软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(1):334−350 [doi: 10.13328/j.cnki.jos.006435]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

任播研究综述*

周敏苑, 郑嘉琦, 窦万春, 陈贵海

(计算机软件新技术国家重点实验室 (南京大学), 江苏 南京 210023)

通信作者: 郑嘉琦, E-mail: jzheng@nju.edu.cn; 窦万春, E-mail: douwc@nju.edu.cn;
陈贵海, E-mail: gchen@nju.edu.cn

摘　要: 任播通过将相同 IP 地址分配到多个终端节点上, 利用 BGP 实现最佳路径选择. 近年来, 随着任播技术发展越来越成熟, 任播被广泛运用到 DNS 和 CDN 服务上. 首先全方位介绍了任播技术, 随后讨论了任播技术目前存在的问题并将这些问题归结为 3 大类: 任播推断的不完善, 任播性能无法保证, 难以控制任播负载均衡. 针对这些问题, 阐述了国内外最新研究进展, 总结了任播研究工作中的相关问题及改进方向, 为相关领域的研究者提供有益的参考.

关键词: 任播; IP 地址; 站点; 内容分发网络; 域名系统

中图法分类号: TP393

中文引用格式: 周敏苑, 郑嘉琦, 窦万春, 陈贵海. 任播研究综述. 软件学报, 2023, 34(1): 334–350. http://www.jos.org.cn/1000-9825/6435.htm

英文引用格式: Zhou MY, Zheng JQ, Dou WC, Chen GH. Survey on Anycast Research. Ruan Jian Xue Bao/Journal of Software, 2023, 34(1): 334–350 (in Chinese). http://www.jos.org.cn/1000-9825/6435.htm

Survey on Anycast Research

ZHOU Min-Yuan, ZHENG Jia-Qi, DOU Wan-Chun, CHEN Gui-Hai

(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

Abstract: Anycast uses BGP to achieve the best path selection by assigning the same IP address to multiple terminal nodes.In recent years, as anycast technology has become more and more common, it has been widely used in DNS and CDN services. This studyfisrtlyintroduces anycast technology in an all-round wayand then discusses current problems of anycast technology and summarizes these problems into three categories: anycast inference is imperfect, anycast performance cannot be guaranteed, and it is difficult to control anycast load balancing. In response to these problems, the latest research progress is described. Finally, the problems in solving anycast problems and the direction of improvementare summarizedtoprovide useful references for researchers in related fields.

Key words: anycast; IP; site; CDN; DNS

任播 (anycast)[1−5], 又称为选播、泛播或任意播, 是 IPv6 中定义的一种新型通信服务, 是 IPv6 中 3 大通信方式之一, 用于在互联网上的多个物理节点之间分配流量. 这些站点会向网络中宣告相同的 IP 地址, 客户端流量通过 BGP 路由协议被路由到最近的站点上. 任播背后的思想是, 客户端想要将数据包发送到提供特定服务或应用程序的多个可能服务器中的任何一个, 但并不关心哪一个. 因此, 可以为任意数量的服务器分配一个任播地址, 每一个拥有相同任播地址的服务器都提供相同的服务.

任播技术具备了很多其他技术不具备的优势[6−9]: (1) 通过在全球部署任播站点, 使用 BGP 选择一个最近的站点为客户提供服务, 大大降低了客户访问延迟; (2) 当一个站点发生错误, 路由系统会将客户端流量路由到另一个

* 基金项目: 国家自然科学基金 (62172206, 61972254)
收稿时间: 2021-03-22; 修改时间: 2021-06-24, 2021-07-31; 采用时间: 2021-08-14; jos 在线出版时间: 2021-10-20
CNKI 网络首发时间: 2022-11-15

© 中国科学院软件研究所 http://www.jos.org.cn

服务站点上, 避免单点失效, 提高了服务鲁棒性; (3) 多个站点分摊流量, 保护源服务器免受 DDoS 的恶意攻击. 这些优势使得任播在部署全球化复制服务 (多个服务器提供相同的信息和服务) 时会具有很强的吸引力, 例如, 目前任播技术已经被广泛地用于如递归 DNS 服务, 权威 DNS 服务, 内容分发网络等基础网络服务中.

自任播发明以来就有许多相关研究, 文献 [10] 中将这些研究分成 7 类, 分别为针对任播体系结构、可扩展性问题、稳定性问题、任播协议安全问题、负载均衡问题、任播路由协议以及 QoS 质量问题的研究. 但随着任播技术越来越成熟, 任播应用越来越广泛, 相较于之前的 7 类问题, 学术界的研究重点逐渐向任播性能问题倾斜, 提升任播性能成为当下任播研究中最热门的方向之一. 通过对以往任播研究分类的思考以及对最新科研成果的总结, 我们将任播相关工作分为以下部分.

(1) 任播基本组件: 其中既包括了任播的基础设施如任播地址识别, 任播站点定位等问题, 同时兼顾了原先的任播体系结构部分;

(2) 任播性能: 涵盖了原先的可扩展性问题, 稳定性问题以及安全问题, 同时也包括了最基本的通信时延, 带宽大小等问题, 通过测量发现任播性能的不足, 针对性的解决任播现存的问题, 终极目标是目的是提升任播性能, 使其充分发挥自身的潜力;

(3) 负载均衡: 利用任播分摊网络各链路及服务器的负载.

