

第12章 广播和多播

12.1 引言

在第1章中我们提到有三种 IP 地址：单播地址、广播地址和多播地址。本章将更详细地介绍广播和多播。

广播和多播仅应用于 UDP，它们对需将报文同时传往多个接收者的应用来说十分重要。TCP 是一个面向连接的协议，它意味着分别运行于两主机（由 IP 地址确定）内的两进程（由端口号确定）间存在一条连接。

考虑包含多个主机的共享信道网络如以太网。每个以太网帧包含源主机和目的主机的以太网地址（48bit）。通常每个以太网帧仅发往单个目的主机，目的地址指明单个接收接口，因而称为单播(unicast)。在这种方式下，任意两个主机的通信不会干扰网内其他主机（可能引起争夺共享信道的情况除外）。

然而，有时一个主机要向网上的所有其他主机发送帧，这就是广播。通过 ARP 和 RARP 可以看到这一过程。多播(multicast) 处于单播和广播之间：帧仅传送给属于多播组的多个主机。

为了弄清广播和多播，需要了解主机对由信道传送过来帧的过滤过程。图 12-1 说明了这一过程。

首先，网卡查看由信道传送过来的帧，确定是否接收该帧，若接收后就将它传往设备驱动程序。通常网卡仅接收那些目的地址为网卡物理地址或广播地址的帧。另外，多数接口均被设置为混合模式，这种模式能接收每个帧的一个复制。作为一个例子，tcpdump 使用这种模式。

目前，大多数的网卡经过配置都能接收目的地址为多播地址或某些子网多播地址的帧。对于以太网，当地址中最高字节的最低位设置为 1 时表示该地址是一个多播地址，用十六进制可表示为 01:00:00:00:00:00（以太网广播地址 ff:ff:ff:ff:ff:ff 可看作是以太网多播地址的特例）。

如果网卡收到一个帧，这个帧将被传送给设备驱动程序（如果帧检验和错，网卡将丢弃该帧）。设备驱动程序将进行另外的帧过滤。首先，帧类型中必须指定要使用的协议（IP、ARP 等等）。其次，进行多播过滤来检测该主机是否属于多播地址说明的多播组。

设备驱动程序随后将数据帧传送给下一层，比如，当帧类型指定为 IP 数据报时，就传往 IP 层。IP 根据 IP 地址中的源地址和目的地址进行更多的过滤检测。如果正常，就将数据报传送给下一层（如 TCP 或 UDP）。

每次 UDP 收到由 IP 传送来的数据报，就根据目的端口号，有时还有源端口号进行数据报

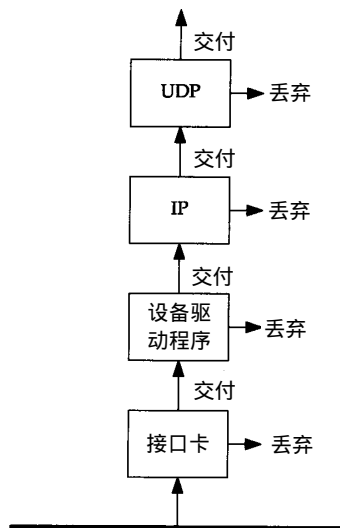


图12-1 协议栈各层对收到帧的过滤过程

过滤。如果当前没有进程使用该目的端口号，就丢弃该数据报并产生一个 ICMP 不可达报文（TCP 根据它的端口号作相似的过滤）。如果 UDP 数据报存在检验和错，将被丢弃。

使用广播的问题在于它增加了对广播数据不感兴趣主机的处理负荷。拿一个使用 UDP 广播应用作为例子。如果网内有 50 个主机，但仅有 20 个参与该应用，每次这 20 个主机中的一个发送 UDP 广播数据时，其余 30 个主机不得不处理这些广播数据报。一直到 UDP 层，收到的 UDP 广播数据报才会被丢弃。这 30 个主机丢弃 UDP 广播数据报是因为这些主机没有使用这个目的端口。

