# 利用多播树实现 Anycast 服务

王晓喃[1] 钱焕延[2]
(南京理工大学 南京 210094)[1] (常熟理工学院 常熟 215500)[2]

**摘 要** IPv6 以两种方式提供 Anycast 服务:一种是将 Anycast 组成员限制在共享一个地址前缀的特殊拓扑区内;另一个是将 Anycast 地址表示的共享某个特性的结点组分散在互联网的各个地方,这种方式使得路由表会随全球 Anycast 组数呈比例增长。无论是哪种方式,它们都存在诸如 Anycast 可扩展局限性等问题。本文提出了一种建立在 Anycast 树之上的通信模型,此模型实现了 Anycast 组成员的动态加入与离开,从真正意义上解决了 Anycast 现存的扩展性问题,同时此模型实现了 Anycast 树自身信息与请求的分布式维护与处理,从而实现了均衡负载功能。本文同时深入分析和讨论了该模型的可行性及其有效性,并论证它可以支持大规模的 Anycast 组的建设。

**关键词** IPv6,Anycast,树,节点

## Implementation of Anycast Service with Multicast Tree

WANG Xiao-Nan[1] QIAN Huan-Yan[2]
(Nanjing University of Science & Technology, Nanjing 210094)[1] (Changshu Institute of Technology, Changshu 215500)[2]

**Abstract** The existing designs for providing Anycast services are either to confine each Anycast group to a preconfigured topological region or to globally distribute routes to individual Anycast groups which causes the routing tables to grow proportionally to the number of all global Anycast groups in the entire Internet, both of which restrict and hinder the application and development of Anycast services. A new kind of Anycast communication model is proposed on the basis of Anycast tree in this paper. Since this model achieves dynamic Anycast group and allows Anycast members to freely leave and join Anycast group it radically solves the existing scalability problem. In addition, this model accomplishes the distributed maintenance and transaction of Anycast service request and the information on Anycast tree so it fulfills the load balance. This paper deeply analyzes and discusses the feasibility and validity of this communication model, and argues that is supports the large-scale Anycast group.

**Keywords** IPv6,Anycast,Tree,Node

## 1 前言

Anycast 是 IPv6 所提供的一种特殊网络服务,它允许服务申请者访问共享同一 Anycast 地址所标识的一组组成员中最近的一个(这里的最近是按路由协议的距离量度来计算)。如图 1 所示,图中 Sender1 和 Sender2 都向同一个 Anycast 地址发出了服务请求数据包,但是该数据包被网络转发到距离发送者最近的一个组成员,这里假设 member1 距离 Sender1 最近,menber2 距离 Sender2 最近。

Multicast 是一种在 IPv4 中就已经存在的网络服务,它允许服务申请者访问共享同一 Multicast 地址所标识的一组组成员;它与 Anycast 的区别在于,Anycast 只访问一个 Anycast 组中距离源主机最近的一个组成员;而 Multicast 是访问一个 Multicast 组中的所有组成员。图 1 中,Sender1 发送一个 Multicast 请求数据包,该数据包被网络同时转发到 Multicast 组的所有组成员 member1 和 member2。

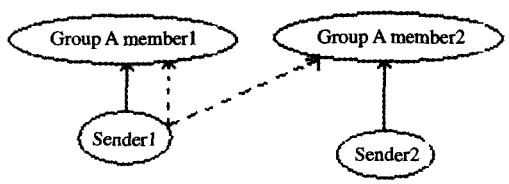


图 1 Anycast 与 Multicast

不难看出,Anycast 与 Multicast 的相似之处就是它们都用一个地址标识一组组成员,而不同之处就是 Anycast 只是将数据包发送给一个组成员,而 Multicast 将数据包发送给所有的组成员。在 IPv6 中,Multicast 采用独立的地址空间,而 Anycast 地址从 Unicast 地址空间中分配,即 Unicast 和 Anycast 地址从结构上没有任何区别。根据以上 Anycast,Unicast 以及 Multicast 之间的关系,我们设计了一种新的 Anycast 通信模型方案,此方案利用 Unicast 的路由方式实现 Anycast 数据包的推进传输,而利用 Multicast 的路由方式实现 Anycast 组成员的管理与维护。下面我们对此模型进行深入的讨论与分析。