针对这 3 个部分, 我们将任播现存的热门问题总结为以下 3 类: 任播推断不完善, 任播性能无法保证, 难以控制任播负载均衡. 解决这些问题一直是学术界和工业界的重要研究方向.

本文首先对任播进行系统性的介绍, 之后从任播的 3 类问题对任播的相关研究进行分析. 通过对不同研究进行的实验以及得到的结论进行综合性梳理、对比和分析, 观察不同研究者给出的任播改进方案, 为研究人员提供可靠的建议以及后续研究目标.

1 任播介绍及发展历史

1.1 任播介绍

在任播中, 一组服务器共享相同的 IP 地址, 并将数据从源计算机发送到拓扑最接近的服务器. 这有助于减少网络延迟和带宽成本, 缩短用户的加载时间, 并提高可用性.

IP 任播架构的基本组件主要有任播地址空间以及任播路由两大部分. 首先, 对于任播地址空间, 在 IPv4 中, 鉴于地址空间的限制, 不可能为任播地址分配一个单独的地址类, 因此, 在 IPv4 环境下, 任播地址将从可用的单播地址池中分配, 也就是说任播地址与单播地址在 IPv4 中并没有本质的区别, 这种做法有利也有弊, 具体的我们将在第 3.1 节进行介绍. 而在 IPv6 中, 开发者们使用特殊格式将任播地址加以区分.

在任播路由中, 任播数据包转发与单播转发没有区别: 路由器基于最短路径将数据包逐跳发送到最近的服务器. 具体的: 位于同一 AS 域的多个服务器对外宣告相同的 IP 地址, 路由器收到这些宣告后, 将其作为主机路由 (由于一个任播地址标识服务器的一个实例, 因此路由系统将到任播地址的路由视为主机路由) 并存入路由表中, 随后选择具有最近路由距离 (AS 条数或链路成本) 的那条路由存入转发表, 之后当路由器收到相应的数据包时, 查找转发表逐跳转发.

1.2 任播的历史

历史上任播最初是在 RFC1546[3]中提出并定义的, 它的最初语义是, 在 IP 网络上通过一个 anycast 地址标识一组提供特定服务的主机, 同时服务访问方并不关心提供服务的具体是哪一台主机 (比如 DNS 或者镜像服务), 访问该地址的报文可以被 IP 网络路由到这一组目标中的任何一台主机上, 它提供的是一种无状态的、尽力而为的服务. RFC1546 也论述了一些潜在的问题. 例如, IP 是无国界的, 而且不会记录较早的数据报传递到了哪里, 最终会导致同一个客户端的数据报会发送到不同服务器上. 任播的第一个有记录的应用是在 1994 年进行的“视频注册”实验. 在该实验中, 一个 UDP 查询被传输到一个任播地址以定位拓扑最近的“假定等效网络资源”. 在同一时期, ISP 开始将任播用于 DNS 服务. 在 1998 年, 互联网架构委员会 (IAB) 举办的路由研讨会上, 向互联网工程任务组

© 中国科学院软件研究所 http://www.jos.org.cn

(IETF) 申请成立关于任播的兴趣小组 (BOF) 用以开展研究任播优缺点的工作. 在随后 1999 年 11 月, IETF 成立了任播 BOF, 讨论了关于任播的很多用途, 其中关于将 TCP 与任播一起使用的设想没有得出明确的结论, 但是对于 DNS 服务, 尽管任播会引入一定的复杂性, 但是它的优越性也是无可比拟的. 另外, 学者们还指出任播将仅限于少数关键用途并且规定任播地址并不能作为源地址 (即任播的使用仅限于路由器以及目的地址). 在 2002 年, “通过共享单播地址分发权威名称服务器”中首次详细说明了 DNS 对任播的使用. 之后, 随着内容分发网络 (CDN) 的兴起, 许多 CDN 也将任播作为关键技术.

1.3 任播与单播, 多播比较

单播使用一对一连接, 其中每个目标地址被唯一的标识为单个接收终端, 单播允许源结点向单一目标结点发送数据报, 如图 1 (图中蓝色节点表示配置单播 IP 的节点, 箭头表示最终连接).

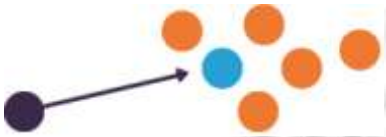


图 1 单播示意图

多播使用一对多连接, 允许源结点向一组目标结点发送数据报, 多播的一种常见应用是流音频. 如图 2 (图中蓝色节点表示配置组播 IP 的节点, 箭头表示最终连接).

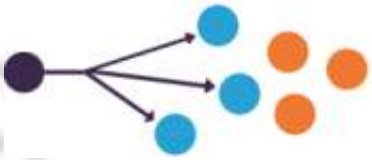


图 2 多播示意图

与上述两种传播方式不同的是, 任播使用一对任一连接, 允许源结点向一组目标结点中的一个结点发送数据报, 而这个结点由路由系统选择, 对源结点透明, 如图 3 (图中蓝色节点表示配置任播 IP 的节点, 箭头表示最终连接).

