IPv6 中解决 Anycast 扩展局限性的一种通信模型

王晓喃[1,2] 钱焕延[1] 钟 林[1]

(南京理工大学 南京 210094)[1] (常熟理工学院 江苏常熟 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 组数成比例增长,从而构成了 Anycast 的可扩展性问题。本文提出了一种建立在 Pastry 基础之上的 Anycast 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,同时此模型也实现了 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。本文同时也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, Pastry, 节点

A Communication Model on How to Solve Anycast Scalability in IPv6

WANG Xiao-Nan[1,2] QIAN Huan-Yan[1] ZHONG Lin[1]

(Nanjing University of Science & Technology, Nanjing 210094)[1]

(Changshu Institute of Technology, Jiangsu, 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 Pastry in this paper. For 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 Anycast tree's information in order to fulfill the load balance. This paper deeply analyzes and discusses the feasibility and validity of this communication model, and argues that it supports the large-scale Anycast group.

Keywords Ipv6, Anycast, Pastry, 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 地址所标识的一组接口中最近的一个(这里的最近是按路由协议的距离量度来计算的)。图 1 说明了 Anycast 的这种功能。

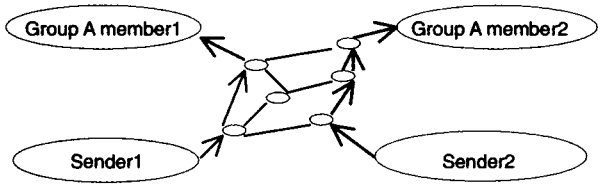

图 1 Anycast 服务

图 1 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离发送者最近的一个组成员(接口)。

Anycast 有广泛的应用,例如,把一个 Anycast 地址分配给所有的域名服务器(DNS),当一个用户从一个网段移动到另一个网段后,他不需要重新配置其本地的 DNS,主机可以使用全局的 Anycast 地址访问距离他最近的本地 DNS。

不难看出,Anycast 是一种非常有用的服务,它在许多应用领域发挥着重要的作用。随着网络新应用、新服务的不断涌现,对它的需求也在不断增长。但是,由于它的研究才刚刚起步,所以在许多方面还存在着制约这种服务实施和发展的种种问题,Anycast 的扩展局限性就是其中之一。

2 Anycast 扩展局限性的分析

IPv6 中的 Anycast 地址是从 Unicast 地址空间中进行分配的,所以 Unicast 和 Anycast 地址从结构上没有任何区别。这样做的目的是为了缓解 Anycast 带来的路由表规模扩张问题,同时也能充分利用现有的路由资源,使得 Anycast 的路由问题变得更简单。传统的 Unicast 路由方法可以将所有共享同一个 Unicast 前缀的目的地址聚合为一条路由项,以达到减小路由表规模的目的,这样,它可以在 Unicast 前缀允许的地址空间中扩展目的主机(或者子网)的数量,也就是说,Unicast 通过层次聚合的形式可以实现扩展性。但是,Anycast 却不能通过这种层次聚合来实现其扩展性。

一个 Anycast 地址表示的是共享某个特性的结点组,这些结点可能分散在互联网的各个地方。如果用 Unicast 路由协议路由 Anycast,则每个全球 Anycast 地址必须作为独立的路由表项处理。这种要求使得路由表会随全球 Anycast 组数成比例增长,从而构成了 Anycast 的扩展性问题。

IPv6 将每个 Anycast 组成员限制在共享一个地址前缀的

王晓喃 博士研究生,研究方向为网络安全与应用;**钱焕延** 教授,博士导师,研究方向为网络安全与应用;**钟 林** 硕士研究生,研究方向为网络安全。

特殊拓扑区内，在这个拓扑区域中，Anycast 地址在单播路由系统中是独立的表项，在该拓扑区域外，Anycast 地址被汇聚到其所在区域的地址前缀的路由项中传播。通过把 Anycast 组限制在一个预定义的区域内，IPV6 减轻了 Anycast 的可扩展性问题，但是并没有根本解决它，因为全球 Anycast 地址仍然需要作为独立的路由项在互联网中传播。