## 2 Anycast 通信模型

### 2.1 Anycast 地址问题

IPv6 中的 Anycast 地址模型与 RFC1546 最初建议的完全不同,前者提出在 Unicast 地址空间中分配 Anycast 地址,这样 Unicast 和 Anycast 地址从结构上没有任何区别;而后者则推荐使用独立的地址模型。本模型采用前者的观点,从 Unicast 地址空间分配 Anycast 地址。

IPv6 的地址格式与 IPv4 不同,一个 IPv6 的 IP 地址由 8 个地址节组成,每节包含 16 个地址位,除了 128 位的地址空间,IPv6 还为点对点通信设计了一种具有分级结构的地址,

其分级结构划分如下所示：

| 3 | 13 | 8 | 24 | 16 | 64 |
|---|---|---|---|---|---|
| FP | TLAID | RES | NLA ID | SLA ID | Interface ID |

其中，FP是可聚合全局地址的格式前缀(例如，001代表单播地址)；TLA ID为顶级聚合标识符；RES为将来使用而保留；NLA ID是次级聚合标识符；SLA ID是站点级聚合标识符；Interface ID为接口标识符。

从以上分析看出，IPv6的地址是分层的。根据IPv6地址的特点，本模型采用如下的Anycast地址格式：

| 3 | 13 | 8 | 24 | 16 | 64 |
|---|---|---|---|---|---|
| Anycast Prefix | Main Domain | | | | Group ID |

一个Anycast地址分为三部分：第一部分是(即前三位)Anycast的地址前缀，其取值范围与Unicast的取值相同，即001；而其随后的TLA ID、RES、NLA ID和SLA ID作为第二部分，即Anycast主域；最后一部分是Anycast组ID。

### 2.2 Anycast树

本模型是建立在Anycast树基础之上的。本模型定义一个Anycast树，可以包括三类节点：第一类节点是根节点，此节点所在的网络区域的Unicast地址空间必须与其所拥有的Anycast地址空间相同，即目的地址为Anycast地址的数据包可以按照正常的Unicast路由方式被路由到此根节点。在本模型中，根节点所在的网络区域称作主域，一种Anycast服务对应唯一的一个Anycast树，一个Anycast树对应唯一的一个根节点。第二类节点是中间节点，也称作树节点，它们不能提供Anycast服务而只用于支撑Anycast树框架，这类节点一般都是路由器。第三类节点是叶子节点，也称作组节点，这类节点是可以提供Anycast服务的节点，一般都是Anycast服务器。在本模型中，根节点与叶子节点都可以提供Anycast服务，并且根节点的Anycast地址与Unicast地址是相同的，而其他组节点以及树节点都具有自己本身的Unicast地址，它与Anycast地址是不同的。

### 2.3 Anycast树的建立

下面讨论Anycast树的建立，即如何把一个新的组成员加入到Anycast树以及一个Anycast组成员如何离开所在的Anycast树。

当一个主机请求加入Anycast组的时候，它首先将自己标记为该组的组节点，然后发送Join消息，其目的地址为请求加入的Anycast组地址，同时记录下本节点的父节点的Unicast地址(即Join消息的下一跳的Unicast地址)。这样，网络系统会把该消息朝着Anycast树根节点的方向路由推进。在路由过程中，Join消息所经过的每个路由器在接收到它之后，都会检查自身是否为该消息中的Anycast组地址所代表的Anycast树的树节点。如果不是，那么此路由器首先将自己标记为Anycast树节点。同时，建立一个孩子节点记录表，将发送Join消息的源主机作为自己的第一个孩子节点加入到自己的孩子节点记录表中，并记录下Join消息中的相关参数，包括发送Join消息节点的Unicast地址、权值、到达本节点所经过的Hop数，以及所属的Anycast组地址等信息，同时记录下本节点的父节点的Unicast地址(即Join消息的下一跳的Unicast地址)，然后它用自己的Unicast地址作为原有Join消息中的源地址，目的地址不变，并修改相应的参数(例如跳数)，将其发送出去。如果是树节点，那么它将发送Join消息的源主机加入到自己的孩子节点记录表中，并记录下其相关参数(参数内容同上)，然后不再转发该Join消息。至此，该主机已经成功加入到所请求的Anycast组中。

