

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：田金勇 (tany_tjy308@263.net)

译文发布时间：2001-6-15

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: 1075

D. Waitzman
C. Partridge
BBN STC
S. Deering
Stanford University
November 1988

远距离矢量多播选路协议

(RFC 1075 Distance Vector Multicast Routing Protocol)

1 备忘录状态

本 RFC 描述了一个距离矢量形式的路由选择协议，这个协议用于在互联网上为多播数据报选路。它起源于选路信息协议 (RIP) [1]，并实现了 RFC1054 中所描述的多播。这是一个实验性协议，这次并不推荐它的实现方式。该备忘录可以任意发布。

2 简介

在 IP 网络上多播的草拟标准目前存在[2]，但没有支持网间多播的路由选择协议。本备忘录描述了实验性的路由选择协议，叫做 DVMRP，它实现了网间多播。DVMRP 使 RIP 中的许多特性和在 Deering[3]中所描述的截断方向路径广播 (TRPB) 算法相结合。

DVMRP 是一个“内部网关协议”；适合在自治系统内的使用，但不能在不同的自治系统之间使用。当前开发的 DVMRP 不能用于为非多播数据报选路，因此要想一个路由器既能为多播数据报又能为单播数据报选路，则它必须运行两个分离的路由选择进程。DVMRP 被设计成易于扩展的，可以扩展成为单播数据报选路。

开发 DVMRP 是为了试验[3]中所描述的算法。RIP 用作这次开发的起始点是因为有一个实现版本可用，而且距离矢量算法与连接状态类算法[4]相比较简单的。另外，为了试验穿越不支持多播的网络可行性，开发了一种叫“隧道”的机制

多播转发算法需要构建基于路由信息的树。构建这颗树需要的状态信息比 RIP 被设计能提供的要多。因为 DVMRP 在某些方面比 RIP 复杂的多。已经具有许多所需要的状态的连接状态算法，可能为 Internet 上多播选路和转发提供了更好的基础。

DVMRP 在一个非常重要的方面与 RIP 有不同之处。RIP 按照路由和转发数据报的方

式思考。DVMRP 的目的是为了了解到多播数据报出发地的返回路径。为了将 DVMRP 解释的和 RIP 一致，单词“目的地”用来代替更恰当的“出发地”但读者应该记住数据报并不被转发到这些目的地，而是起源于那里。

本备忘录被组织为下列部分：

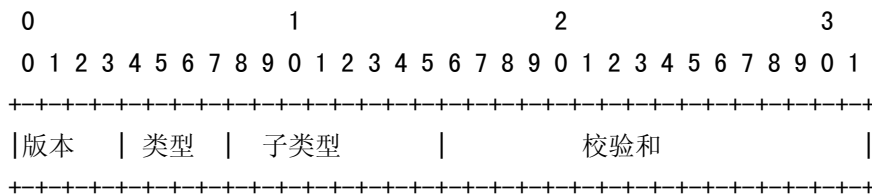
- 对 DVMRP 进行描述。
- 解释隧道。
- 展示路由算法。
- 展示转发算法。
- 列出不同的时间值。
- 说明配置信息。

本备忘录不分析距离矢量路由，也不充分解释距离矢量算法；要想获得这方面主题的更多信息，请参看[1]。在本备忘录中执行路由和转发功能的一个进程或多个进程被称作“路由器”。

3 协议描述

DVMRP 使用 Internet 组管理协议 (IGMP) 交换路由数据报[2]。DVMRP 数据报由两部分组成：一个短的、固定长度的 IGMP 头部，和一个特征数据流。

固定长度的 DVMRP 报文的 IGMP 头部是：



版本是 1。

DVMRP 的类型是 3。

子类型是以下之一：

- 1=应答；报文提供了到一些目的地的路由。
- 2=请求；报文询问到一些目的地的路由。
- 3=非成员报告；报文提供非成员报告。
- 4=非成员取消；报文取消先前非成员报告。

校验和是除了 IP 头部以外，以 16 位对齐的全部报文的反码和的反码。计算校验和时，校验和字段为零。

DVMRP 报文的剩余部分是特征数据流。使用特征数据流的原因是提供易扩充性(通过增加新标签来开发新命令)和减少报文中冗余数据的数量。数据流中的成分被叫做命令，为了便于对齐，它的长度是 16 位的倍数。命令被组织为八位命令数字代码，并至少带有一个八位数据部分。要求所有命令按 16 位对齐。

出现错误的报文将在处理过程中检测到错误的地方被丢弃。任何在错误出现之前由于报文的内容而发生的状态改变，将不会恢复到它原来的值。

某些命令在它们的规范说明中定义了缺省的值。因为缺省值可能会因为协议向前发

展而改变。一个谨慎的实现不会发送依赖缺省值的报文。

DVMRP 报文的长度被限制为 512 字节，这不包括 IP 头部。

3.1 NULL 命令

```
格式: 0 1 2 3 4 5 6 7    0 1 2 3 4 5 6 7
      +-----+-----+
      |           0           | |   忽略   |
      +-----+-----+
      +-----+-----+
```

