

第13章 IGMP：Internet组管理协议

13.1 引言

12.4节概述了IP多播给出，并介绍了D类IP地址到以太网地址的映射方式。也简要说明了在单个物理网络中的多播过程，但当涉及多个网络并且多播数据必须通过路由器转发时，情况会复杂得多。

本章将介绍用于支持主机和路由器进行多播的Internet组管理协议（IGMP）。它让一个物理网络上的所有系统知道主机当前所在的多播组。多播路由器需要这些信息以便知道多播数据报应该向哪些接口转发。IGMP在RFC 1112中定义 [Deering 1989]。

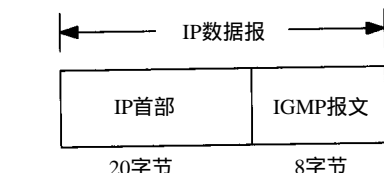


图13-1 IGMP报文封装在IP数据报中

正如ICMP一样，IGMP也被当作IP层的一部分。IGMP报文通过IP数据报进行传输。不像我们已经见到的其他协议，IGMP有固定的报文长度，没有可选数据。图13-1显示了IGMP报文如何封装在IP数据报中。

IGMP报文通过IP首部中协议字段值为2来指明。

13.2 IGMP报文

图13-2显示了长度为8字节的IGMP报文格式。

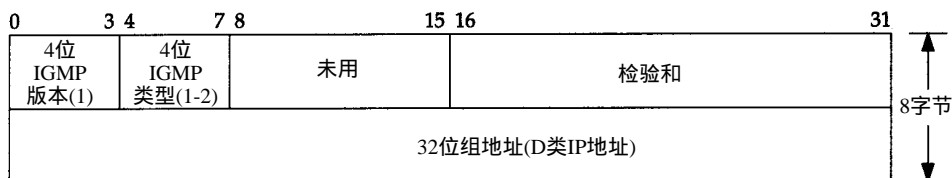


图13-2 IGMP报文的字段格式

这是版本为1的IGMP。IGMP类型为1说明是由多播路由器发出的查询报文，为2说明是主机发出的报告报文。检验和的计算和ICMP协议相同。

组地址为D类IP地址。在查询报文中组地址设置为0，在报告报文中组地址为要参加的组地址。在下一节中，当介绍IGMP如何操作时，我们将会更详细地了解它们。

13.3 IGMP 协议

13.3.1 加入一个多播组

多播的基础就是一个进程的概念（使用的术语进程是指操作系统执行的一个程序），该进程在一个主机的给定接口上加入了一个多播组。在一个给定接口上的多播组中的成员是动态

的——它随时因进程加入和离开多播组而变化。

这里所指的进程必须以某种方式在给定的接口上加入某个多播组。进程也能离开先前加入的多播组。这些是一个支持多播主机中任何 API所必需的部分。使用限定词“接口”是因为多播组中的成员是与接口相关联的。一个进程可以在多个接口上加入同一多播组。

Stanford大学伯克利版Unix中的IP 多播详细说明了有关socket API的变化，这些变化在Solaris 2.x和ip(7)的文档中也提供了。

这里暗示一个主机通过组地址和接口来识别一个多播组。主机必须保留一个表，此表中包含所有至少含有一个进程的多播组以及多播组中的进程数量。

13.3.2 IGMP 报告和查询

多播路由器使用IGMP报文来记录与该路由器相连网络中组成员的变化情况。使用规则如下：

- 1) 当第一个进程加入一个组时，主机就发送一个 IGMP报告。如果一个主机的多个进程加入同一组，只发送一个IGMP报告。这个报告被发送到进程加入组所在的同一接口上。
- 2) 进程离开一个组时，主机不发送 IGMP报告，即便是组中的最后一个进程离开。主机知道在确定的组中已不再有组成员后，在随后收到的 IGMP查询中就不再发送报告报文。
- 3) 多播路由器定时发送 IGMP查询来了解是否还有任何主机包含有属于多播组的进程。多播路由器必须向每个接口发送一个 IGMP查询。因为路由器希望主机对它加入的每个多播组均发回一个报告，因此IGMP查询报文中的组地址被设置为 0。
- 4) 主机通过发送 IGMP报告来响应一个 IGMP查询，对每个至少还包含一个进程的组均要发回IGMP报告。

使用这些查询和报告报文，多播路由器对每个接口保持一个表，表中记录接口上至少还包含一个主机的多播组。当路由器收到要转发的多播数据报时，它只将该数据报转发到（使用相应的多播链路层地址）还拥有属于那个组主机的接口上。