在这种情况下，本文提出了一种解决 Anycast 扩展局限性的通信模型。下面就此模型给以具体的分析和讨论。

3 解决 Anycast 扩展局限性的通信模型

本模型是建立在点对点应用的底层网络协议 Pastry 基础之上的。所以，在详细讨论本通信模型之前，首先介绍一下 Pastry 的基本原理。

3.1 Pastry

Pastry 是点到点应用的一种底层协议，具有分散性、可扩展性、高容错性以及高效性等特点。Pastry 网络节点可以在 Internet 网络中自动形成一个覆盖网络，实现均衡负载、备份、路由等功能。

在 Pastry 中，每个加入 Pastry 的节点都被赋予一个随机生成的 128 位 NodeID，并且每种应用都用一个 Key 来表示。Pastry 将每个应用都分配到某个节点上来实施，其中，实施这个应用的节点的 NodeID 值与此应用的 Key 值之差的绝对值最小。当主机请求某种应用时，Pastry 会将该请求自动转发到负责实施该应用的节点上。在 Pastry 中，每个节点只维护一个很小的路由表，该表中含有 $\log N$ 个表项，此处，N 为整个 Pastry 覆盖网路中的节点数。每个表项都记录着 NodeID 与其 IP 之间的对应关系。所以，当主机请求一个应用服务时，该请求消息至多经过 $\log N$ 次 hop 就可以到达目的节点（即负责实施该应用服务的节点）。

3.2 Anycast 树的建立

在本模型中，每个 Anycast 组被赋予一个组号 GroupID，它相当于 Pastry 中用于表示应用的 Key 值。那么，NodeID 值与 GroupID 值最接近的节点就作为此 Anycast 组的树根节点。

本模型定义了两种 Anycast 组节点：一种为能够提供 Anycast 服务的组成员节点，称作 Anycast 组成员节点，简称组节点，另外一种是不能提供 Anycast 服务而只用于支撑 Anycast 树框架作用的节点，称作 Anycast 树成员节点，简称树节点。不难看出，Anycast 树是由组节点和树节点组成的，它们之间没有交集，即组节点一定不是树节点，树节点也一定不是组节点。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。

当一个节点请求加入 Anycast 组的时候，首先将自己标记为组节点，然后发送 Join(GroupID) 消息，这个消息的应用 Key 值就是所要加入的 Anycast 组的组号 GroupID，所以，Pastry 会把该消息路由到此 GroupID 所表示的 Anycast 树的根节点。在路由过程中，Join 消息所经过的每个节点在接收到它之后，都会检查自身是否为该消息中的 GroupID 所代表的 Anycast 树的树节点或者组节点，如果是，就将发送 Join 消息的节点作为自己的孩子节点，并将 Join 消息中的相关参数加入到自己的孩子节点记录表中，这些参数包括发送 Join 消息节点的 Unicast 地址、权值、到达本节点所经过的 Hop 数、以及 GroupID 等信息，同时，停止转发所接收到的 Join 消息；如果不是，那么该节点首先将自己标记为 Anycast 树节点，同时，建立一个孩子节点记录表，并将发送消息的节点作为自己的第一个孩子节点，将 Join 消息中的相关参数（同上）加入到自己的孩子节点记录表中，同时记录下本节点的父节点的 IP 地址（即下一跳的 IP 地址）。最后，它用自己的 Unicast 地址取代原有 Join 消息中的 Unicast 地址，并修改相应的参数（例如跳数与权值等，修改的方法参见以下的章节），将其发送出去。

这里需要说明的是，如果一个树节点请求成为本 Anycast 树的组节点时，它只需要把自己标记为组节点，并记录下自己的相关参数，而不需要发送 Join 消息，如图 2 所示。