描述: NULL 命令用来提供附加对齐或填充到 32 位。

3.2 地址家族指示符 (AFI) 命令

```
格式: 0 1 2 3 4 5 6 7    0 1 2 3 4 5 6 7
      +-----+-----+
      |           2           | |   家族   |
      +-----+-----+
      +-----+-----+
```

家族的值:

2=IP 地址家族，它的地址是 32 位长。

缺省: 家族 (Family) =2

描述: AFI 命令为数据流中后继地址提供了地址家族 (直到出现不同的 AFI 命令)。

如果接受者不支持地址家族会出现错误。

3.3 子网掩码 (Subnetmask) 命令

```
格式: 0 1 2 3 4 5 6 7    0 1 2 3 4 5 6 7
      +-----+-----+
      |           3           | | 计数 (count) |
      +-----+-----+
      +-----+-----+
```

附加参数, AFI=IP:

```
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
+-----+-----+
| 子网掩码                                     |
+-----+-----+
```

计数(count)为 0 或 1。

缺省: 假定跟随的路由是到网络的, 使用每条路由的目的地网络掩码的一个掩码。

描述: 子网掩码命令提供了用于后继路由的子网掩码。对子网掩码中的位有一些要求: 0 到 7 位必须为 1, 所有位不应为 1。

如果计数为 0, 则没有子网掩码适用, 假设跟随的路由是到网络的, 使用每个路径的目

的地的网络掩码的一个掩码。如果计数是 1，则子网掩码应该出现在数据流中，并且具有在给定地址族下的合适的长度。

如果计数不为 0 或 1，则出错。

子网掩码不应该被送到适合的网络之外。

要想了解有关 IP 子网的更多的信息请参考[6]。

3.4 度量 (Metric) 命令

```
格式: 0 1 2 3 4 5 6 7    0 1 2 3 4 5 6 7
      +-----+ +-----+
      |         4         | | 值 (value) |
      +-----+ +-----+
```

值(value)是度量单位，它是一个在 1 到 255 之间的无符号值。

缺省：无。

描述：度量命令提供了后继目的地的度量。度量与发送 DVMRP 路由更新的路由器有关。

3.5 flag0 命令

```
格式: 0 1 2 3 4 5 6 7    0 1 2 3 4 5 6 7
      +-----+ +-----+
      |         5         | | 值 (value) |
      +-----+ +-----+
```

值 (value) 中位的意义：

位 7：目的地不可达。

位 6：分裂水平隐藏路径。

缺省：所有位为零。

描述：flags0 命令提供一个设置许多标志的方式。唯一定义的标志——位 6 和位 7——能被用来提供带有无穷大的度量的路由的更多的信息。如果路由器收到了一个它不支持的标志，则应该忽略这个标志。该命令之所以叫做 flag0，是为了允许将来定义附加的标志命令 (flags1, 等等)。

这是一个实验性命令，可能将来会改变。

3.6 无穷大 (Infinity) 命令

```
格式: 0 1 2 3 4 5 6 7    0 1 2 3 4 5 6 7
      +-----+ +-----+
      |         6         | | 值 (value) |
      +-----+ +-----+
```

值 (value) 是无穷大 (Infinity)，它是一个在 1 到 255 之间的无符号值。

缺省：值为 16。

描述：infinity 命令定义流中的后继度量无穷大性。

如果 infinity 为 0，或少于当前的度量值，则出错。

3.7 目的地址 (DA) 命令

```
格式:  0 1 2 3 4 5 6 7   0 1 2 3 4 5 6 7
        +-----+ +-----+
        |           7           | |   计数           |
        +-----+ +-----+
```

计数 (count) 的附加参数数组, AFI=IP:

```
  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
  +-----+
  | 目的地址 1                                     |
  +-----+
```

```
  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
  +-----+
  | 目的地址 2                                     |
  +-----+
```

计数提供的地址的数目, 从 1 到 255。地址的长度依赖于当前地址家族。提供的地址数目受到 512 字节的报文长度限制。

缺省: 无。

描述: DA 命令提供了一个目的地列表。尽管这种格式能表达达到主机的路由, 路由算法仅仅支持网络和子网路由。当前度量 (metric), 无穷大 (infinity), flags0, 子网掩码 (subnetmask) 与一个单一的目的地址结合, 定义了一条路由。当前度量必须少于或等于当前无穷大 (infinity)。

当计数等于零时, 出错。

3.8 请求目的地址 (RDA) 命令

```
格式:  0 1 2 3 4 5 6 7   0 1 2 3 4 5 6 7
        +-----+ +-----+
        |           8           | | 计数 (count) |
        +-----+ +-----+
```