图13-3显示了两个IGMP报文，一个是主机发送的报告，另一个是路由器发送的查询。该路由器正在要求那个接口上的每个主机说明它加入的每个多播组。

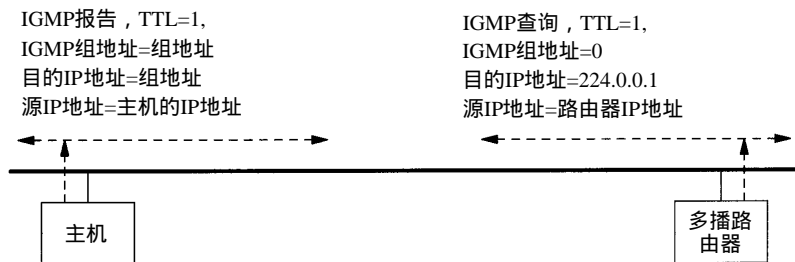


图13-3 IGMP的报告和查询

对TTL字段我们将在本节的后面介绍。

13.3.3 实现细节

为改善该协议的效率，有许多实现的细节要考虑。首先，当一个主机首次发送 IGMP报告

(当第一个进程加入一个多播组)时,并不保证该报告被可靠接收(因为使用的是IP交付服务)。下一个报告将在间隔一段时间后发送。这个时间间隔由主机在 0~10秒的范围内随机选择。

其次,当一个主机收到一个从路由器发出的查询后,并不立即响应,而是经过一定的时间间隔后才发出一些响应(采用“响应”的复数形式是因为该主机必须对它参加的每个组均发送一个响应)。既然参加同一多播组的多个主机均能发送一个报告,可将它们的发送间隔设置为随机时延。在一个物理网络中的所有主机将收到同组其他主机发送的所有报告,因为如图13-3所示的报告中的目的地址是那个组地址。这意味着如果一个主机在等待发送报告的过程中,却收到了发自其他主机的相同报告,则该主机的响应就可以不必发送了。因为多播路由器并不关心有多少主机属于该组,而只关心该组是否还至少拥有一个主机。的确,一个多播路由器甚至不关心哪个主机属于一个多播组。它仅仅想知道在给定的接口上的多播组中是否还至少有一个主机。

在没有任何多播路由器的单个物理网络中,仅有的 IGMP通信量就是在主机加入一个新的多播组时,支持IP多播的主机所发出的报告。

13.3.4 生存时间字段

在图13-3中,我们注意到IGMP报告和查询的生存时间(TTL)均设置为1,这涉及到IP首部中的TTL字段。一个初始TTL为0的多播数据报将被限制在同一主机。在默认情况下,待传多播数据报的TTL被设置为1,这将使多播数据报仅局限在同一子网内传送。更大的TTL值能被多播路由器转发。

回顾6.2节,对发往一个多播地址的数据报从不会产生ICMP差错。当TTL值为0时,多播路由器也不产生ICMP“超时”差错。

在正常情况下,用户进程不关心传出数据报的TTL。然而,一个例外是Traceroute程序(第8章),它主要依据设置TTL值来完成。既然多播应用必须能够设置要传送数据报的TTL值,这意味着程序设计接口必须为用户进程提供这种能力。

通过增加TTL值的方法,一个应用程序可实现对一个特定服务器的扩展环搜索(expanding ring search)。第一个多播数据报以TTL等于1发送。如果没有响应,就尝试将TTL设置为2,然后3,等等。在这种方式下,该应用能找到以跳数来度量的最近的服务器。

从224.0.0.0到224.0.0.255的特殊地址空间是打算用于多播范围不超过1跳的应用。不管TTL值是多少,多播路由器均不转发目的地址为这些地址中的任何一个地址的数据报。

13.3.5 所有主机组

在图13-3中,我们看到了路由器的IGMP查询被送到目的IP地址224.0.0.1。该地址被称为所有主机组地址。它涉及在一个物理网络中的所有具备多播能力的主机和路由器。当接口初始化后,所有具备多播能力接口上的主机均自动加入这个多播组。这个组的成员无需发送IGMP报告。