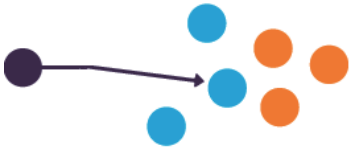


图 3 任播示意图

1.4 任播的集水区

根据任播的工作原理, 每个站点负责处理一片区域的用户请求, 那么这一片区域的用户属于相同的集水区 (catchment). 集水区是路由到特定任播站点的一组用户, 而集水区映射则对应于探测用户到特定站点的过程. 了解任播集水区对于任播性能 (吞吐量, 延迟和负载平衡), 抵御 DDoS 以及网络管理, 内容过滤都非常重要. 任播运营商会对任播部署进行工程设计以达到最大程度降低用户延迟的目的[9,11], 有些系统在全球范围内部署数十个甚至上百个站点, 在这种情况下, 运营商需要通过测量任播集水区的大小推算站点所需承载的流量, 个性化配置各站点容量大小.

除了性能之外, 任播集水区还可以迎合特定国家地区的政策进行内容过滤. 例如我国的就实施了 DNS 过滤以满足特定政策. 此类情况在全球范围内并不罕见[12−14], 因此网络政策的制定需要与集水区保持配套. 历史上也曾出现过两者不匹配的情况, 都造成了不良后果, 例如: 2010 年, 中国北京的 I-Root 站点的集水区被扩展到了境外, 使得中国边界之外的一些区域也受到了中国网络政策的影响[15].

© 中国科学院软件研究所 http://www.jos.org.cn

任播集水区测量工作自任播开始应用起就受到了广泛关注, 具体测量任播集水区的工作有: 早期的工作[16]将Open DNS (一个免费域名解析服务提供商) 解析器与 PlanetLab 和 Netalyzr 结合使用来映射任播服务的集水区. 之后, 随着技术发展, 最常见的测量方法是将可在全球各地提供物理探测点的公共或私有测量平台 (如 RIPE Atlas 和 PlanetLab)[1,8,16−19]作为预部署的测量平台, 这些系统可以在外部 (无需服务运营商的支持) 测量任播服务. 但它们的主要缺点是内置节点规模增长较慢, 并且相对于互联网用户的分布, 内置节点的部署经常会出现向某些区域歪斜的情况 (比如 RIPE 平台的节点主要集中在欧洲与北美洲). 除此之外, 任播运营商也能够通过分析自身的流量和服务器日志来预测当前的任播集水区[2,8,20]. 例如最近的工作[2,12]分别检查了微软 Bing CDN 和其他 CDN 关于任播的使用情况. 作为服务运营商, 日志分析不需要任何外部测量, 并且可以覆盖整个服务. 缺点是只能对已有服务进行分析, 而无法在部署前进行评估; 其次, 由于隐私, 存储或检索成本问题, 可能导致日志文件不可用. 文献 [21] 提出了一种新的测量任播集水区的方法, 其作者通过任播站点向 IPv4 命中列表[22]中的地址发送 ICMP Echo 请求 (pings), 然后观察其他任播站点收到的 ICMP Echo 回复, 从而确定每一个站点的集水区. 总的来说, 集水区测量仍面临以下两大测量难点.

(1) 网络的动态变化以及站点的添加或删除导致任播集水区的动态变化;

(2) 有限的测量点无法测得完整的任播集水区.

对于尚未部署的任播服务, 预测其集水区也就是预测任播部署后的运行情况, 通过预测可以挑选最优站点子集以及配置各个站点容量以提升任播服务性能[20]; 对于已经部署的任播服务, 测量其集水区有利于进行任播管理, 对于了解网络拓扑以及路由策略都有很大帮助[23]. 总而言之, 集水区的测量对于了解任播, 提升任播性能具有重大意义.

1.5 任播与 HTTP/TCP

由于任播的特殊工作机制, 任播的路由选择具有一定的随机性, 特别是当存在两条等长 AS 路径时. 这使得有些学者认为任播不适用于有状态的协议, 如 TCP, 这就是为什么任播在诸如 DNS (基于 UDP) 之类的无状态服务中广受欢迎. 对于任播 TCP, 主要有两种极端情况出现: 拆分路径路由和网络拓扑更改.

● 分离路径路由: 客户端计算机与两个或多个任播站点等距, 并且因为路由器的网络负载平衡机制导致传输到任播地址的数据包在这些站点之间交替.

● 网络拓扑更改: 客户端计算机开始与任播站点之一进行通信. 这时若有一处网络连接突然出现或终止导致另一处的任播站点成为最接近的站点, 从而导致客户端与原本站点的 TCP 连接中断, 其数据包被路由到另一个节点.

针对这两种情况, 有的学者认为任播自身的稳定性足够应用到有状态服务中[24−26](如 CDN), 当网络出现问题, 与先前站点建立的连接断开, 然后与新站点创建一个新的连接, 这样一来不需要任何复杂的操作还能维持 CDN 的正常运行. 如果路由经常更改, BGP 会注意到震荡的路由并惩罚该条路由, 一旦某条路由累积足够的惩罚值, 那么该路由就会在一定时间段内收到抑制, BGP 也就不再宣告该路由, 该机制维持了一个基本稳定的网络环境, 因此任播可以被用于有状态服务. 当然也有学者发明了有关提升任播稳定性的方法[27,28], 尽管这些方法是否已经被用于现实的任播部署中尚不明确, 但是在完善任播稳定性方面做出了不小的贡献.