一个主机请求加入一个Anycast组的过程如图2所示。

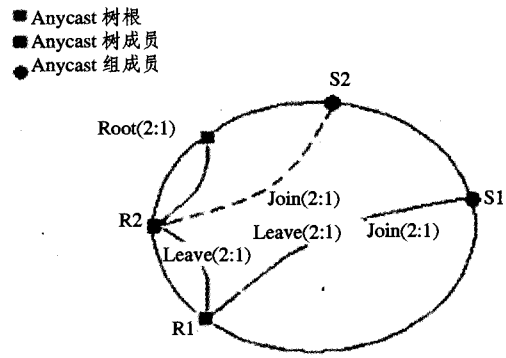


图2 Anycast树的建立过程

图2中，主机S1请求加入Anycast地址为2::1的Anycast组，它首先将自己标记为该Anycast组的组节点，然后发出Join消息要求加入此Anycast组，此Join消息的目的地址为Anycast地址，同时记录下本节点的父节点R1的Unicast地址(即Join消息的下一跳的Unicast地址)。这样，网络系统将该消息首先路由到路由器R1，R1接收到该消息之后，首先将自己标识为该Anycast树的树节点，然后创建孩子节点记录表，将S1加入到此表中并保存其S1的相关参数，包括S1的Unicast地址、权值、到达本节点所经过的Hop数以及所属的Anycast组地址等信息，然后用自己的Unicast地址取代Join消息中的源地址，最后将该Join消息转发到下一跳路由器R2，并且记录下本节点的父节点R2的Unicast地址。同样，R2接收到该消息之后，首先将自己标记为该Anycast树的树节点，然后创建孩子节点记录表并将R1加入自己的孩子列表，同时记录下R1的Unicast地址、权值、源节点S1到达本节点的Hop数，以及它所属的Anycast组的地址等相关信息，最后用自己本身的Unicast地址取代Join消息中的源地址，并且记录下本节点的父节点Root的Unicast地址，将其转发到下一跳，即根节点。根节点接收到该消息之后，由于它本身是此Anycast树的组节点，因此它直接将R2加入到自己的孩子列表中并记录下其相关参数，并且停止转发该消息。至此，S1成功地成为Anycast树的一个组成员。接下来，主机S2也申请成为Anycast地址为2::1的Anycast组的组节点，首先它将自己标记为该Anycast组的组节点，并发送Join消息请求加入该组。Join消息的目的地址为Anycast地址，同时记录下本节点的父节点R2的Unicast地址(即Join消息的下一跳的Unicast地址)。网络系统首先将该消息路由到路由器R2。R2接收到此消息之后，因为R2已经被标记为Anycast地址为2::1的Anycast组的树节点，所以它直接将S2加入到自己的孩子列表中并记录下S2的Unicast地址、权值、到达本节点所经过的Hop数，以及所属的Anycast组地址等信息，同时停止转发该请求。至此，S2也成功地成为Anycast树的一个组成员。

不难看出，上述的Anycast组节点的加入过程可以保证所有的Anycast节点(包括树节点和组节点)组成一个树状结构。

下面分析一个 Anycast 组节点如何离开所在的 Anycast 树。

如果一个组节点申请离开其所在的 Anycast 组，它首先删除自身组节点的信息与身份，然后发送一个 Leave 消息给它的父节点。父节点接收到这个 Leave 消息之后，会检查自身对应 Leave 消息中 Anycast 地址的 Anycast 树的孩子节点记录表并从中删除此组节点，然后判断此时的记录表是否为空。如果为空，那么它将删除自身的树节点信息，然后发送一个 Leave 消息给它的父节点。父节点接收到这个 Leave 消息之后，同样从自身的孩子列表中删除发送 Leave 消息的孩子节点，然后判断此时的孩子列表是否为空。如果为空，将继续重复上述过程直到根节点或者某个节点。该节点在删除发送 Leave 请求的孩子节点之后，其孩子记录表不为空。