多播的出现减少了对应用不感兴趣主机的处理负荷。使用多播，主机可加入一个或多个多播组。这样，网卡将获悉该主机属于哪个多播组，然后仅接收主机所在多播组的那些多播帧。

12.2 广播

在图 3-9 中，我们知道了四种 IP 广播地址，下面对它们进行更详细的介绍。

12.2.1 受限的广播

受限的广播地址是 255.255.255.255。该地址用于主机配置过程中 IP 数据报的目的地址，此时，主机可能还不知道它所在网络的网络掩码，甚至连它的 IP 地址也不知道。

在任何情况下，路由器都不转发目的地址为受限的广播地址的数据报，这样的数据报仅出现在本地网络中。

一个未解的问题是：如果一个主机是多接口的，当一个进程向本网广播地址发送数据报时，为实现广播，是否应该将数据报发送到每个相连的接口上？如果不是这样，想对主机所有接口广播的应用必须确定主机中支持广播的所有接口，然后向每个接口发送一个数据报复制。

大多数 BSD 系统将 255.255.255.255 看作是配置后第一个接口的广播地址，并且不提供向所属具备广播能力的接口传送数据报的功能。不过，`routed`（见 10.3 节）和 `rwhod`（BSD `rwho` 客户的服务器）是向每个接口发送 UDP 数据报的两个应用程序。这两个应用程序均用相似的启动过程来确定主机中的所有接口，并了解哪些接口具备广播能力。同时，将对应于那种接口的指向网络的广播地址作为发往该接口的数据报的目的地址。

Host Requirements RFC 没有进一步涉及多接口主机是否应当向其所有的接口发送受限的广播。

12.2.2 指向网络的广播

指向网络的广播地址是主机号为全 1 的地址。A 类网络广播地址为 `netid.255.255.255`，其中 `netid` 为 A 类网络的网络号。

一个路由器必须转发指向网络的广播，但它也必须有一个不进行转发的选择。

12.2.3 指向子网的广播

指向子网的广播地址为主机号为全 1 且有特定子网号的地址。作为子网直接广播地址的 IP 地址需要了解子网的掩码。例如，如果路由器收到发往 128.1.2.255 的数据报，当 B 类网络

128.1的子网掩码为255.255.255.0时, 该地址就是指向子网的广播地址; 但如果该子网的掩码为255.255.254.0, 该地址就不是指向子网的广播地址。

12.2.4 指向所有子网的广播

指向所有子网的广播也需要了解目的网络的子网掩码, 以便与指向网络的广播地址区分开。指向所有子网的广播地址的子网号及主机号为全 1。例如, 如果目的子网掩码为255.255.255.0, 那么IP地址128.1.255.255是一个指向所有子网的广播地址。然而, 如果网络没有划分子网, 这就是一个指向网络的广播。

当前的看法[Almquist 1993]是这种广播是陈旧过时的, 更好的方式是使用多播而不是对所有子网的广播。

[Almquist 1993] 指出RFC 922要求将一个指向所有子网的广播传送给所有子网, 但当前的路由器没有这么做。这很幸运, 因为一个因错误配置而没有子网掩码的主机会把它的本地广播传送到所有子网。例如, 如果IP地址为128.1.2.3的主机没有设置子网掩码, 它的广播地址在正常情况下的默认值是 128.1.255.255。但如果子网掩码被设置为255.255.255.0, 那么由错误配置的主机发出的广播将指向所有的子网。

1983年问世的4.2BSD是第一个影响广泛的TCP/IP的实现, 它使用主机号全0作为广播地址。一个最早提到广播IP地址的是IEN 212 [Gurwitz and Hinden 1982], 它提出用主机号中的1比特来表示IP广播地址 (IENs 是互联网试验注释, 基本上是RFC的前身)。RFC 894 [Hornig 1984]认为4.2BSD使用不标准的广播地址, 但RFC 906 [Finlayson 1984]注意到对广播地址还没有Internet标准。RFC编辑在RFC 906中加了一个脚注承认缺少标准的广播地址, 并强烈推荐将主机号全1作为广播地址。尽管1986年的4.3BSD采用主机号全1表示广播地址, 但直到90年代早期, 操作系统 (著名的是SunOS 4.x) 还继续使用非标准的广播地址。