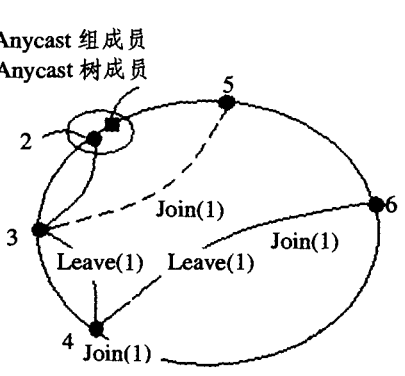

图 2 Anycast 树的建立过程

在图 2 中，GroupID 为 1，NodeID 为 2 的节点作为此 Anycast 组的根节点，因为它的 NodeID 值与 GroupID 值最接近。首先，节点(6)将自己标记为 GroupID 为 1 的 Anycast 组的组节点，并发出 Join(1)消息要求加入此 Anycast 组，Pastry 将该消息首先路由到 NodeID 为 4 的节点，此节点接收到该消息之后，将节点(6)加入到自己的孩子列表，并将自己标识为该 Anycast 树的树节点，然后转发该消息到下一跳(3)，同样，节点(3)接收到该消息之后，将节点(4)加入到自己的孩子列表，并将自己标记为该 Anycast 树的树节点，然后转发该消息到下一跳(2)。节点(2)接收到该消息之后，将节点(3)加入到自己的孩子列表，并将自己标记为该 Anycast 树的树节点，因为它是根节点，所以，不再转发该消息。接下来，NodeID 为 5 的节点也申请成为 GroupID 为 1 的 Anycast 组的组节点，首先它将自己标记为 GroupID 为 1 的 Anycast 组的组节点，并发送 Join(1)消息请求加入该组，那么，Pastry 首先将该消息路由到 NodeID 为 3 的节点，节点(3)接收到此 Join(1)消息之后，将该节点(5)加入到自己的孩子列表，因为节点(3)已经被标记为 GroupID 为 1 的 Anycast 组的树节点，所以，不再转发该请求。最后，树节点(4)申请加入 Anycast 组成员，那么，它将自己标记为 Anycast 组节点，并且记录下自己的相关参数，而无需发送 Join(1)消息。

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。这里分为两种情况：一种是 Anycast 组节点为叶子组节点的情况，另外一种是 Anycast 组节点为非叶子组节点的情况。

如果一个叶子组节点申请离开某个 Anycast 组，它首先删除自身组节点的信息，然后发送一个 Leave(GroupID)消息

给它的父节点，父节点接收到这个 Leave 消息之后，它会检查自身对应 Leave 消息中的 GroupID 的 Anycast 组的孩子节点记录表并从中删除此叶子组节点，如果此父节点本身也是该 Anycast 树的组节点，那么到此为止，一个叶子组节点已经成功离开这个 Anycast 组；否则，如果这个父节点只是该 Anycast 树的树节点，那么它需要判断此时的删除发送 Leave 消息节点之后的孩子节点记录表是否为空，如果为空，那么它将删除自身的树节点信息，然后继续发送一个 Leave (GroupID)消息给它的父节点，这个过程一直重复直到根节点或者某个节点，该节点为此 Anycast 的组节点，或者它本身是树节点并且在删除申请离开的节点之后，其对应 Anycast 组的孩子记录表不为空。如果一个非叶子组节点申请离开某个 Anycast 树，它只需要把自己标记为树节点，并且删除自己与组节点相关的参数(例如权值等)即可，此时，Anycast 树节点已经不再提供 Anycast 服务，而只是负责接收与转发 Anycast 树中的消息。

如图 1 所示，如果组节点(4)请求离开 Anycast 树，它只需要将自己标记为树节点，并且删除自己相关的参数即可。接下来，NodeID 为 6 的节点请求离开 GroupID 为 1 的 Anycast 树，它首先删除自身的组节点相关参数，然后发送 Leave(1)消息到它的父节点(4)，父节点(4)接收到该 Leave 消息之后，根据消息中的 GroupID 删除相应 Anycast 组的孩子组节点(6)，由于现在节点(4)本身只是此 Anycast 的树节点，所以，它需要检查此时该 Anycast 组的孩子记录表是否为空，因为此表为空，所以，它先删除自身树节点的参数，然后继续发 Leave(GroupID)消息给其父节点(3)，父节点(3)接收到 Leave 消息之后，同样根据消息中的 GroupID 删除相应 Anycast 组的孩子节点(4)，由于节点(3)本身也只是此 Anycast 的树节点，所以检查 Anycast 组的孩子记录表是否为空，因为此时该表不为空(NodeID 为 5 的节点是它的孩子节点)，所以不再发送 Leave 消息，至此，节点(6)成功地离开 GroupID 为 1 的 Anycast 组。