如图 2 所示，如果 Anycast 组成员 S1 请求离开 Anycast 地址为 2::1 的 Anycast 树，它首先删除自身组节点身份与相关参数，然后发送 Leave 消息到它的父节点 R1。R1 接收到该 Leave 消息之后，首先根据消息中的 Anycast 地址删除相应 Anycast 组的孩子组节点 S1，然后检查此时的孩子记录表是否为空。因为此表为空，所以它删除自身树节点的身份与相关参数，然后继续发送 Leave 消息给其父节点 R2。父节点 R2 接收到 Leave 消息之后，同样根据消息中的 Anycast 地址删除相应 Anycast 组的孩子组节点 R1，并且检查 Anycast 组的孩子记录表是否为空。因为此时该表不为空（S2 是它的孩子节点），所以 R2 继续保留 Anycast 树成员的身份，并且停止转发该 Leave 消息。至此，S1 成功地离开 Anycast 地址为 2::1 的 Anycast 组。

### 2.4 路由分析

上面章节已经提到过，本模型将节点分为组节点和树节点，而只有组节点才提供 Anycast 服务。这样，当一个主机申请 Anycast 服务时，它首先发送一条 Anycast 地址转换为 Unicast 地址的请求，本模型会将该请求路由到最佳 Anycast 组节点上进行处理，此最佳组节点会将自身的 Unicast 地址作为应答消息的一部分返回给源主机，此后，源主机与 Anycast 组成员之间就可以按照正常的 Unicast 通信模式进行直接通信了。下面具体讨论本模型如何获取最佳 Anycast 组节点。

在本模型中，每种 Anycast 服务都被赋予一个 Anycast 地址，Anycast 地址转换请求消息通过这个 Anycast 地址可以被网络系统朝着 Anycast 树根节点的方向路由推进。在路由过程中，每经过一个路由器，它都会检查自己是否为此 Anycast 树的树节点。如果是，那么就查找当前以此节点为根节点的子树中最优的组节点，否则将该消息向下一跳推进。

本模型采用权值的方式来获取最优的组节点。本模型规定每个 Anycast 节点（包括组节点和树节点）都有一个树权值，其中，组节点（除根结点以外）的树权值为自身的权值，而树节点的树权值为孩子节点中权值最大的值，根节点的树权值为自身权值与其孩子节点中权值最大的值。本模型采用如下的公式来计算权值：

$$Val\ 组节点\ i = N_{组节点i}$$

$$Val\ 树节点\ i = \max_{1}^{k}(N_{孩子节点j})$$

$$Val\ 根节点 = \max_{1}^{k}(N_{根节点}, N_{孩子节点j})$$

其中，$Val_i$ 表示节点 $i$ 的树权值；$N_i$ 表示节点 $i$ 自身的权值。每个组节点的权值根据不同服务的性质与质量要求，可以采用不同的度量单位，比如组节点当前所处理的会话数，组节点到达某个树节点的跳数，等等。图 3 是在图 2 的基础上建立起来的带有树权值的 Anycast 树。

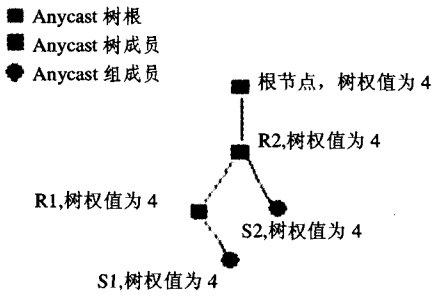


图 3 Anycast 节点的权值

如图 3 所示，S1 为一个组节点，所以此节点的树权值就是它本身的权值 4，S2 也是一个组节点，其组节点的树权值也是它本身的权值 2。而 R1 只有 S1 这一个孩子节点，所以 R1 的树权值为 S1 的树权值 4，同时 R1 记录下此树权值的来源指向 S1；R2 有两个孩子节点 R1 和 S2，那么它的树权值为其孩子节点树权值最大的值 4，其来源指向为 R1，而根节点的权值为 3，而它的孩子节点的权值为 4，所以它的树权值为孩子节点的权值 4，树权值来源指向为 R2。