2 任播的应用

任播的出现成为部署全球化复制服务有了新的选择, 其中就包括 DNS 和 CDN. 在介绍任播的应用之前, 我们简要讨论任播的优势及目标. 结合文献 [7,29−31], 我们列举了如下几个.

(1) 弹性: 文献 [31] 中提及任播两个目标之一就是提升 DNS 基础结构应对拒绝服务攻击 (denial-of-service, DOS) 的弹性. 最常见的 DOS 是通过发送大量伪造的查询请求, 使得被攻击方资源被消耗殆尽 (CPU 超负荷或内存不足) 从而无法响应正常查询. 此类问题并不能通过简单的提升服务器性能得到解决, 因为在当前大多数服务部署中, 服务器已经可以承载比其周围网络更高的攻击负载, 导致服务器无响应的原因更多的是因为网络拥塞而非服务器查询超载. 因此使用任播的一大期望就是通过部署大量的全球节点以分摊攻击流量从而减弱甚至消除 DOS 造成的影响. 从理论上来看, 部署的节点越多, 部署的范围越广, 节点故障或攻击可能导致大范围服务中断的可能性就越小.

© 中国科学院软件研究所 http://www.jos.org.cn

(2) 可扩展性: 顾名思义, 可以通过简单地增加站点的方法来解决服务器资源不足的问题. 以 DNS 为例, 服务随着互联网的不断发展, DNS 查询数量也急剧增长, 单个服务器必然无法承载如此庞大的查询量, 这时增加服务器站点就成为解决瓶颈的可行方法. 而任播的出现使得增加站点变得更加方便简单, 因为每个站点都提供相同的服务, 并且对客户透明, 同时也无需分配额外的地址段.

(3) 快速故障转移: 由于部署大量的分布式节点, 单点故障导致服务不可用的问题可以得到缓解, 当一个站点失效, 通过删除该站点对应的路由, 原本路由到出错站点的流量会被路由到其他正常节点上, 一旦该错误站点恢复, 那原先的流量也会随之复原. 因此, 任播中的故障转移速度取决于基础路由算法的收敛速度, 这一过程可以是快速的 (OSPF) 也有可能是缓慢的 (BGP). 需要注意的是该目标与弹性的区别在于: 弹性主要针对 DOS, 而快速故障转移主要针对单个站点失效.

(4) 性能提升: 任播的另一个目标是提升服务质量. 通过在用户附近部署站点, 大大降低了用户访问时延, 由于任播使用 BGP 进行路由, 不仅能够有效传递数据包, 而且通过路径选择可以选择拓扑最近的站点实现数据包的最优传输. 然而, 网络拓扑与地理位置并不密切相关[32], 因此在地理位置上靠近客户端部署站点不一定是使查询时间最小化的有效策略. 任播提升性能这一目标是否完全实现我们将在第 3.2 节进行说明.

(5) 可靠性: 最后, 任播应该增加服务的可靠性. 在客户端附近部署站点应通过减少查询必须遍历的网络元素的数量来提高可靠性. 例如, TCP 连接需要知道两端的地址和端口, 以及另一端的缓冲区大小或窗口以及其他信息. 如果没有正确的信息, 该协议将无法理解正在发生的事情, 并会终止连接. 显然, 有状态通信的端点保持同步很重要. 连接的所有数据包必须到达同一目的地, 以使两端都满意. 因此, 可靠性对于单播来说很容易达成, 因为单播可以保证给定地址只有唯一目的地, 但是任播的自然属性是无状态连接服务[33], 也即当前数据包的传递并不依赖于前一个数据包. 因此, 随着部署节点数量的增加, 路由表中竞争的路由数量也随之增加, 在这些情况下, 路由搅动和查询失败的可能性都将增加.

2.1 任播在 DNS 中的应用

首先, DNS 的一个节点表示在特定位置的一组 DNS 服务器以及相关联的网络设备. 使用了任播的 DNS, 每个 DNS 查询会被发送到最佳节点, 该节点是根据所使用的路由协议所确定的最近节点. 发出 DNS 请求的客户端通常不知道与哪个节点进行通信, 但是, 大多数任播部署都通过特殊的 DNS 查询提供此信息. 另外除了通过 DNS 宣告的任播 IP 地址访问 DNS 服务器外, 也可以使用单播 IP 地址访问 DNS 服务器, 我们将其称为内部地址.

对于最常见的路由协议 BGP, 使用任播服务的 DNS 会将它的每个节点向网络中宣告同一网络前缀 (服务前缀) 的可达性信息, 其中包含任播 IP 地址. 来自不同节点的宣告将在域间路由系统中竞争, 并根据 BGP 路由选择过程进行传播. 由此出现了两种类型的任播节点: 全局节点和本地节点. 全局节点旨在为整个互联网提供服务, 因此必须具有足够的带宽和处理能力来处理全球客户请求. 本地节点旨在仅向称为节点的服务区的有限区域提供服务. 本地节点主要通过 BGP 策略机制实现: 一种方法是通过人为地延长了全局节点声明的 AS 路径. 由于 BGP 路由选择算法使用的最重要的指标之一是 AS 路径的长度, 因此这将导致首选本地节点通告的路径. 另一种方法是将本地节点发出的公告标记“no-export”的 community 属性值, 该属性值要求其路由公告不传播到其他 AS.