综上所述，Anycast 组节点的加入和离开都是分布处理的，并不是集中在某个固定的节点上，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，同时也实现了负载均衡的作用。此外，Join 和 Leave 消息的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。

3.3 路由分析

上边已经提到过，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，本模型会将该请求路由到最佳 Anycast 组节点上处理。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 Key 值，即 GroupID，Anycast 服务申请消息通过这个 GroupID 可以被 Pastry 朝着 Anycast 根节点的方向路由。在路由的过程中，每经过一个节点，该节点都会检查自己是否为该 Anycast 的节点(包括组节点和树节点)。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，组节点的树权值为其自身节点与其孩子节点的权值中较大的值，而树节点的树权值为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：

$$Val_{组节点i}=Max(N_{组节点i},\overset{k}{\underset{1}{Max}}(N_{孩子节点j}))$$

$$Val_{树节点i}=\overset{k}{\underset{1}{Max}}(N_{孩子节点j})$$

$$N_i=(K_1\times B_i)\times(K_2\times C_i)/(K_3\times Hop_i)$$

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，B_i 表示节点 i 当前的带宽，C_i 表示节点 i 自身的处理能力，Hop_i 表示孩子中权值最大的节点到达节点 i 的跳数，K_1，K_2，K_3 分别为带宽，处理能力以及跳数的调整参数。这些参数根据服务的性质与质量要求，可以采用不同的值。图 3 是在图 1 的基础上建立起来的带有权值的 Anycast 树，这里为了方便计算，K_1，K_2，K_3 都取值为 1。

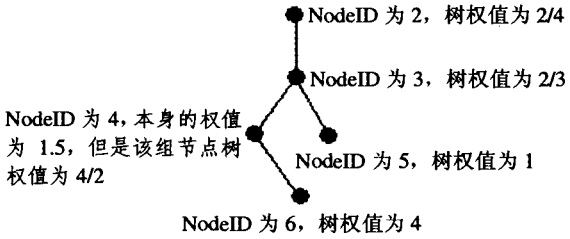

图 3 Anycast 节点的权植

如上图所示，组节点(6)为一个叶子组节点，所以此节点的树权值就是它本身的权值 4，组节点(5)也是一个叶子组节点，其组节点的树权值也是它本身的权值 1。组节点(4)自身的权值为 1.5，而其孩子节点的权值为 4/2=2，这里的除数 2 是指节点(6)到达节点(4)的跳数，因为孩子节点(6)在本身的位置跳数为 1，而到达节点(4)之后，其跳数要增加 1，所以，组节点(4)的树权值为 2，同时，组节点(4)记录下此树权值的来源指向为节点(6)。树节点(3)的树权值为其孩子节点(4 和 5)树权值最大的值 2/3，其来源指向为节点(4)，而树节点(2)只有一个孩子节点，所以它的树权值为 2/4，树权值来源指向为节点(3)。这样，当一个 Anycast 服务请求消息到达树节点(3)时，因为它是一个树节点，所以，根据其树权值的来源指向，将该请求消息转发给组节点(4)，同理，组节点(4)将此消息转发给组节点(6)，至此，获取了以树节点(3)为根节点的子树中的最优组节点(6)。

这里需要说明的是，某些 Anycast 树可能只包括一个根节点，而这个根节点又可能只是 Anycast 的一个树节点，在这种情况下，Anycast 树的根节点会向发送服务请求的节点返回一个错误信息，通知其服务不可达。

在本方案中，每个 Anycast 组节点的 B_i 与 C_i 值都是随着时间的变化而变化的，所以每个节点的树权值也会随之相应变化。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。

