# IPv6 中利用多播树解决 Anycast 扩展局限性*)

王晓喃[1,2]　唐振民[2]

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

**摘　要**　提出了一种利用多播树实现 Anycast 服务的一种通信模型，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。深入分析和讨论了该模型的可行性及其有效性。在 IPv6 模拟环境下，实验数据表明通过本模型获取 Anycast 服务(比如文件下载服务)的 TRT 值要比现有的 Anycast 通信模型下获取同样服务的 TRT 值缩短很多，服务质量也有明显提高。

**关键词**　IPv6，Anycast，树，节点，多播

## Solving Anycast Scalability with Multicast Tree in IPv6

WANG Xiao-nan[1,2]　TANG Zhen-min[2]

(Changshu Institute of Technology, Changshu 215500, China)[1]　(Nanjing University of Science & Technology, Nanjing 210094, China)[2]

**Abstract**　A new kind of Anycast communication model was proposed on the basis of multicast tree technology. Since this model achieves dynamic Anycast group, it 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. Deeply analyzed and discussed the feasibility and validity of this communication model, and the experimental data in IPv6 simulation demonstrated that the TRT of one Anycast service(for example, file downloading) acquired through this communication model is shorter than the one through the current communication model.

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

## 1　前言

Anycast 是 IPv6 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单位来计算)。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，member2 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。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组成员。

## 2　相关工作

文献[4,5]提出了利用现有 Multicast 通信模型以及协议来实现 Anycast 服务，但是没有给出具体的实现方案以及实现过程，并且只是采用静态机制来选择 Anycast 最优成员，而没有根据网络的动态变化而提供相应的动态选择机制。在这种情况下，本文针对 IPv6 网络，提出了一种新的 Anycast 通信模型。本模型具有如下特点：1)针对 IPv6 网络，本文将 Unicast 技术与 Multicast 技术结合起来提出了一种实现 Anycast 服务的全新通信模型；2)本模型可以根据网络的拥挤情况动态地选择最优 Anycast 组成员；3)本模型解决了 Anycast 扩展局限性问题并且可以采用多种距离度量方式来动态地确定 Anycast 最优组成员。

下面我们对此通信模型进行详细的讨论和分析。

## 3　Anycast 通信模型

### 3.1　Anycast 地址问题

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 3 部分，第一部分是(即前 3 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 是第二部分，即 Anycast 主域，最后一部分是 Anycast 组 ID。

### 3.2　Anycast 树

本模型是建立在 Anycast 树基础之上的。本模型定义一个 Anycast 树包括 3 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

*)国防科工委应用基础资金资助项目(JI300D004)，部委预研课题(课题编号：51316080101)，南京理工大学研究生创新基金(2007060005)。

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，只用于支撑 Anycast 树框架，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的 Unicast 地址，它与 Anycast 地址是不同的。

### 3.3 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，首先将自己标记为该组的组节点，然后构建 Join 消息，此消息包括主机本身的 Unicast 地址、所要加入的 Anycast 组选择最优组成员所采用的距离度量参数（比如跳数、当前处理的会话数或者是主机的处理能力等等）以及申请加入的 Anycast 组地址等信息。目的地址为请求加入的 Anycast 组地址，然后将其发送出去，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 组地址所确定的 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，将申请加入 Anycast 组的源主机作为自己的第一个孩子节点加入到孩子节点记录表中，并记录下 Join 消息中的相关参数，即申请加入 Anycast 组的主机的 Unicast 地址、它到达此树节点的距离参数以及申请加入的 Anycast 组地址等信息，同时记录下本节点的子节点的 Unicast 地址（即 Join 消息的源 Unicast 地址）以及父节点的 Unicast 地址（即 Join 消息的下一跳的 Unicast 地址）。然后用自己的 Unicast 地址取代原有 Join 消息中的源地址，目的地址不变，并修改相应的距离度量参数（例如跳数），将其发送出去；如果是树节点，那么它将申请加入 Anycast 组的源主机加入到自己的孩子节点记录表中，并记录下 Join 消息中相关参数（参数内容同上）以及子节点的 Unicast 地址（即 Join 消息的源 Unicast 地址），然后用自己的 Unicast 地址取代原有 Join 消息中的源地址，目的地址不变，并修改相应的距离度量参数，将其发送出去。然后，每个接收到 Join 消息的路由器都会重复上述过程，直到此 Join 消息到达根节点为止。至此，该主机成功加入到所请求的 Anycast 组中，如图 1 所示。