现实中, 不同的根服务器使用不同的部署策略. 为了说明方便, 我们将仅包含全局节点的任播部署称为平面部署 (flat), 如果包含少量彼此靠近的全局节点且包含大量的本地节点, 我们称之为分层部署 (hierarchical), 最后, 将既包含大量本地节点又包含大量全局节点的部署称之为混合部署 (hybrid). 这 3 种部署策略对应到 DNS 根服务器部署中的例子分别为 J-root, F-root, K-root. 对于平面部署, 一方面增加了 DNS 服务的鲁棒性 (因为所有节点都是全局节点, 单个节点失效所导致的请求失败可以通过转移到其他健康节点实现快速恢复); 但另一方面, 由于全局节点必须部署在互联网连结性和带宽都充足的区域, 部署成本高且无法兼顾“偏远”的服务区域. 相对于平面部署, 分层部署既有优点也有缺点, 优点是由于本地节点不需要处理来自全球的客户请求, 它可以部署在互联网连结性和带宽都有限制的区域, 这使得任播在部署时可以有多种选择. 同样, 本地节点的失效可能会导致其服务区客户的服务质量大幅下降甚至可能出现中断服务的结果. 另一方面, 不在本地节点服务区域中的客户端将向全局节点发

© 中国科学院软件研究所 http://www.jos.org.cn

送 DNS 查询, 由于全局节点集中在较小的地理区域中, 这将导致遥远的客户端产生较高的查询延迟. 最后, 如果来自本地节点的公告被错误的传播到全球路由中, 那么节点和周围的网络基础结构可能会因为查询请求的激增而超负荷. 而混合部署兼具平面以及分层部署的优点, 并解决了部分缺点, 但由于部署节点数量较大, 导致成本偏高.

如今, 任播已大量运用于 DNS 根服务器之中, 文献 [9] 展示了部分根服务器的站点数量及分布, 由此可见任播对 DNS 的贡献.

2.2 任播在 CDN 中的应用

内容分发网络 (CDN) 在现代网络中起着重要作用, 随着互联网数据传输速率的提高以及消费者对慢速下载速度的容忍度降低, 尤其视频和语音应用程序在抖动和延迟方面特别敏感, 传统的大型网络运营商开始建设自己的 CDN 网络. CDN 是代理服务器的全球分布式网络, 可将内容以高可用性和低延迟交付给最终用户. CDN 的目标是通过从最接近最终用户的服务器提供内容来优化传输. 大多数 CDN 的基本架构非常简单, 由分布在互联网上的一组 CDN 站点组成[34], 这些 CDN 站点充当代理, 用户通过标准协议从 CDN 站点检索内容, 任何 CDN 都致力于将用户请求发送到最佳站点, 此过程通常称为重定向 (redirection)[35], 如图 4. 重定向是 CDN 中最重要也是最具挑战性的问题之一, 原因是并非所有内容都可从任一站点获得, 并非所有站点始终都在运行, 站点可能因任务过多而变得超载, 最重要的是重定向的首要任务是将用户连接至与其紧邻的站点以确保良好的用户体验. 目前主流 CDN 使用的最多的重定向机制是 DNS 和任播.

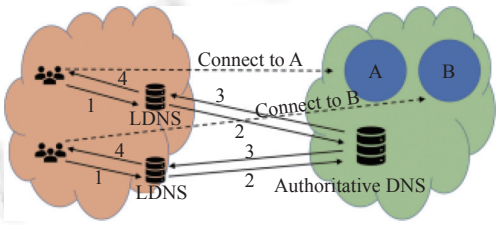


图 4 CDN 工作流程

● DNS: 如图 4, 客户将通过属于 CDN 的主机名获取 CDN 托管的资源, 客户端首先向通常由 ISP 配置的客户端本地 DNS 解析器 (LDNS) 发送 DNS 解析请求, LDNS 收到请求后会解析主机名, 并将其转发至 CDN 的权威域名服务器. 最终 CDN 根据 LDNS 信息, 返回最佳的站点 IP 地址. 但是由于 CDN 必须以 LDNS (而非客户端) 的粒度进行决策, 因此基于 DNS 的重定向面临一些挑战. 例如: LDNS 可能与它所服务的客户端相距较远, 或者单个 LDNS 可能服务较大地理区域的客户端, 因此权威域名解析器做出的最优站点选择可能并不能满足客户的需求, 这种情况在公共 DNS 解析器 (例如 Google Public DNS 和 OpenDNS) 中非常普遍.

