，其目标地址就是“这个网络”。子网字段的长度可以是 0- (31-N)，N 是 IP 网络字段的长度 (8, 16 或 24)。如果请求的主机不知道自己的地址，就可能把请求中的源地址置为 0，回复则是广播的。因为一个网络自由一种地址格式，所有就没有必要匹配请求和回复。这种方式应尽量避免，因为它会增加不必要的网络流量。

类型 A1 可能从网关和主机收到

类型 A2 可能从网关和起网关作用的主机收到。

5. 2 例子

下面例子中，我们假设请求主机的地址是 36.40.0.123，网关是 36.40.0.62，处于网络 36.0.0.0 中，使用 8 位子网。

首先，假设广播是允许的，主机发送如下数据包：

源地址： 36. 40. 0. 123
目标地址： 36. 255. 255. 255
协议： ICMP = 1
类型： Address Format Request = A1
编码： 0

36.40.0.62 将收到这个数据包，并回复如下：

源地址： 36. 40. 0. 62
目标地址： 36. 40. 0. 123
协议： ICMP = 1
类型： Address Format Reply = A2
编码： 8

下面的例子假设地址 **255.255.255.255** 表示“广播到这个物理网络”。上面的例子就无能为力了。因为这样的广播可能要广播到多个子网。我们建议的最有效的方法是，主机首先找到自己的地址（可以使用在参考[4]中描述的“反向地址解析协议”），然后将 **ICMP** 请求发送到 **255.255.255.255**。

```
源地址:          36.40.0.123
目标地址:        255.255.255.255
协议:            ICMP = 1
类型:            Address Format Request = A1
编码:            0
```

网关就可以直接回复给请求主机。

假设 **36.40.0.123** 是无盘工作站，并不知道自己的主机号。它可以发送下面数据：

```
源地址:          0.0.0.0
目标地址:        255.255.255.255
协议:            ICMP = 1
类型:            Address Format Request = A1
编码:            0
```

36.40.0.62 将收到这个数据包，并回复：

```
源地址:          36.40.0.62
目标地址:        36.40.255.255
协议:            ICMP = 1
类型:            Address Format Reply = A2
编码:            8
```

注意，网关使用最小的广播范围回复（发送到 **36.255.255.255** 将会在许多子网中广播，而并不单单是需要的子网）。即使这样，这个广播也造成不必要的网络负载。因此我们建议尽量少的使用“匿名（**0.0.0.0**）”源地址。

如果不允许广播，假设主机有邻接网关的硬编码信息，则 **36.40.0.123** 会发送：

```
源地址:          36.40.0.123
目标地址:        36.40.0.62
协议:            ICMP = 1
类型:            Address Format Request = A1
编码:            0
```

36.40.0.62 的回复和上例一样

注意

有些 A 类网络中的主机分配的主机号就是其以太网硬件地址的低 24 位。

我们讨论的因特网广播是基参考[6]的。

如果不支持广播，则假设有主机知道邻接网关地址，并发送 **ICMP** 到这个网关。

这就是前面提到的在同一个网络中，透明子网和显式子网的并存。

参考

- D.R. Boggs, J.F. Shoch, E.A. Taft, 和 R.M. Metcalfe 著,《Pup: 一种因特网结构》, IEEE 通讯学报, COM-28, 4, 612-624 页, 1980 年 4 月
- David D. Clark 著《名字, 地址, 端口和路由》, RFC-814, MIT-LCS, 1982 年 7 月
- Yogan K. Dalal 和 Robert M. Metcalfe 著,《广播数据包的反向传递》, Comm. ACM 21, 12,1040-1048 页, 1978 年 12 月
- Ross Finlayson, Timothy Mann, Jeffrey Mogul 和 Marvin Theimer 著,《逆向地址解析协议》, RFC-903, 斯坦福大学, 1984 年 6 月
- R.M. Metcalfe 和 D.R. Boggs 著,《以太网: 分布式的局域网包交换》, Comm. ACM 19,7, 395-404 页, 1976 年 7 月
- Jeffrey Mogul 著, 《广播数据包》, RFC-919, 斯坦福大学, 1984 年 10 月
- Da Jeffrey Mogul 著,《一种以太网地址解析协议》, RFC-826, 1982 年 9 月
- Jon Postel 著,《因特网协议》, RFC-791, USC-ISI, 1982 年 9 月。
- Jon Postel 著,《因特网信报控制协议》, RFC-792, USC-ISI, 1981 年 9 月