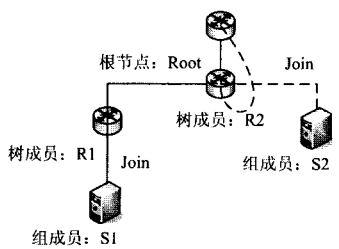


图 1　Anycast 树的建立过程

图 1 中，主机 S1 请求加入一个 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，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 S1 的相关参数，包括 S1 的 Unicast 地址、它到达 R1 的距离参数以及申请加入的 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子节点 S1 以及父节点 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 S1 加入自己的孩子列表，同时记录下 S1 的 Unicast 地址、S1 到达本节点的距离参数以及 S1 申请加入的 Anycast 组地址等相关信息，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子节点 R1（Join 消息的源地址）以及父节点 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，重复上述过程，同时停止转发该 Join 消息。至此，S1 成功地成为 Anycast 树的一个组成员。接下来，主机 S2 也申请成为同一个 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，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 已经被标记为此 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子节点 S2 以及父节点 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，重复上述过程，同时停止转发该 Join 消息。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，此 Leave 消息包括此组节点的 Unicast 地址以及所在的 Anycast 组地址。父节点接收到这个 Leave 消息之后，它会检查自身对应 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。无论此时的记录表是否为空，它都将继续发送一个 Leave 消息给它的父节点。父节点接收到这个 Leave 消息之后，继续重复上述过程，直到根节点为止。

如图 2 所示，如果 Anycast 组成员 S1 请求离开它所在的 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 S1，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，然后发送 Leave 消息给其父节点 Root，Root 节点重复上述过程，同时停止转发该 Leave 消息。至此，S1 成功地离开 Anycast 组。

### 3.4 权值计算

本模型采用权值的方式来获取最优 Anycast 组节点。我们知道，一个树节点的孩子节点记录表中记录着以这个树节点为根节点的子树的所有叶子节点（组节点）的相关信息，其中权值域就是记录每个叶子节点在此树节点的权值。在本模型中，我们假设一个树节点有 $N$ 个叶子节点，$D_i$ 表示其第 $i$ 个叶子节点到达此树节点的距离（此处的距离可以选择多种度量单位，例如跳数、综合处理能力或者是当前正在处理的会话数等等），那么我们就可以采用如下公式来计算权值 $W_i$，其中 $i=1,2,\cdots,N$。

$$W_i=\frac{\left(\frac{1}{D_i}\right)^r}{\sum_{j=1}^{n}\left(\frac{1}{D_j}\right)^r}$$

其中，$r$ 为非负实数。

在上述公式中，我们注意到如果 $r=0$ 的时候，那么一个树节点的 $N$ 个叶子节点的权值都相同，也就是它们被选择成为最优 Anycast 组成员的几率是相同的。而当 $r$ 的值趋向无穷大时，对于任意 $D_j<D_i$，其中 $i,j=1,2,\cdots,N$，并且 $i\neq j$，那么我们可以用 $\alpha_i/D_j$ 来代替 $D_i$，此处 $0<\alpha_i<1$，这样就有 $\lim_{r=\infty}\alpha i^r=0$，那么我们可以得到如下结果：

$$W_j=\lim_{r\to\infty}\frac{\left(\frac{1}{D_j}\right)^r}{\sum_{i=1}^{n}\left(\frac{1}{D_i}\right)^r}=\lim_{r\to\infty}\frac{\left(\frac{1}{D_j}\right)^r}{\left(\frac{1}{D_j}\right)^r\left(1+\sum_{i=1,i\neq j}^{n}\alpha_i^r\right)}=1,$$

$$W_i=\lim_{r\to\infty}\frac{\left(\frac{1}{D_i}\right)^r}{\sum_{k=1}^{n}\left(\frac{1}{D_k}\right)^r}=\lim_{r\to\infty}\frac{\left(\frac{1}{D_j}\right)^r\alpha i^r}{\left(\frac{1}{D_j}\right)^r\left(1+\sum_{k=1,k\neq j}^{n}\alpha_i^r\right)}=0$$

其中，$i,j=1,2,\cdots,N$，并且 $i\neq j$。上述结果表明当 $r$ 趋于无穷大时，树节点总是会选择距离自己最近的叶子节点作为最优 Anycast 组节点。

在本模型中，如果网络比较稳定，那么我们就可以选取一个固定的 $r$ 值，否则就根据网络的拥塞情况来动态地调整 $r$ 值。下面我们分两种情况讨论如何选取 $r$ 值：1）网络没有拥塞的情况。我们知道，客户发送的 Anycast 服务请求数据包要经过如下 3 个过程：网络传输、路由器排队并且被转发以及 Anycast 服务器处理，而我们只对前两个过程感兴趣，因为它们往往决定着客户获取 Anycast 服务的响应时间。当网络没有拥塞情况发生的时候，数据包到达路由器排队的时间可以忽略不计，那么网络传输时间就决定着客户获取 Anycast 服务的响应时间。这种情况下，我们希望距离树节点最近的 Anycast 组成员被选取为最优 Anycast 组成员。因此，在网络没有拥塞的情况下，$r$ 值要尽量大。2）网络出现拥塞的情况。当网络出现拥塞时，数据包到达路由器排队等待转发的时间会成为客户获取 Anycast 服务的响应时间的决定因素。这种情况下，我们希望把客户的 Anycast 服务请求数据包分散到以树节点为子树的不同的 Anycast 组成员上处理，以便减少排队所带来的延迟。因此，在网络出现拥塞的情况下，$r$ 值要尽量小。