● 任播: 同样的, 客户端首先向通常由 ISP 配置的客户端本地 DNS 解析器 (LDNS) 发送 DNS 解析请求, LDNS 收到请求后会解析主机名, 并将其转发至 CDN 的权威域名服务器. 此时, 使用任播的 CDN 不用考虑返回哪个站点的 IP 地址, 而是直接返回站点任播 IP 地址. BGP 根据选路原则将用户重定向到拓扑最近的 CDN 站点, 当用户被重定向至特定站点之后, 再由具体的服务器提供服务. 由于 CDN 通过任播进行用户流量的重定向, 而任播基于 BGP 选择最佳站点, 这样一来大大简化了 CDN 的任务负担. 与 DNS 重定向相比, 任播的优势在于每个客户端的重定向都是独立的, 避免了上述 LDNS 的问题, 当然任播也面临一些问题, 我们将在第 3 节进行介绍. 目前许多知名的 CDN 公司包括 CloudFlare, CacheFly, Microsoft/bing 都成功运行了基于任播的 CDN 并且取得了不错的效果.

2.3 任播在移动网络中的应用

在移动 IP 网络上, 每个移动节点都通过其归属地址进行识别, 当从本地链路移动到外地链路时, 移动节点首先从自动配置的无状态地址或来自 DHCPv6 的有状态地址获取转交地址 (care-of address), 并寻找最近的本地代理进行绑定更新 (global dynamic home agents discovery, GDHAD), 最终才能与通信节点进行通信. 而任播则是支持

© 中国科学院软件研究所 http://www.jos.org.cn

移动节点发现最近本地节点的最佳方案, 通过多个本地代理宣告相同的任播 IP 地址, 移动节点可以找到一个路由距离最近的本地代理并发送 ICMP 消息, 当本地服务器收到移动节点发送的 ICMP 消息后, 它会返回一个包含自身单播 IP 地址的 ICMP 回复包, 移动节点在收到 ICMP 回复后, 得到了最近的本地代理单播 IP 地址, 随后, 移动节点向新的本地代理发送绑定更新.

3 任播问题介绍

3.1 任播推断不完善

任播的推断主要包括对任播前缀的枚举以及对任播站点的定位. 对 IP 任播的了解不仅有助于表征任播性能, 故障排除和基础结构映射[36], 而且对于与安全相关的任务 (例如检查机制检测)[37]也都非常有用. 然而, 对任播推断的研究也有很多困难, 如: 不像多播和广播地址, 仅通过观察 IP 地址并不能分辨它是否属于任播 IP, 由于任播地址和单播地址的不可区分性, 至今还没有一个完整准确的任播地址数据集, 尽管在 IPv6 中, 使用特殊格式将任播地址加以区分[38], 但 IPv4 中部署任播的方法 (也可在 IPv6 中使用) 是为多个主机分配相同的单播地址, 这一事实使它对路由系统和终端用户都是不透明的, 这种不透明性给任播地址测量带来了挑战. 又例如, 虽然通过基于延迟的定位方法对单播地址定位取得良好的效果[39,40], 但对于任播来说, 此技术并不适用.

3.2 任播性能无法保证

关于任播的性能, 已经进行了很多相关研究, 尽管不同研究采用不同的数据集, 使用不同的衡量指标 (metric), 得到的结果都表明任播存在一定缺陷.

研究者们主要通过两种指标来衡量任播性能: 往返时间 (RTT) 和相对地理距离. 以 RTT 作为评价指标的主要有文献 [9,29,41−44], 其中文献 [41] 采用微软的 CDN 进行数据收集, 研究发现对于大多数客户端来说, 尽管缺乏集中控制, 任播仍然表现良好: 任播将大约 20% 的客户指向次优的边缘服务器. 而文献 [42] 在 F-root 以及 K-root 两个 DNS 根服务器上, 使用 PlanetLab 平台对任播引起的额外时延进行测量, 他们发现到达最低延迟站点的数据包很少, 但其余数据包的额外延迟开销也很小, 这说明任播将大部分客户请求都路由到次优站点. 文献 [29] 在同一年对 K-root 服务器进行研究, 得到了相同的结论. 更近一点的, 在 2016 年, 文献 [9] 使用 RIPE Atlas 探针来测量所有使用任播服务的 DNS 根服务器的 RTT. 他们得出的结论是, 拥有“一些站点”足以获得与拥有“多个站点”一样好的性能. 2010 年, 文献 [44] 评估了部署任播的根 DNS 服务器的整体延迟. 他们的结果表明, 从 2007 年到 2008 年, 任播整体延迟逐渐减少, 然后到 2009 年初逐渐增加. 鉴于当时的研究还处于早期阶段, 他们无法解释这种趋势的原因. 而该问题在文献 [45] 中得到解释, 随着站点数量的不断增加, IP 任播的性能通常会下降, 文中也指出该现象出现的根源为 BGP 的路由选择, 过多的站点往往会影响 BGP 做出最优选择.