这样，当一个 Anycast 地址转换请求消息到达 R2 时，因为 R2 本身是此 Anycast 地址所对应的 Anycast 树的一个树节点，所以根据其树权值的来源指向，将该请求消息转发给 R1。同理，R1 将此消息转发给组节点 S1。至此，获取了以树节点 R2 为根节点的子树中的最优组节点 S1。S1 接收到该请求之后，将自己的 Unicast 地址作为响应信息的一部分返回给源主机。这样，源主机就可以利用接收到的 Unicast 地址与最优 Anycast 组成员进行直接通信了。

在本通信模型中，我们定义只有根节点与叶子节点可以提供 Anycast 服务。在某些极端情况下，一个 Anycast 树可能只包括一个根节点，那么所有发送到这个 Anycast 地址的数据包都会按照正常的 Unicast 路由方式被路由到根节点来处理。因此，本模型保证了在任何情况下客户端都能获取 Anycast 服务。

在本模型中，我们采用当前的会话数为度量单位来确定每个节点的树权值，这样每个 Anycast 组节点的树权值会随着时间的变化而相应变化。为了确保 Anycast 节点的树权值的正确性和有效性，Anycast 树中的每个树节点都必须定期向其孩子节点发送查询消息，以便及时更新自身的树权值。在本方案中，Anycast 节点采用如下的参数和算法来确定更新树权值的频率：参数包括时间间隔 L、最大阈值 T、步长 R、当前的阈值 M、其初始化值为 T。Anycast 节点每隔 L 检测一次本节点的当前服务请求流量，如果当前的流量值与上次检测到的流量值之差的绝对值大于 M，那么就发送查询消息给其孩子节点。其孩子节点接收到这个查询消息之后，将当前本身的树权值返回给父节点。父节点选择出最大的树权值作为自身的树权值，并保存该树权值的来源指向。如果当前的流量值与上次检测到的流量值之差的绝对值小于 M，那么就将 M 的值减去步长 R。这样，在客户流量比较平稳的情况下，Anycast 节点至少能每隔 T/R 的时间单位发送一次查询消息。反之，它至多每隔 L 时间单位发送一次查询消息。

## 3 性能分析

本模型是建立在 Anycast 树基础之上的，它从根本意义上解决了 Anycast 的扩展局限性问题，从而真正地实现了高质量、响应速度快的 Anycast 服务。

在本模型中，当一个主机发送一个 Anycast 地址转换请求时，此请求所到的第一个 Anycast 树节点一定是整个 Anycast 树中距离源节点最近的树节点。然后以此树节点为子树根节点，再根据其子树成员的树权值(可以采用多种度量方式，本模型中采用当前的会话数)，查找到最佳组节点。不难看出，本模型通过这种多度量单位选择最优组成员的方式来为客户提供响应速度最快的高质量 Anycast 服务。此外，由于本模型采用树状结构，允许 Anycast 节点可以动态地加入或离开，并不受物理位置的限制，从而解决了 Anycast 扩展局限性问题。

在本模型中，Anycast 组节点的加入和离开都是分布式处理的，并不是集中在某个固定节点上，这就解决了由于瓶颈可能导致网络阻塞或者节点超负载而宕机的问题。此外，由于加入与离开消息的数据传输只需要跨越很小的物理网络并且此类消息的数据传输量也非常小，因此对网络性能基本没有影响。本模型中的 Anycast 树状结构的信息是采用分布式管理与维护的，即每个节点只负责管理和维护以其为根节点的子树所包含的叶子节点的信息，这就实现了 Anycast 树中节点信息的分布式维护与管理，从而实现了负载均衡的作用。此外，在本模型中不同客户发出的服务请求消息会被不同的最优 Anycast 组节点处理，这同样保证了 Anycast 服务请求可以均衡地分布在 Anycast 组成员之间，从而得到高效的处理。本模型已经在 IPv6 的模拟环境下成功实施，每个节点的树权值是以当前的会话数为度量单位的。