4 性能分析

本模型是建立在点对点应用的底层网络协议 Pastry 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，当一个节点发送一个 Anycast 服务请求时，此请求所到的第一个 Anycast 节点是整个 Anycast 树中距离源节点最近的节点，然后以此节点为子树根节点，再根据其子树成员的带宽、处理能力以及距离子树根节点的跳数等综合参数，按照一定的算法查找到最佳组节点。这样就可以保证该服务请求得到最高效的处理，提高服务质量，缩短响应时间。不难看出，本模型查找最优 Anycast 组节点的复杂度为 $O(\log(n))$，其中 n 为子树的节点个数。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，从而真正意义上解决了 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其孩子节点的信息，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，在本模型中不同主机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。

上述所有这些特点都充分地说明本模型可以很好地支持高效的、大规模的 Anycast 组的建立。

目前，该模型在 IPv6 的模拟环境下运行良好。

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 Partridge C, Mendez T, Milliken W. Host anycasting service. RFC1546, 1993

2 Deering S, Hinden R. Internet Protocol Version 6 (Ipv6) specification, RFC 2460, 1998

3 Hinden R, Deering. IP version 6 addressing architecture. RFC 2373, 1998

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

5 Johnson D, Deering S. Reserved Ipv6 Subnet anycast addresses. RFC2526, 1999

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

7 Narten T, Nordmark E, Simpson W. Neighbor discovery for IP version 6 (Ipv6), RFC1970, 1996

8 Huitema C. Routing in the internet. Prentice Hall, 1996

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

（上接第 24 页）

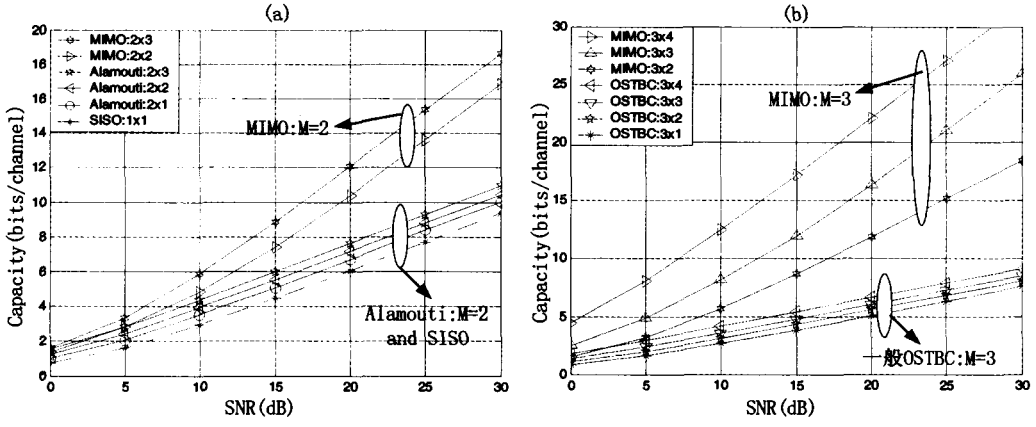

图 3 MIMO 同 STBC 容量对比 (a)为 $M=2$，Alamouti 和 MIMO 容量比较(b)为 $M=3$，OSTBC 和 MIMO 容量比较

3 Telatar E. Capacity of multi-antenna Gaussian channels. AT&T-Bell Labs Internal Tech. Memo June, 1995

4 Alamouti S M. A simple transmitter diversity scheme for wireless communications. IEEE J on Sel Areas In Comm, 1998, 16(8): 145I～1458

5 Hassibi B, Hochwald B M. High-rate codes that are linear in space and time. IEEE Trans. On Information Theory, 2002, 48(7): 1804～1824

6 Tarokh V, Jafarkhani H, Calderbank A R. Space-time block codes from orthogonal designs. IEEE Trans. On Information Theory, 1999, 45: 1456～1467

7 Foschini G J. Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. Bell Labs Technical Journal, Autumn, 1996. 41～59

8 Wolniansky P W, Foschini G J, Golden G D, Valenzuela R A. V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel. 1998 URSI International Symposium on Signals, Systems, and Electronics, 1998. 295～300

9 3G TS 25. 221 V3. 3. 0(2000. 3). http://www. 3gpp. org/