12.3 广播的例子

广播是怎样传送的? 路由器及主机又如何处理广播? 很遗憾, 这是难以回答的问题, 因为它依赖于广播的类型、应用的类型、TCP/IP实现方法以及有关路由器的配置。

首先, 应用程序必须支持广播。如果执行

```
sun % ping 255.255.255.255
/usr/etc/ping: unknown host 255.255.255.255
```

打算在本地电缆上进行广播。但它无法进行, 原因在于该应用程序 (ping) 中存在一个程序设计上的问题。大多数应用程序收到点分十进制的 IP地址或主机名后, 会调用函数 `inet_addr(3)` 来把它们转化为 32 bit 的二进制IP地址。假定要转化的是一个主机名, 如果转化失败, 该库函数将返回 - 1 来表明存在某种差错 (例如是字符而不是数字或串中有小数点)。但本网广播地址 (255.255.255.255) 也被当作存在差错而返回 - 1。大多数程序均假定接收到的字符串是主机名, 然后查找 DNS (第14章), 失败后输出差错信息如“未知主机”。

如果我们修复 ping 程序中这个欠缺, 结果也并不总是令人满意的。在 6 个不同系统的测试中, 仅有一个像预期的那样产生了一个本网广播数据报。大多数则在路由表中查找 IP地址 255.255.255.255, 而该地址被用作默认路由器地址, 因此向默认路由器单播一个数据报。最

终该数据报被丢弃。

指向子网的广播是我们应该使用的。在6.3节中，我们向测试网络（见扉页前图）中IP地址为140.252.13.63的以太网发送数据报，并接收以太网中所有主机的应答。与子网广播地址关联的每个接口是用于命令ifconfig（见3.8节）的值。如果我们ping那个地址，预期的结果是：

```
sun % arp -a                                ARP高速缓存空

sun % ping 140.252.13.63
PING 140.252.13.63: 56 data bytes
64 bytes from sun (140.252.13.33): icmp_seq=0. time=4. ms
64 bytes from bsdi (140.252.13.35): icmp_seq=0. time=172. ms
64 bytes from svr4 (140.252.13.34): icmp_seq=0. time=192. ms

64 bytes from sun (140.252.13.33): icmp_seq=1. time=1. ms
64 bytes from bsdi (140.252.13.35): icmp_seq=1. time=52. ms
64 bytes from svr4 (140.252.13.34): icmp_seq=1. time=90. ms
^?                                           键入中断以停止显示
----140.252.13.63 PING Statistics----
2 packets transmitted, 6 packets received, -200% packet loss
round-trip (ms)  min/avg/max = 1/85/192
sun % arp -a                                再检验ARP缓存
svr4 (140.252.13.34) at 0:0:c0:c2:9b:26
bsdi (140.252.13.35) at 0:0:c0:6f:2d:40
```

IP通过目的地址（140.252.13.63）来确定，这是指向子网的广播地址，然后向链路层的广播地址发送该数据报。

在6.3节提到的这种广播类型的接收对象为局域网中包括发送主机在内的所有主机，因此可以看到除了收到网内其他主机的答复外，还收到来自发送主机（sun）的答复。

在这个例子中，我们也显示了执行ping广播地址前后ARP缓存的内容。这可以显示广播与ARP之间的相互作用。执行ping命令前ARP缓存是空的，而执行后是满的（也就是说，对网内其他每个响应回显请求的主机在ARP缓存中均有一个条目）。我们提到的该以太网数据帧被传送到链路层的广播地址（0xffffffff）是如何发生的呢？由sun主机发送的数据帧不需要ARP。

如果使用tcpdump来观察ping的执行过程，可以看到广播数据帧的接收者在发送它的响应之前，首先产生一个对sun主机的ARP请求，因为它的应答是单播的。在4.5节我们介绍了一个ARP请求的接收者（该例中是sun）通常在发送ARP应答外，还将请求主机的IP地址和物理地址加入到ARP缓存中去。这基于这样一个假定：如果请求者向我们发送一个数据报，我们也很可能想向它发回什么。