也有研究使用了相对地理距离作为比较任播选择情况的指标, 2006 年, 文献 [46] 中分析了 C-root, F-root, K-root 的数据, 并报告了它们的平均附加距离 (超过其最接近站点的距离) 分别为 6 000 km, 2 000 km 和 2 000 km. 文献 [47] 对 K-root 的任播性能进行实验分析, 结果表明 45% 的客户端请求既没有被路由到最近站点, 也没有被路由到延迟最低站点. 最近的研究[45]对任播进行测量时也发现了同样的问题, 但是测量的结果并没有之前的研究那么乐观, 有近 2/3 的客户端请求被路由到了非最优的站点, 并且有超过 1/3 的查询请求被路由到 1 000 km 外的站点, 超过 8% 的查询求情被路由到 5 000 km 外的站点. 文献 [48] 证实了上述实验结果, 通过对 Root DNS 的研究, 作者发现任播导致的各种膨胀 (延迟膨胀, 路径膨胀)[49]在 Root DNS 中非常普遍, 影响了超过 95% 的用户, 但是这类膨胀对于用户的体验几乎没有影响, 原因是 DNS 中的缓存非常有效 (用户并不需要每次都向 Root DNS 发送请求). 当目光转向 CDN 中的任播时, 作者又发现只有 35% 的 CDN 用户经历了任何膨胀且经历的数量小于 Root DNS, 作者把这种现象归结于 CDN 使用广泛的对等连接以及其他工程技术. 这些结果表明, 之前关于任播效率低下的说法仅仅反映了任播在单个应用程序环境下的实验, 而不是任播的技术潜力, 并且证明了上下文在衡量系统性能时的重要性.

除了上述两种指标以外, 也有一些学者从其他方面入手研究任播, 如任播的亲和性 (affinity). 任播的亲和性衡

© 中国科学院软件研究所 http://www.jos.org.cn

量的是将来自客户端的连续任播数据包传递到同一任播服务器的程度[41]. 最早一批对任播的研究主通过分析根DNS日志来观察任播的性能, 比如文献[50,51], J-root和K-root的操作员基于服务器收集的客户日志分析任播的负载以及亲和性. 在负载分布方面, 两项研究均报告了其各自部署中负载满足偏斜分布, 而在亲和性方面两项研究出现了差异, J-root运营商报告的客户端实例呈现出较弱的亲和性, 也就是客户端的连续任播数据包被传递到不同的任播服务器上, 由此他们推测任播可能不适合有状态服务. 相比之下, K-root运营商发现他们的大多数客户都具有很好的亲和性. 另外文献[42,52]也对任播亲和性进行研究, 文献[42]使用PlanetLab平台针对K-root, F-root以及.org顶级域名部署进行分析, 在结论中, 他们报告任播具有中等的亲和性. 文献[52]同样针对DNS根节点进行测量实验并且报告任播具有较差亲和性. 而同样是对DNS根节点进行试验测量, Wei等人[24]却发现了相反的结果, 他们认为任播具有良好的亲和性, 大多数用户在访问任播服务时并不会更换站点. 但文中也指出, 对于少量用户, 访问任播服务时会经历频繁的站点切换, 且这种不稳定性是持续的. 文献[41]通过大规模的主动测量并部署一个小型任播服务发现以下结论: 任播除对很小一部分客户外的其他所有客户都具有很好的亲和性, 而小部分客户观察任播较差的亲和性可以归因于网络中的动态负载平衡机制, 另外, 该文献也指出单纯的任播部署不会实现各服务器的负载均衡, 但是可以通过运营商操控各个任播站点之上的BGP公告来实现粗粒度的负载均衡.

我们将不同文献采用的衡量指标及选取的研究对象总结为表1.

表1 有关任播性能研究文献总结

文献	年份	衡量指标	研究对象	实验平台(方法)
[8]	2015	RTT, 相对地理距离	CDN	JavaScript beacon
[9]	2016	RTT, 相对地理距离	Root DNS	RIPE Atlas
[24]	2017	亲和性	Root DNS	RIPE Atlas
[29]	2006	RTT	Root DNS: K-root	RIPE Atlas
[30]	2005	亲和性	Root DNS	PlanetLab
[41]	2006	RTT, 亲和性	Root DNS	King
[42]	2006	RTT, 亲和性	Root DNS: F- and K-root	PlanetLab
[43]	2013	RTT	Root DNS	King
[44]	2010	RTT	Root DNS	SEIL probes
[45]	2018	RTT, 相对地理距离	Root DNS	RIPE Atlas
[47]	2015	相对地理距离	Root DNS: K-root	RIPE Atlas
[48]	2021	RTT, 相对地理距离	Root DNS, CDN	RIPE Atlas
[50]	2004	负载, 亲和性	Root DNS: J-root	DNS log
[51]	2005	负载, 亲和性	Root DNS: K-root	DNS log
[52]	2005	亲和性	Root DNS	PlanetLab
[53]	2019	RTT	Root DNS, CDN	RIPE Atlas

综上所述, 任播无论是在传输时延还是物理距离方面都没有达到理想状态, 在亲和性方面, 对于任播的测量结果也没有达到公认的程度. 因此, 优化任播性能是当务之急.

3.3 难以控制任播负载均衡

任播的设计初衷是希望可以通过设立多个站点用于分摊流量以达到各站点负载均衡的目的. 但是根据任播的工作原理, 我们知道任播本身并不具备控制流量的能力——究其原因是BGP的选路策略不会考虑各站点的负载, 只会考虑如何选择最优路径. 从宏观上看, 任播多站点的设计分摊了原先单一站点的流量, 但是从每个站点的角度看, 各自站点的负载并不均衡, 甚至有可能出现某些站点超载的情况, 因此, 任播本身无法平衡服务器的负载. 目前任播技术被广泛地用于CDN中, 每个CND都希望为每个用户提供最优的服务, 尽管服务器过载的情况并不经常发生, 一旦发生, 若CDN无法做出合理应对, 无疑会影响用户体验. 如果CDN使用任播作为流量重定向策略, 单个站点有可能会因为用户流量过多而超载, 更令人担忧的是, 文献[41]指出受路由延迟收敛影响, 用户使用任播服务时并不能体验快速的故障恢复. 因此, 如何平衡任播负载是当前亟待解决的问题之一.