“计数”附加参数数组, AFI=IP:

```
  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
  +-----+
  | 请求目的地址 1                                     |
  +-----+
```

```

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
+-----+
| 请求目的地址 2                                     |
+-----+

```

计数是指提供的地址的数目，从 0 到 255。地址长度依赖于当前地址家族。提供的地址数目受 512 字节报文长度的限制。

缺省：无。

描述：RDA 命令提供了一个路由请求的目的地址列表。为所有路由的路由请求被编码为 count=0。

3.9 非成员报告 (NMR) 命令

```

格式： 0 1 2 3 4 5 6 7    0 1 2 3 4 5 6 7
      +-----+ +-----+
      |          9          | | 计数 (count) |
      +-----+ +-----+

```

计数 (count) 的附加参数数组，AFI=IP：

```

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
+-----+
| 多播地址 1                                     |
+-----+

```

```

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
+-----+
| 保持时间 1                                     |
+-----+

```

```

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
+-----+
| 多播地址 2                                     |
+-----+

```

```

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
+-----+
| 保持时间 2                                     |
+-----+

```

计数是多播地址和提供的保持时间对的数目，它的值从 1 到 255。地址的长度依赖于当前地址家族。提供的保持时间对的数目受 512 字节的报文长度限制。

缺省：无。

描述：NMR 命令是实验性的，并没有在一个具体的实现中被测试。每个多播地址和保持时间对一起被叫做非成员报告。非成员报告告诉接收路由器发送路由器没有给定组中的后继组成员。基于这条信息，接收路由器能停止为了列表中特定的多播地址而向发送路由器转发数据报。时间对表示 NMR 有效的时间，以秒为单位。

如果计数等于 0，则出错。

在一个有 NMR 命令的报文中，仅有的其它命令是 AFI, flags0, 和 NULL 命令。与 flags0 相关的标志没有被定义，但这种情况可能会在将来改变。

3.10 非成员报告取消 (NMR Cancel) 命令

```
格式:  0 1 2 3 4 5 6 7   0 1 2 3 4 5 6 7
      +-----+-----+
      |          10          | |计数 (count) |
      +-----+-----+
```

计数 (count) 附加参数数组, AFI=IP

```
      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
      +-----+-----+-----+-----+
      | 多播地址 1                                     |
      +-----+-----+-----+-----+

      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
      +-----+-----+-----+-----+
      | 多播地址 2                                     |
      +-----+-----+-----+-----+
```

计数是提供的多播地址的数目，它的值从 1 到 255。地址长度依赖于当前地址家族。提供的地址数目受 512 字节的报文长度的限制。

缺省：无。

描述：NMR Cancel 命令是实验性的，并没有在具体的实现中被测试。对于每个列出的多播地址，任意先前相应的非成员报告被取消了。当对于给定的多播地址，没有相应的非成员报告，对那个多播地址的 Cancel 命令应该被忽略。

当计数 (count) 等于 0 时，出错。

在一个有 NMR Cancel 命令的报文中，仅有的其它命令是 AFI, flags0, 和 NULL 命令。与 flags0 相关的标志没有被定义，但这种情况可能会在将来改变。

3.12 例子 (字节在 {} 中)，不包括报文头部：

3.12.1 提供一条到 IP 地址为 128. 2. 251. 231 的单独的路由。路径的字段为：

度量 (metric) 为 2，无穷大 (infinity) 为 16，子网掩码为 255. 255. 255. 0：

Subtype 为 1, AFI 2, Metric 2, Infinity 16, Subnet Mask 255. 255. 255. 0
 {2} {2} {4} {2} {6} {16} {3} {1} {255} {255} {255} {0}
 DA Count=1 [128. 2. 251. 231]
 {7} {1} {128} {2} {251} {231}

3. 12. 2 提供一条到 IP 地址为 128. 2. 251. 231 和 128. 2. 236. 2 的路由

它的度量 (metric) 为 2, 无穷大 (infinity) 为 16, 子网掩码为 255.255.255.0:
 Subtype 1, AFI 2, Metric 2, Infinity 16, Subnet Mask 255.255.255.0
 {2} {2} {4} {2} {6} {16} {3} {1} 255} {255} {255} {0}
 DA Count=2 [128.2.251.231] [128.2.236.2]
 {7} {1} {128} {2} {251} {231} {128} {2} {236} {2}

3. 12. 3 请求到 IP 目的地址的所有路由。

Subtype 2, AFI 2, RDA Count = 0
 {2} {2} {8} {0}

3. 12. 4 组 224. 2. 3. 1 和 224. 5. 4. 6 (它们的保持时间为 20 秒) 和组 224. 7. 8. 5 (它的保持时间为 40 秒) 的非成员报告。