我们使用的ping程序有些特殊，原因在于它使用的编程接口（在大多数Unix实现中是低级插口(raw socket)）通常允许向一个广播地址发送数据报。如果使用不支持广播的应用如TFTP，情况又如何呢？（TFTP将在第15章详细介绍。）

```
bsdi % tftp                                启动客户程序
tftp> connect 140.252.13.63                说明服务器的IP地址
tftp> get temp.foo                          试图从服务器或获取一个文件
tftp: sendto: Permission denied
tftp> quit                                  终止客户程序
```

在这个例子中，程序立即产生了一个差错，但不向网络发送任何信息。产生这一切的原因在于，插口提供的应用程序接口API只有在进程明确打算进行广播时才允许它向广播地址发送UDP

数据报。这主要是为了防止用户错误地采用了广播地址（正如此例）而应用程序却不打算广播。

在广播UDP数据报之前，使用插口中API的应用程序必须设置SO_BROADCAST插口选项。

并非所有系统均强制使用这个限制。某些系统中无需进程进行这个说明就能广播UDP数据报。而某些系统则有更多的限制，需要有超级用户权限的进程才能广播。

下一个问题是是否转发广播数据。有些系统内核和路由器有一选项来控制允许或禁止这一特性（见附录E）。

如果让路由器bsdi能够转发广播数据，然后在主机slip上运行ping程序，就能够观察到由路由器bsdi转发的子网广播数据报。转发广播数据报意味着路由器接收广播数据，确定该目的地址是对哪个接口的广播，然后用链路层广播向对应的网络转发数据报。

```
slip % ping 140.252.13.63
PING 140.252.13.63 (140.252.13.63): 56 data bytes
64 bytes from 140.252.13.35: icmp_seq=0 ttl=255 time=190 ms
64 bytes from 140.252.13.33: icmp_seq=0 ttl=254 time=280 ms (DUP!)
64 bytes from 140.252.13.34: icmp_seq=0 ttl=254 time=360 ms (DUP!)

64 bytes from 140.252.13.35: icmp_seq=1 ttl=255 time=180 ms
64 bytes from 140.252.13.33: icmp_seq=1 ttl=254 time=270 ms (DUP!)
64 bytes from 140.252.13.34: icmp_seq=1 ttl=254 time=360 ms (DUP!)

^?                               键入中断以停止显示
--- 140.252.13.63 ping statistics ---
3 packets transmitted, 2 packets received, +4 duplicates, 33% packet loss
round-trip min/avg/max = 180/273/360 ms
```

我们观察到它的确正常工作了，同时也看到BSD系统中的ping程序检查重复的数据报序列号。如果出现重复序列号的数据报就显示DUP!，这意味着一个数据报已经在某处重复了，然而它正是我们所期望看到的，因为我们正向一个广播地址发送数据。

我们还可以从远离广播所指向的网络上的主机上来进行这个试验。在主机angogh.cx.berkeley.edu（和我们的网络距离14跳）上运行ping程序，如果路由器sun被设置为能够转发所指向的广播，它还能正常工作。在这种情况下，这个IP数据报（传送ICMP回显请求）被路径上的每个路由器像正常的数据报一样转发，它们均不知道传送的实际上是广播数据。接着最后一个路由器netb看到主机号为63，就将其转发给路由器sun。路由器sun觉察到该目的IP地址事实上是一个相连子网接口上的广播地址，就将该数据报以链路层广播传往相应网络。

广播是一种应该谨慎使用的功能。在许多情况下，IP多播被证明是一个更好的解决办法。

12.4 多播

IP多播提供两类服务：

1) 向多个目的地址传送数据。有许多向多个接收者传送信息的应用：例如交互式会议系统和向多个接收者分发邮件或新闻。如果不采用多播，目前这些应用大多采用TCP来完成（向每个目的地址传送一个单独的数据复制）。然而，即使使用多播，某些应用可能继续采用TCP来保证它的可靠性。

2) 客户对服务器的请求。例如，无盘工作站需要确定启动引导服务器。目前，这项服务是通过广播来提供的（正如第16章的BOOTP），但是使用多播可降低不提供这项服务主机的负担。