### 3.5 路由分析

上面已经提到过，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 3.4 节中所描述的权值来获取最优组节点。其中，叶子节点到达树节点的距离可以根据不同的服务质量要求而采取不同的度量单位，比如当前所处理的会话数、组节点到达树节点的跳数等等。我们假设采用跳数为度量单位，计算权值公式里的 $r$ 取值为 1，那么我们得到建立在图 1 基础之上的 Anycast 树，如图 2 所示。

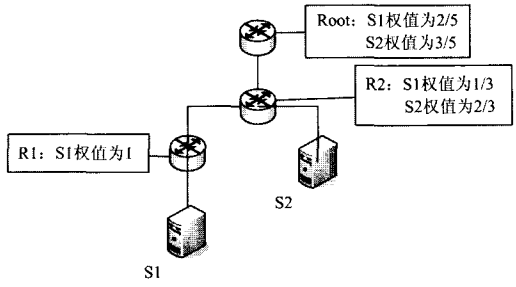


图 2 Anycast 树节点的权值

如图 2 所示，S1 与 S2 分别为组节点，S1 在 R1 的权值为 1；而 S1 与 S2 在 R2 的权值分别为 1/3 与 2/3，而 S1 与 S2 在 Root 的权值分别为 2/5 与 3/5。

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，所以它选取权值最大的 S2 为最优 Anycast 组节点。然后根据孩子记录表中对应的子节点的 Unicast 地址，将该请求消息转发给 S2，至此获取了以树节点 R2 为根节点的子树中的最优组节点 S2。S2 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。

在本通信模型中，由于网络的拥塞情况可能随时变化，因此本模型中的每个树节点可以根据当前的网络拥塞情况来动态地选取 $r$ 值，以便更好地提供 Anycast 服务。如果在一个树节点中有多个叶子节点的权值相同，那么我们采用轮流分

配的原则，依次将它们作为最优 Anycast 组节点。

## 4 性能分析

为了验证本 Anycast 通信模型的有效性和高效性，我们在 IPv6 模拟环境下实现了此模型，并将此模型与现有的 Anycast 通信模型进行了比较与分析。

在 IPv6 模拟环境下，我们将 32 个节点连接到万兆以太网上，同时利用 Modelnet 来模仿广域网物理拓扑结构。在我们的试验中，每个节点都是一个路由器，并且都与一台主机（所有主机的处理能力都相同）相连。我们设定一个 Anycast 组包括 6 个成员。为了有效地测试网络拥塞情况，我们设置路由器之间链路层通信能力为 45Mbps，客户端与路由器以及 Anycast 组成员与路由器之间的链路层通信能力为 12Mbps。

在上述的试验环境中，我们采用跳数为距离度量单位，实现了本模型以及现有 Anycast 通信模型下的 Anycast 服务。现有 Anycast 通信模型下的 Anycast 服务是指客户端的 Anycast 服务请求由距离（本试验采用跳数为度量单位）自己最近的 Anycast 服务器处理，我们通过搭建静态路由表实现了现有的 Anycast 通信模型下的 Anycast 服务。

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 值与现有的 Anycast 通信模型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。我们通过客户端在上述两种实现方式中获取同样的 Anycast 服务，得到如下的 TRT 性能分析图（$R=TRT_{Normal}/TRT$）。

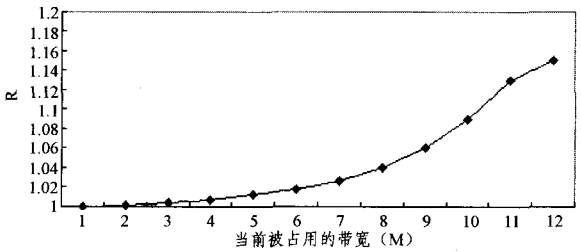


图 3 TRT 性能分析图

其中，$R$ 为在本模型中客户获取 Anycast 服务的 TRT 值与现有的 Anycast 通信模型下客户获取 Anycast 服务的 TRT 值的比值，$TRT_{Normal}$ 为在现有的 Anycast 通信模型下客户获取 Anycast 服务的 TRT 值，$TRT$ 为在本模型中客户端获取 Anycast 服务的 TRT 值。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现有的 Anycast 通信模型下客户获取 Anycast 服务的响应时间。这是因为当网络出现拥塞时，本模型能自动地调节分配策略（即调整 $r$ 值），将 Anycast 数据包分配给不同的 Anycast 服务器来处理，这样大大减少了数据包在路由器排队以及在 Anycast 服务器排队的时间，从而大大减小了 TRT 值。在本试验中，在网络拥塞的情况下，$r$ 取值为 0；不拥塞的情况下，$r$ 取值为 1000。

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。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，本模型采用跳数），查找到最优组节点。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。同时，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 树信息的分布式维护与管理，从而实现了负载均衡的作用。最后，本模型根据网络的拥塞情况会采取不同的策略把不同客户发出的服务请求消息分配给不同的最优 Anycast 组节点处理，这样使得 Anycast 服务请求均衡地分布在 Anycast 组成员之间，从而得到高效的处理。

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

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

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

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

[5] Ballani H, Francis P. Towards a Global IP Anycast Service // 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 // 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 // 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