Subtype 3,
 AFI 2, NMR Count = 3 [224. 2. 3. 1, 20]
 {2} {2} {10} {3} {224} {2} {3} {1} {0} {0} {0} {20}
 [224. 5. 4. 6, 20] [224. 7. 8. 5, 40]
 {224} {5} {4} {6} {0} {0} {0} {20} {224} {7} {8} {5} {0} {0} {0} {40}

3. 13 命令总结

值	名字	在同一报文中允许的其它命令
-----	----	-----
0	Null	Null, AFI, Subnetmask, Metric, Flags0, Infinity, DA, RDA, NMR, NMR-cancel
2	AFI	Null, AFI, Subnetmask, Metric, Flags0, Infinity, DA, RDA, NMR, NMR-cancel
3	Subnetmask	Null, AFI, Subnetmask, Metric, Flags0, Infinity, DA, RDA
4	Metric	Null, AFI, Subnetmask, Metric, Flags0,

		Infinity, DA
5	Flags0	Null, AFI, Subnetmask, Metric, Flags0, Infinity, DA
6	Infinity	Null, AFI, Subnetmask, Metric, Flags0, Infinity, DA
7	DA	Null, AFI, Subnetmask, Metric, Flags0, Infinity, DA
8	RDA	Null, AFI, Subnetmask, Flags0, RDA
9	NMR	Null, AFI, Flags0, NMR
10	NMR-cancel	Null, AFI, Flags0, NMR-cancel

4 隧道

隧道是在被不支持多播路由的网关隔开的路由器之间发送数据报的一种方法。它充当两个路由器之间的虚拟网络。例如，有一台运行在斯坦福大学的路由器，和一台运行在 BBN 上的路由器，这两台路由器可以被一个允许多播数据报穿越因特网的隧道连接。我们认为隧道是过渡性的手段。

隧道是用弱封装的常规多播数据报来实现的。弱封装使用一个特殊的两元素 IP 松散源路由[5]。（这种封装形式比“强”封装（预先考虑一个完全新的 IP 头部）要好，因为它不要求隧道终端知道彼此的最大重装缓冲器大小。它也有始发者的生存时间值和任何出现的其它的 IP 选项的正当的特性所带来的好处。）

隧道有一个本地终端和远程终端，度量和与它相关的阈值。在隧道每一端的路由器仅需在本地和远程终端上达成一致就行。要了解有关隧道是如何配置的信息，参看第八部分。因为不知道一个隧道终端之间的中间网关的数目，所以需要进一步研究来确定合适的度量和阈值。

要在隧道上发送数据报，会出现如下事件：

- 一个空 IP 选项被插进数据报中。这为松散源路由 IP 选项提供了优先选取的对齐方式。
- 一个两元素松散源路由 IP 选项被插进数据报中。
- 设置源路由指针指向在源路由中的第二个元素。
- 在源路由中的第一个元素被起始主机（起始 IP 源地址）的地址所替换。
- 在源路由中的第二个元素被起始主机所提供的多播目的地址（起始的 IP 目的地址）所替换。**
- IP 源地址被路由器的合适的外出物理接口（本地隧道终端）的地址所替换。
- IP 目的地址被远程路由器（远程隧道的终端）的地址所代替。
- 使用非多播路由算法传送数据报到远程路由器。

中间的非多播网关将为隧道化的数据报选择到远程隧道终端的路由。因为数据报的 IP 源地址已经被本地隧道终端地址所替换，ICMP 出错信息将到达起始多播路由器。这种特性是所需的，因为发送多播数据报（多播路由器决定将该数据报放进隧道中）的主机并不知道使用了隧道。如果当封装这个数据报时，数据报的 IP 源地址没有改变，任何 ICMP 错误被送

到起始主机。

当远程隧道终端收到隧道化的数据报时，发生下列事件：

IP 源地址被松散源路由中的第一个元素所替换。

IP 目的地址被松散源路由的第二个元素所替换。

空选项和松散源路由选项被从数据报中移出。这种处理是必须的，因为主机并不知道它接收的数据报是从隧道发送来的。

因为没有特定的网络与一个隧道相联系，所以不用为一个隧道跟踪本地组成员。隧道的唯一的邻居是一个远程终端。路由信息通过隧道交换，但并不为一个隧道生成一个路由。路由信息应该以一个单播数据报发送，直接到达远程隧道终端；它们不能用 IP 松散源路由。

为隧道使用源路由和记录选项的原因是：

我们考虑定义我们自己的 IP 选项来处理隧道，但我们担心中间网关不能透明的传递它们不知道的 IP 选项。使用新选项的数据报不会穿越因特网。如果我们能生成一个新的 IP 选项，这样比较好，但这目前还不能做到。记住这是一个过渡设计，允许我们在当前环境中进行实验。