本模型的性能分析是通过在 IPv6 模拟环境下比较客户通过本模型获取 Anycast 服务的 TRT 值与正常情况下以跳数为度量单位获取 Anycast 服务的 TRT 值来实现的。因为从用户角度来看，所提供服务的 TRT 值越小，用户认为服务质量越好。我们在两种实现方式中，客户端获取同样的 Anycast 服务，彼此交互的信息数据量也相同，那么就得到如下的 TRT 性能分析图(图 4)。

$R = TRT_{Normal} / TRT$

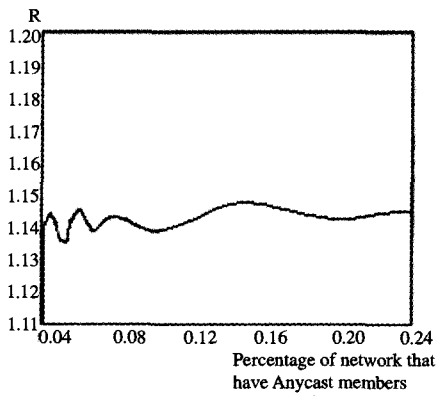


图 4

其中，$R$ 为在本模型中客户获取 Anycast 服务的 TRT 值与正常情况下客户获取 Anycast 服务的 TRT 值的比值，$TRT_{Normal}$ 为在正常情况下客户获取 Anycast 服务的 TRT 值，TRT 为在本模型中客户端获取 Anycast 服务的 TRT 值。从上图可以看出，$R$ 的比值趋于 1.145。这个试验结果表明，在本模型中客户获取 Anycast 服务的整体响应时间优于在正常情况下客户获取 Anycast 服务的响应时间。

**结束语** Anycast 是 IPv6 的一个新特性，它可以支持许多服务。本文在 IPv6 的模拟环境下，提出了实现 Anycast 服务的一种新的通信模型，用以解决目前 Anycast 服务所存在的一些问题。Anycast 作为一种新型的通信模式，具有广泛的前景，但是它还存在许多问题，有待进一步探讨和研究。

## 参 考 文 献


1 Hagino J I, Ettikan K. An analysis of Ipv6 anycast Internet Draft. Internet Engineering Task Force, 2001

2 Katabi D, Wroclawski J. A framework for scalable global IP-Anycast (GIA). In: Proc of SIGCOMM, New York: ACM Press, 2000. 3～15

3 Castro M, Druschel P, Kermarrec A M, et al. Scalable application-level anycast for highly dynamic groups. Prentice Hall, 2003

4 Afergan M, Wein J, LaMeyer A. Experience with Some Principles for Building an Internet-scale reliable System. In: Proceeding of Second Workshop on Real, Large Distributed System, Dec 2005

5 Ballani H, Francis P. Towards a Global IP Anycast Service. In: Proceeding of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. Aug 2005

6 Castro M, Druschel P, Kermarrec A, et al. Scalable Application-level Anycast for Highly Dynamic Groups. In: International Workshop on Networked Group Communication, Sept 2003

7 Dilley J, Maggs B, Parikh J, et al. Globally Distributed Content Delivery. IEEE Internet Computing, 2002, 6(5)

8 Doi S, Ata S, Kitamura H, et al. Design, Implementation and Evaluation of Routing Protocols for IPv6 Anycast Communication. In: IEEE 19th International Conference on Advanced Information Networking and Applications, Mar 2005

9 Kim D, Meyer D, Kilmer H, et al. Anycast Rendevous Point (RP) Mechanism Using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP). RFC 3446, Jan 2003

10 Doi S, Ata S, Kitamura H, et al. Protocol design for anycast communication in IPv6 network. In: Proceedings of 2003 IEEE Paci c Rim Conference on Communications, Computers and Signal Processing (PACRIM'03), (Victoria), Aug 2003. 470～473

11 Doi S, Ata S, Kitamura H, et al. IPv6 Anycast for Simple and Effective Communications. IEEE Communications Magazine, 2004, 42(5): 163～171