12.4.1 多播组地址

图12-2显示了D类IP地址的格式。

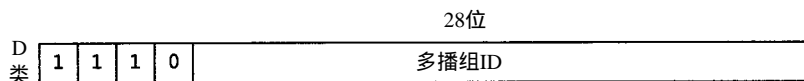


图12-2 D类IP地址格式

不像图1-5所示的其他三类IP地址（A、B和C），分配的28 bit均用作多播组号而不再表示其他。

多播组地址包括为1110的最高4 bit和多播组号。它们通常可表示为点分十进制数，范围从224.0.0.0到239.255.255.255。

能够接收发往一个特定多播组地址数据的主机集合称为主机组（host group）。一个主机组可跨越多个网络。主机组中成员可随时加入或离开主机组。主机组中对主机的数量没有限制，同时不属于某一主机组的主机可以向该组发送信息。

一些多播组地址被IANA确定为知名地址。它们也被当作永久主机组，这和TCP及UDP中的熟知端口相似。同样，这些知名多播地址在RFC最新分配数字中列出。注意这些多播地址所代表的组是永久组，而它们的组成员却不是永久的。

例如，224.0.0.1代表“该子网内的所有系统组”，224.0.0.2代表“该子网内的所有路由器组”。多播地址224.0.1.1用作网络时间协议NTP，224.0.0.9用作RIP-2（见10.5节），224.0.1.2用作SGI公司的dogfight应用。

12.4.2 多播组地址到以太网地址的转换

IANA拥有一个以太网地址块，即高位24 bit为00:00:5e（十六进制表示），这意味着该地址块所拥有的地址范围从00:00:5e:00:00:00到00:00:5e:ff:ff:ff。IANA将其中的一半分配为多播地址。为了指明一个多播地址，任何一个以太网地址的首字节必须是01，这意味着与IP多播相对应的以太网地址范围从01:00:5e:00:00:00到01:00:5e:7f:ff:ff。

这里对CSMA/CD或令牌网使用的是Internet标准比特顺序，和在内存中出现的比特顺序一样。这也是大多数程序设计员和系统管理员采用的顺序。IEEE文档采用了这种比特传输顺序。Assigned Numbers RFC给出了这些表示的差别。

这种地址分配将使以太网多播地址中的23bit与IP多播组号对应起来，通过将多播组号中的低位23bit映射到以太网地址中的低位23bit实现，这个过程如图12-3所示。

由于多播组号中的最高5 bit在映射过程中被忽略，因此每个以太网多播地址对应的多播组是不唯一的。32个不同的多播组号被映射为一个以太网地址。例如，多播地址224.128.64.32（十六进制e0.80.40.20）和224.0.64.32（十六进制e0.00.40.20）都映射为同一以太网地址01:00:5e:00:40:20。

既然地址映射是不唯一的，那么设备驱动程序或IP层（见图12-1）就必须对数据报进行过滤。因为网卡可能接收到主机不想接收的多播数据帧。另外，如果网卡不提供足够的多播数据帧过滤功能，设备驱动程序就必须接收所有多播数据帧，然后对它们进行过滤。

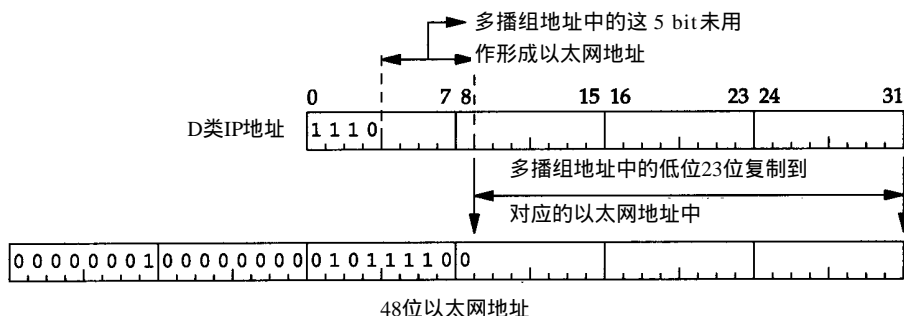


图12-3 D类IP地址到以太网多播地址的映射

局域网网卡趋向两种处理类型：一种是网卡根据对多播地址的散列值实行多播过滤，这意味仍会接收到不想接收的多播数据；另一种是网卡只接收一些固定数目的多播地址，这意味着当主机想接收超过网卡预先支持多播地址以外的多播地址时，必须将网卡设置为“多播混杂(multicast promiscuous)”模式。因此，这两种类型的网卡仍需要设备驱动程序检查收到的帧是否真是主机所需要的。