包含 LSRR 选项的隧道化分组有以下特征：

字段	值
源地址	= 源网关地址
目的地址	= 目的网关地址
LSRR 指针	= 指向 LSRR 地址 2
LSRR 地址 1	= 源主机
LSRR 地址 2	= 多播目的地址

由于为隧道使用 LSRR 选项而引起的两个问题是“中间网关能忽略这个选项？”和“目的网关能恰当的检测出 LSRR 被用于一个隧道吗？”

当中间网关收到一个数据报，它检查目的地址。对于一个隧道化的数据报，目的地址并不等于接收网关的地址。因此，LSRR 选项不会被检查出来，中间网关将把该数据报转发到目的地址的下一跳。

当目的网关接收到一个数据报时，它注意到该数据报的目的地址与它自己的一个地址相匹配。因为源路由还没有用完，它将查看下一个 LSRR 选项地址。这个地址是多播地址。因为主机被禁止把多播地址放到源路由上，网关能推断出 LSRR 用于隧道。这里的不足之处是 LSRR 中的多播地址可能有一些其它意义。不过当前还没有其它的意义被定义。

如果隧道化的数据报被错误的定址到不支持多播的目的网关上，则目的网关将试图找到一条到多播地址的路由。这将失败，而且一个 ICMP 目的不可达错误报文被送到隧道化的数据报源端。因为隧道化数据报的源地址已被调整为源多播网关的地址，ICMP 错误将不会被送到起始主机，这台主机并不知道隧道的存在。

5 路由算法

这一部分对距离矢量路由算法做一个简要的介绍。要了解更多的信息请参阅[1]。

虽然 DVMRP 能表示到单个主机的路由，这个转发和路由算法仅支持网络和子网路由。

在以下的讨论中，术语“虚接口”用来代表一个物理接口或一个隧道本地终端。物理接口是一个网络接口，例如是以太网卡。到目的地的路由将通过一个虚接口。术语“虚网络”用来代表一个物理网络或隧道，它仅能在参考物理网络上路由。

TRPB 算法通过计算最短（反向）路径树，将多播数据报从源端（物理）网络转发到这

个数据报所有可能的接收者。每个多播路由器必须确定它在树中相对于特定源端的位置，而且确定它的那些虚接口在最短路径树上。数据报就从这些虚接口中转发出来。排除不在最短路径树上的虚接口的过程叫做“修剪”(pruning)。

考虑一个虚拟网络，使用 Deering 的术语，如果一台路由器的责任是通过它的连接虚拟接口向一个虚拟网络转发数据报，则这台路由器叫做这个虚拟网络的“父亲”。虚拟网络也能被认为是这个路由器的“孩子”虚拟网络。使用孩子信息，路由器能进行反向路径广播。

不必要的的数据报可能仍会被送到一些网络上，而这些网络没有这些数据报的任何接收者。

有两种接收者：属于一个特定多播组的主机和多播路由器。如果在一个虚拟网络上没有多播路由器，则认为虚拟网络沿树向上到达一个给定的源端，这个虚拟网络是一个“叶子”网络。如果一个网络是给定源端的一个叶子，而且在这个网络上没有特定组的成员，则没有接收者接收从源端到这个网络上的组的数据报。那个网络的父亲路由器能放弃发送这个网络上的数据报，或“截断”最短路径树。跟踪和使用这个信息的算法是截断反向路径广播(TRPB)算法。

确定那些虚拟网络是否是叶子并不简单。如果任何临近路由器认为一个给定的虚拟网络在到一个给定目的地的路径上，那么这个虚拟网络不是叶子。否则，它是叶子。这是一个选举的功能。如果一个带有由分裂水平处理毒害的度量的路由被某一路由器发送，则那个路由器使用那个虚拟网络作为那个路由的向上树路径(即，那个路由器投票认为这个虚拟网络相对于这条路由的目的地来说不是叶子)。因为在一个虚拟网络上的路由器是动态的，而且所有路由更新信息并不被路由器保存，所以需要有一个试探法来决定一个网络是一个叶子。当时间长度为 LEAF_TIMEOUT 秒的保持定时器正在运行时，DVMRP 采样在一个虚拟接口上的路由更新信息。每个虚拟接口有一个保持时间定时器。如果当保持定时器还在运行时，或在任意其它时间收到一条带有一个被分裂水平处理毒害的度量的路由，那么这条路由的合适的虚拟接口“变坏”了，——它不是叶子。对每条路由来说，当保持定时器超时，任何没有变坏的虚拟接口被认为是叶子。

对于一个更好的转发算法——反向路径多播算法的描述，参见[3]。

一个路由实体应该有以下要素：

- 目的地址(多播数据报源端) *
- 目的地址的子网掩码 *
- 到达目的地址的下一跳路由器
- 到达下一跳路由器的虚拟接口 *
- 孩子虚拟接口列表 *
- 叶子虚拟接口列表 *
- 每个虚拟接口的主要路由器地址
- 每个虚拟接口的次要路由器地址
- 表示实体状态的标志集合
- 度量
- 无穷大