13.4 一个例子

现在我们已经了解了一些IP多播的细节,再来看看所包含的信息。我们使sun主机能够支

持多播，并将采用一些多播软件所提供的测试程序来观察具体的过程。

首先，采用一个经过修改的 `netstat` 命令来报告每个接口上的多播组成员情况（在 3.9 节显示了 `netstat -ni` 命令的输出结果）。在下面的输出中，用黑体表示有关的多播组。

```
sun % netstat -nia
Name  Mtu  Network  Address          IpKts Ierrs  OpKts Oerrs  Coll
le0   1500  140.252.13. 140.252.13.33    4370   0      4924   0      0
      224.0.0.1
      08:00:20:03:f6:42
      01:00:5e:00:00:01
sl0   552   140.252.1  140.252.1.29     13587   0      15615   0      0
      224.0.0.1
lo0   1536  127      127.0.0.1        1351    0      1351    0      0
      224.0.0.1
```

其中，`-n` 参数将以数字形式显示 IP 地址（而不是按名字来显示它们），`-i` 参数将显示接口的统计结果，`-a` 参数将显示所有配置的接口。

输出结果中的第 2 行 `le0`（以太网）显示了这个接口属于主机组 `224.0.0.1`（“所有主机”），和两行地址，后一行显示相应的以太网地址为：`01:00:5e:00:00:01`。这正是我们期望看到的以太网地址，和 12.4 节介绍的地址映射一致。我们还看到其他两个支持多播的接口：`SLIP` 接口 `sl0` 和回送接口 `lo0`，它们也属于所有主机组。

我们也必须显示 IP 路由表，用于多播的路由表同正常的路由表一样。黑体表项显示了所有传往 `224.0.0.0` 的数据报均被送往以太网：

```
sun % netstat -rn
Routing tables
Destination      Gateway          Flags    Refcnt  Use    Interface
140.252.13.65    140.252.13.35   UGH      0       32     le0
127.0.0.1        127.0.0.1       UH       1       381    lo0
140.252.1.183    140.252.1.29    UH       0       6      sl0
default          140.252.1.183   UG       0       328    sl0
224.0.0.0      140.252.13.33   U       0       66     le0
140.252.13.32    140.252.13.33   U        8      5581    le0
```

如果将这个路由表与 9.2 节中 `sun` 路由器的路由表作比较，会发现只是多了有关多播的条目。

现在使用一个测试程序来让我们能在一个接口上加入一个多播组（不再显示使用这个测试程序的过程）。在以太网接口（`140.252.13.33`）上加入多播组 `224.1.2.3`。执行 `netstat` 程序看到内核已加入这个组，并得到期望的以太网地址。用黑体字来突出显示和前面 `netstat` 输出的不同。

```
sun % netstat -nia
Name  Mtu  Network  Address          IpKts Ierrs  OpKts Oerrs  Coll
le0   1500  140.252.13. 140.252.13.33    4374   0      4929   0      0
      224.1.2.3
      224.0.0.1
      08:00:20:03:f6:42
      01:00:5e:01:02:03
      01:00:5e:00:00:01
sl0   552   140.252.1  140.252.1.29     13862   0      15943   0      0
      224.0.0.1
lo0   1536  127      127.0.0.1        1360    0      1360    0      0
      224.0.0.1
```

我们在输出中再次显示了其他两个接口：`sl0` 和 `lo0`，目的是为了重申加入多播组只发生在一个接口上。

图13-4显示了tcpdump对进程加入这个多播组的跟踪过程。

```
1  0.0                8:0:20:3:f6:42 1:0:5e:1:2:3 ip 60:
                        sun > 224.1.2.3: igmp report 224.1.2.3 [ttl 1]

2  6.94 (6.94)        8:0:20:3:f6:42 1:0:5e:1:2:3 ip 60:
                        sun > 224.1.2.3: igmp report 224.1.2.3 [ttl 1]
```

图13-4 当一个主机加入1个多播组时tcpdump 的输出结果

当主机加入多播组时产生第1行的输出显示。第2行是经过时延后的IGMP报告，我们介绍过报告重发的时延是10秒内的随机时延。

在两行中显示硬件地址证实了以太网目的地址就是正确的多播地址。我们也看到了源 IP 地址为相应的 sun 主机地址，而目的 IP 地址是多播组地址。同时，报告的地址和期望的多播组地址是一致的。

最后，我们注意到，正像指明的那样，TTL是1。当TTL的值为0或1时，tcpdump在打印时用方括号将它们括起来，这是因为 TTL在正常情况下均高于这些值。然而，使用多播我们期望看到许多TTL为1的IP数据报。

在这个输出中暗示了一个多播路由器必须接收在它所有接口上的所有多播数据报。路由器无法确定主机可能加入哪个多播组。

多播路由器的例子

继续前面的例子，但我们将在 sun 主机中启动一个多播选路的守护程序。这里我们感兴趣的并不是多播选路协议，而是要研究所交换的 IGMP查询和报告。即使多播选路守护程序只运行在支持多播的主机（sun）上，所有的查询和报告都将在那个以太网上进行多播，所以我们在该以太网中的其他系统中也能观察到它们。