即使网卡实现了完美的多播过滤（基于48 bit的硬件地址），由于从D类IP地址到48 bit的硬件地址的映射不是一对一的，过滤过程仍是必要的。

尽管存在地址映射不完美和需要硬件过滤的不足，多播仍然比广播好。

单个物理网络的多播是简单的。多播进程将目的IP地址指明为多播地址，设备驱动程序将它转换为相应的以太网地址，然后把数据发送出去。这些接收进程必须通知它们的IP层，它们想接收的发给给定多播地址的数据报，并且设备驱动程序必须能够接收这些多播帧。这个过程就是“加入一个多播组”（使用“接收进程”复数形式的原因在于对一确定的多播信息，在同一主机或多个主机上存在多个接收者，这也是为什么要首先使用多播的原因）。当一个主机收到多播数据报时，它必须向属于那个多播组的每个进程均传送一个复制。这和单个进程收到单播UDP数据报的UDP不同。使用多播，一个主机上可能存在多个属于同一多播组的进程。

当把多播扩展到单个物理网络以外需要通过路由器转发多播数据时，复杂性就增加了。需要有一个协议让多播路由器了解确定网络中属于确定多播组的任何一个主机。这个协议就是Internet组管理协议（IGMP），也是下一章介绍的内容。

12.4.3 FDDI和令牌环网络中的多播

FDDI网络使用相同的D类IP地址到48 bit FDDI地址的映射过程[Katz 1990]。令牌环网络通常使用不同的地址映射方法，这是因为大多数令牌控制中的限制。

12.5 小结

广播是将数据报发送到网络中的所有主机（通常是本地相连的网络），而多播是将数据报发送到网络的一个主机组。这两个概念的基本点在于当收到送往上一个协议栈的数据帧时采用不同类型的过滤。每个协议层均可以因为不同的理由丢弃数据报。

目前有四种类型的广播地址：受限的广播、指向网络的广播、指向子网的广播和指向所有子网的广播。最常用的是指向子网的广播。受限的广播通常只在系统初始启动时才会用到。

试图通过路由器进行广播而发生的问题，常常是因为路由器不了解目的网络的子网掩码。结果与多种因素有关：广播地址类型、配置参数等等。

D类IP地址被称为多播组地址。通过将其低位 23 bit映射到相应以太网地址中便可实现多播组地址到以太网地址的转换。由于地址映射是不唯一的，因此需要其他的协议实现额外的数据报过滤。

习题

- 12.1 广播是否增加了网络通信量？
- 12.2 考虑一个拥有50台主机的以太网：20台运行TCP/IP，其他30台运行其他的协议族。主机如何处理来自运行另一个协议族主机的广播？
- 12.3 登录到一个过去从来没有用过的 Unix系统，并且打算找出所有支持广播的接口的指向子网的广播地址。如何做到这点？
- 12.4 如果我们用ping程序向一个广播地址发送一个长的分组，如

```
sun % ping 140.252.13.63 1472
PING 140.252.13.63: 1472 data bytes
1480 bytes from sun (140.252.13.33): icmp_seq=0. time=6. ms
1480 bytes from svr4 (140.252.13.34): icmp_seq=0. time=84. ms
1480 bytes from bsdi (140.252.13.35): icmp_seq=0. time=128. ms
```

它正常工作，但将分组的长度再增加一个字节后出现如下差错：

```
sun % ping 140.252.13.63 1473
PING 140.252.13.63: 1473 data bytes
sendto: Message too long
```

究竟出了什么问题？

- 12.5 重做习题 10.6，假定8个RIP报文是通过多播而不是广播（使用 RIP 版本2）。有什么变化？