标有‘*’的行表示直接由转发算法使用的字段。

孩子和叶子接口列表能用位图实现。

5.1 发送路由报文

使用 DVMRP 路由报文能达到三个基本的目的：周期性的提供所有路由信息，为最近

改变的路由免费提供路由信息，为响应一个请求提供一些或所有路由信息。

送到物理接口的路由报文的 IP TTL 字段应为 1。

何时发送路由报文的规则：

—每过 FULL_UPDATE_RATE 秒，路由器应该发送带有所有路由信息的 DVMRP 报文给它的所有虚拟接口。为了在路由器发送更新信息时阻止它们同步，应该使用一个实时定时器。

—在路由改变时，路由更新信息应该为这个路由发送。为了避免网络被触发更新所淹没，触发更新之间必须有一些延时；建议使用 TRIGGERED_UPDATE_RATE 秒作为间隔时间。当 DVMRP 路由器重新启动时，对所有路由的请求应该被在所有虚拟接口上发送。

—如果可能，当 DVMRP 路由器将中止执行时，它应该在所有虚拟接口上，为所有路由发送带有等于无穷大的度量的 DVMRP 报文。

当报文发送到经由支持多播的网络连接的路由器上时，它应该被多播到地址 224.0.0.4。因此，路由器必须侦听每个支持多播的物理接口上的多播地址 224.0.0.4。如果不支持多播，则使用广播。就像已经提到的那样，到隧道的路由更新信息应以单播数据报的形式发送到远程隧道终端。

当发送路由报文时，除了响应特定的路由请求外（经由非零计数的 RDA 命令），必须进行毒害的分裂水平处理。这意味着给定一个使用网络 X 的路由，送到网络 X 上的路由信息必须包括度量为无穷大的路由，还应该包括设置在 FLAGS0 命令上的适当的标志。

毒害的分裂水平是减少路由循环的可能性的一种方式。另一中 RIP 中没有的方法是选择路由中的较好的无穷大。对于在一个小规模，连接良好的网络上传播的路由，小于 16 位的无穷大可能更好些。无穷大的值越小，计数到无穷大的事件发生的时间越短。在穿越一个大规模的互联网时，16 位的无穷大可能太小。以计数到无穷大事件发生的时间变长为代价，无穷大应该被增加。

在因特网上多播的一个概念是使用“阈值”来限制多播数据报离开一个网络。在子网或自治系统的边缘的多播路由器可能要求一个数据报具有大的 TTL 值，以便离开一个网络。这种机制使得大部分多播数据报处于一个网络中，减少了外部的通信量。如果一个应用程序想把多播超出它的本地网络的范围，那么它的数据报的 TTL 字段的值至少是阈值和到网络边缘的距离的和。必须有一个配置选项允许为物理接口和隧道指定阈值。

当一个路由器启动时，它必须在它的每个虚拟接口上为所有的路由发送一个请求。这个请求是一个带有 RDA 命令的报文，RDA 命令的计数等于 0。

5.2 接收路由报文

路由器必须知道路由报文到达的虚拟接口。因为路由报文的地址可能是所有多播路由器的 IP 地址，而且因为有隧道，接收接口不能仅仅通过检查报文的 IP 目的地址而被识别。

对于在路由报文中表示的每条路由，以下必须出现：

IF 为一个路由设置了度量：

THEN 增加有报文到达的虚拟接口的度量。

在路由表中查找路由的目的地址。

IF 路由没有出现在表中：

THEN 试图在路由表中发现到相同网络的一条路由。

IF 那条路由出现在表中：

THEN IF 这条路由和被发现的路由来自相同的路由器：

THEN CONTINUE 下一条路由。

```
    IF 路由没有一个无穷大的度量：
    THEN 在路由表中增加这条路由。
    CONTINUE 下一条路由。
IF 这条路由和被发现的路由来自同一个路由器：
THEN 清除路由定时器。
    IF 收到一个路由，它与被发现的路由的度量不同：
    THEN 使用新的路由和无穷大来改变被发现的路由。
        IF 度量与无穷大相等：
        THEN 置路由定时器的值为 EXPIRATION_TIMEOUT.
        CONTINUE 下一条路由。
IF 接收到的无穷大与被发现的无穷大不同：
THEN 将发现路由器无穷大改为接收到的无穷大。
    将被发现的路由的度量改为接收到的无穷大和发现路由度量的最小值。
ELSE IF 收到一个度量，（它小于被发现路由的度量或（路由定时器的当前值至少接近
    EXPIRATION_TIMEOUT 的一半，而且被发现路由的度量等于接收到的度量，
    这个度量小于接收到的无穷大））：
    THEN 使用接收到的路由改变路由表。清除路由定时器。
CONTINUE 下一条路由。
```