在启动选路守护程序之前，加入另外一个多播组 224.9.9.9，图13-5显示了输出的结果。

```
1  0.0                sun > 224.0.0.4: igmp report 224.0.0.4
2  0.00 ( 0.00)       sun > 224.0.0.1: igmp query
3  5.10 ( 5.10)       sun > 224.9.9.9: igmp report 224.9.9.9
4  5.22 ( 0.12)       sun > 224.0.0.1: igmp query
5  7.90 ( 2.68)       sun > 224.1.2.3: igmp report 224.1.2.3
6  8.50 ( 0.60)       sun > 224.0.0.4: igmp report 224.0.0.4
7  11.70 ( 3.20)      sun > 224.9.9.9: igmp report 224.9.9.9
8  125.51 (113.81)    sun > 224.0.0.1: igmp query
9  125.70 ( 0.19)     sun > 224.9.9.9: igmp report 224.9.9.9
10 128.50 ( 2.80)     sun > 224.1.2.3: igmp report 224.1.2.3
11 129.10 ( 0.60)     sun > 224.0.0.4: igmp report 224.0.0.4
12 247.82 (118.72)    sun > 224.0.0.1: igmp query
13 248.09 ( 0.27)     sun > 224.1.2.3: igmp report 224.1.2.3
14 248.69 ( 0.60)     sun > 224.0.0.4: igmp report 224.0.0.4
15 255.29 ( 6.60)     sun > 224.9.9.9: igmp report 224.9.9.9
```

图13-5 当多播选路守护程序运行时tcpdump 的输出结果

在这个输出中没有包括以太网地址，因为已经证实了它们是正确的。也删去了 TTL等于1的说明，同样因为它们也是我们期望的那样。

当选路守护程序启动时，输出第1行。它发出一个已经加入了组 224.0.0.4的报告。多播地址224.0.0.4是一个知名的地址，它被当前用于多播选路的距离向量多播选路协议 DVMRP

(Distance Vector Multicast Routing Protocol)所使用 (DVMRP在RFC 1075中定义[Waitzman, Partridge, and Deering])。

在该守护程序启动时, 它也发送一个 IGMP查询 (第2行)。该查询的目的 IP地址为 224.0.0.1 (所有主机组), 如图13-3所示。

第一个报告 (第3行) 大约在5秒后收到, 报告给组 224.9.9.9。这是在下一个查询发出之前 (第4行) 收到的唯一报告。当守护程序启动后, 两次查询 (第2行和第4行) 发出的间隔很短, 这是因为守护程序要将其多播路由表尽快建立起来。

第5、6和7行正是我们期望看到的: sun主机针对它所属的每个组发出一个报告。注意组 224.0.0.4是被报告的, 而其他两个组则是明确加入的, 因为只要选路守护程序还在运行, 它始终要属于组224.0.0.4。

下一个查询位于第8行, 大约在前一个查询的2分钟后发出。它再次引发三个我们所期望的报告 (第9、10和11行)。这些报告的时间顺序与前面不同, 因为接收查询和发送报告的时间是随机的。

最后的查询在前一个查询的大约2分钟后发出, 我们再次得到了期望的响应。

13.5 小结

多播是一种将报文发往多个接收者的通信方式。在许多应用中, 它比广播更好, 因为多播降低了不参与通信的主机的负担。简单的主机成员报告协议 (IGMP)是多播的基本模块。

在一个局域网中或跨越邻近局域网的多播需要使用本章介绍的技术。广播通常局限在单个局域网中, 对目前许多使用广播的应用来说, 可采用多播来替代广播。

然而, 多播还未解决的一个问题是在广域网内的多播。[Deering and Cheriton 1990] 提出扩展目前的路由协议来支持多播。9.13节中的[Perlman 1992]讨论了广域网多播的一些问题。

[Casner and Deering 1992] 介绍了使用多播和一个称为 MBONE (多播主干) 的虚拟网络在整个Internet上传送IETF会议的情况。

习题

- 13.1 我们知道主机通过设置随机时延来调度 IGMP的发送。一个局域网中的主机采取什么措施才能避免两台主机产生相同的随机时延?
- 13.2 在[Casner and Deering 1992]中, 他们提到UDP缺少两个通过MBONE传送音频采样数据的条件: 分组失序检测和分组重复检测。你怎样在 UDP上增加这些功能?