5.3 邻居

必须有一个列表保存在每个附属的网络上的临近多播路由器。信息可从接收到的 DVMRP 路由报文中获得。如果在 NEIGHBOR_TIMEOUT 秒中没有收到一个邻居的任何消息，则认为它已经关机了。

5.4 本地组成员

像[2]所要求的那样，多播路由器必须跟踪附属于它的有多播能力的网络上的组成员。每过 QUERY_RATE 秒，在每个网络上，应由一个指定的路由器发送一个 IGMP 成员请求给所有多播组地址（224.0.0.1）。IGMP 成员请求将使主机在一个短延迟内用 IGMP 成员报告作为响应。主机将为一个组发送一个报告声明这个组的多播地址。

成员请求报文的 IP TTL 字段为 1。

在一个网络上的路由器选举或“指定”一个单一的路由器发出请求。被指定的路由器是网络中 IP 地址最小的路由器。当启动时，路由器在获知（假设通过路由报文）还有一个更低的地址的路由器之前，它认为自己就是被指定的路由器。为了在启动时获知一个网络上出现的组成员，路由器应该多播许多成员请求，每次请求之间有一个短的延时。我们建议发送三个请求，每次的间隔是四秒。

多播路由器必须接收所有送到所有多播地址的数据报。当从一个接口上收到有关一个组的 IGMP 成员报告，它必须记录这个接口上这个组的存在性和时间，而且如果这个组已经被记录，则更新时间。被记录的组成员必须设置了超时时间。如果过了 MEMBERSHIP_TIMEOUT 秒后，没有收到一个被记录的组的组成员报告，则这个被记录的组将被删除。

6 转发算法

这一部分讲述多播转发算法和必须为这个算法保存的状态。

转发算法用于确定到达一个物理接口或隧道的多播数据报应该如果被处理。如果多播数据报是被淹没的，则在一个虚拟接口上接收到的数据报应该从所有其它的虚拟接口上转发出去。因为在互联网上的冗余路径，所有数据报应该被复制。路由算法提供的孩子和叶子信息用于修剪到所有可能的目的地的树的分支。

在路由实体中，每个虚拟接口都有一个占支配性的路由器地址。这个地址是在那个虚拟接口上，具有到目的地较低度量的路由（它的度量不等于无穷大）的路由器的地址。这个占支配性的路由器地址不是为下一跳的虚拟接口准备的。

在路由器实体中，每个虚拟接口中还有一个从属路由器地址。这个地址是认为自己是这个虚拟网络的父亲的路由器的地址。因此，从属路由器地址不是为到一个叶子网络的虚拟接口准备的。

管理在路由实体中孩子和叶子的算法如下：

当路由器启动时：

为每个虚拟接口生成一个路由实体，它带有：

- 在它的孩子列表中所有其它的虚拟接口，
- 一个空的叶子列表，
- 没有占支配性的路由器地址，
- 没有从属路由器地址。

为每个虚拟接口开始一个保持定时器，值为 `LEAF_TIMEOUT`。

当接收到一个新的路由：

生成这个路由实体，它带有：

- 在它的孩子列表中，包含除了接收到新路由的虚拟接口以外的所有虚拟接口，
- 空叶子列表，
- 没有占支配性的路由器地址，
- 没有从属路由器地址。

为除了接收到新路由的那个虚拟接口以外的所有虚拟接口开始保持定时器，值为 `LEAF_TIMEOUT`。

当在一个虚拟接口 `V` 上接收到一个邻居 `N` 送来的路由，它有一个比在路由表中那一个低的度量（或相同的度量，如果 `N` 的地址小于 `V` 的地址），对于这条路由：

If `N` 是 `V` 的占支配性的路由器，不让 `N` 再当占支配性的路由器，并且在孩子列表中加入 `V`。

当在虚拟接口 `V` 上接收到邻居 `N` 送来的路由，它的度量等于无穷大（分裂水平标志应该被设置），对这条路由：

If `V` 在叶子列表中，从叶子列表删除 `V`。

If `V` 没有占支配性的路由器，把 `N` 记录为占支配性的路由器。

当在虚拟接口 `V` 上接收到一个邻居 `N` 送来的路由，它的度量不同于无穷大（没有分裂水平标志），对这条路由：

If `N` 是 `V` 的占支配性的路由器，不让 `N` 再当占支配性的路由器，并且为 `V` 开始保持定时器。

当一个虚拟接口 `V` 的定时器超时，对每条路由：

If `V` 没有占支配性的路由器，在叶子列表中加入 `V`。

当虚拟接口 V 的邻居 N 失败，对每一条路由：

IF N 是 V 的占支配性的路由器，不让 N 再当占支配性的路由器，并为 V 开始保持定时器。

转发算法是：

IF IP 的 TTL 字段小于 2：

THEN CONTINUE 下一个数据报。

寻找到 IP 数据报的源端的路由。

IF 没有路由存在：

THEN CONTINUE 下一个数据报。

IF 没有为这个路由在下一跳虚拟接口上收到数据报

THEN CONTINUE 下一个数据报。

IF 数据报被隧道化：

THEN 用在 IP 松散源路由中的第一个地址代替数据报的源地址。

用在 IP 松散源路由中的第二个地址代替数据报的目的地址。

删除数据报的松散源路由和空 (null) 选项，并相应调整 IP 头部的长度字段。

IF 数据报的目的地址是组 224.0.0.0 或组 224.0.0.1：

THEN CONTINUE 下一个数据报。

FOR 每一个虚拟接口 V

DO IF V 在数据报源端的孩子列表中：

THEN IF V 不在源端的叶子列表中

OR 在 V 上有目的组的成员

THEN IF IP 的 TTL 字段比 V 的阈值要大：

THEN 将 TP 的 TTL 字段减 1

将数据报从 V 中转发

7 时间值

这一部分包括不同速率和超时，以及它们的意思，它们的值的列表。所有值都以秒为单位。

路由环境的动态性影响以下速率。较低的速率将允许环境发生改变时的快速适应，代价是浪费了网络带宽。

FULL_UPDATE_RATE=60

—带有完整的路由表的路由报文被发送的时间间隔。

TRIGGERED_UPDATE_RATE=5

—触发路由报文可能被发送的时间间隔。

提高以下的速率和超时值可能增加分组被转发到一个虚拟接口的时间值。

QUERY_RATE=120

—发出本地组成员请求的时间间隔。

MEMBERSHIP_TIMEOUT=2*QUERY_RATE+20

—本地组成员关系在没有证实的情况下的最长有效时间。

LEAF_TIMEOUT=2*QUERY_RATE+20

—为每个虚拟接口设置的保持定时器的超时值。

增加下面的超时值会增加路由算法的稳定性，代价是路由环境改变时较慢的反应。

NEIGHBOR_TIMEOUT=4*FULL_UPDATE_RATE

—在没有被证实的情况下，一个邻居被承认的时间。这对于超时路由，和设置孩子和叶子标志时很重要。

`EXPIRATION_TIMEOUT=2*FULL_UPDATE_RATE`

—在没有被证实的情况下，一条路由被认为是有效的的时间。当定时器超时，分组将不会在这条路由上转发，路由更新将认为这条路由有一个无穷大的度量。

`GARBAGE_TIMEOUT=4*FULL_UPDATE_RATE`

—在没有被证实的情况下，一条路由存在的时间。当定时器超时，路由更新将不再具有这条路由的任何信息。这条路由将被删除。

8 配置选项

一条路由应该可以被配置下列信息：

—隧道描述：本地终端，远程终端，度量，和阈值。如果没有提供阈值，度量应该被用于缺省的阈值。

—对于一个物理接口：度量，无穷大，阈值和子网掩码。如果没有提供阈值，度量应该被用于缺省的阈值。

9 结论

本备忘录展示了 DVMRP（一种可扩展的远距离矢量路由协议）和 TRPB 路由算法。在这篇文档里提到的思想的一个具体实现已经被完成，正在测试阶段。

与 RIP 相比，在 DVMRP 中增加的特征使得它更加灵活，代价是更加复杂的处理。做为一个距离矢量算法，DVMRP 仍然用不足之处。因为连接状态类算法维持 DVMRP 所要维持的状态信息中许多信息，而这些信息超出了 RIP 的需要，所以一个多播状态类的路由协议应该被开发。

TRPB 算法能促使不必要的数据报被发送。反向路径多播算法（RPM）可能是一个更好的算法。设计 NMR 和 NMR-cancel DVMRP 报文的目的是支持 RPM。对于这个主题需要更深入的研究。

10 致谢

我们将感谢 Robb Foster, Alan Dahlbom, Ross Callon 和 IETF 主机工作组提供了他们的思想。

11 参考书目

- [1] Hedrick, C., "Routing Information Protocol", [RFC 1058](#), Rutgers University, June 1988.
- [2] Deering, S., "Host Extensions for IP Multicasting", [RFC 1054](#), Stanford University, May 1988.
- [3] Deering, S., "Multicast Routing in Internetworks and Extended LANs", SIGCOMM Summer 1988 Proceedings, August 1988.
- [4] Callon, R., "A Comparison of 'Link State' and 'Distance Vector' Routing Algorithms", DEC,

November 1987.

[5] Postel, J., "Internet Protocol", [RFC 791](#), USC/Information Sciences Institute, September 1981.

[6] Mills, D., "Toward an Internet Standard Scheme for Subnetting", [RFC 940](#), University of Delaware, April 1985.