© 中国科学院软件研究所 http://www.jos.org.cn

4 任播问题研究进展

4.1 关于任播推断的测量研究

4.1.1 任播前缀测量

在网络测量中, 主要有两大测量方法: 主动测量和被动测量. 采用主动测量时, 研究人员可以根据需要主动的选择测量对象, 测量方法, 测量时间, 定制化的在网络中进行端到端的性能参数测量, 其优点显而易见: 测量的数据类型能够最大程度满足研究需求. 缺点是主动测量依赖于面向网络的测量系统并需要向网络中注入额外流量, 从而限制了主动测量的规模. 而被动测量则是通过收集已有的设备运行数据并加以数据分析最终得到需要的信息. 一般来说, 被动测量无需产生额外流量, 不会增加网络负担; 但是被动测量往往无法像主动测量那样简洁明了的获得所需数据, 仍需对测量结果进行复杂的数据分析, 而测量结果的局限性甚至会导致已有测量结果无法分析得到所需数据.

文献 [37] 使用主动测量方法, 通过延迟测量 (来自分布式 traceroute 代理) 和 BGP 路由信息 (来自公共路由器) 对任意播前缀进行检测: 当 IP 前缀的传输树中有多个位于不相交地理位置的本地 ISP 时, 此 IP 前缀很可能是任播. 文献 [1,54] 也使用主动测量的方法, 通过使用基于光速违规检测的延迟测量研究任播前缀: 对于给定的一个 IP 地址, 从每一个探测点 (vantage point, VP) 测量访问某 IP 的时延, 当两个观察点测得的时延和小于这两个观察点直线距离的光速传播时延, 那么就认为该 IP 地址属于任播 IP (如图 5).

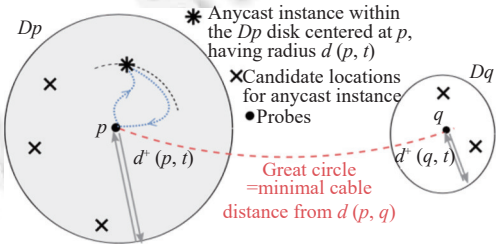


图 5 通过延迟测量发现任播实例 [54]

图中显示有两个探测点 (p, q) 分别对某 IP 地址测量 RTT 值 $(\delta(p, t), \delta(q, t))$, 将该值乘以光速得到探测点到 IP 地址距离上限 $(d^+(p, t), d^+(q, t))$, 随后根据这两个探测点的经纬度信息求出两点之间的物理距离 $d(p, q)$, 由于数据包的传递速度不可能大于光速, 若:

$$d(p,q)gt;d+(p,t)+d+(q,t),$$

则说明探测点 p, q 与两个不同的任播站点通信, 该 IP 地址为任播地址. 该方法虽然简单明了, 但是存在以下两点局限性, 首先采用光速作为数据包传输速度的上限有可能会导致两个临近的任播站点不易被发现; 另外, 在传输时间的测量上也有可能出现误差, 文献中测量传输时延的方法是测量 10 次 RTT 并将最小值取半, 该方法默认数据包传输的前向和后向路径是对称的, 但实际情况中两者可能并不对称, 因而引入一定误差.

文献 [55] 同样使用主动测量的方法, 不同的是, 作者通过将任播探测点作为源节点, 向目标 IP 发送 ICMP Echo 请求, 如果有多个探测点收到回复, 则认定该目标 IP 地址是任播地址. 具体方法如下: 对于单播来说, 全球任意一个探测点 (这些探测点使用任播地址) 对其发送请求数据包, 始终只有一个探测点收到回复数据包, 这是基于任播的稳定路由假设 [24], 而对于任播, 由每个探测点发出的数据包被路由到不同的任播站点上, 因此会有不止一个的探测点收到回复. 本文根据该差别, 设计出用任播测量任播的方法并将之命名为 MAnycast^2. 该方法的局限性也很明显, 成功识别任播 IP 需要探测点与至少两个任播站点进行通信, 由于具有许多上游运营商的任播网络通常操纵其 BGP 公告使得一些任播站点往往只服务于很小的区域, 所以当不满足此最小连接要求时, MAnycast^2 方法可能无法检测到任播服务.

上述实验方法均采用主动测量, 基于被动测量来获取任播前缀的方法不多, 目前为止只有一篇文献进行相关

© 中国科学院软件研究所 http://www.jos.org.cn

工作[56]. 因为任播和单播的部署模式不同, 作者提出了一种被动方法来检测任播前缀的方法, 他们使用来自路由收集器的公共 BGP 数据, 利用 BGP 路由信息来表征任意类型的前缀. 具体的, 他们选择了如下 BGP 路由信息.

- N: 上游 AS 的数量: 所谓上游 AS 定义如下: 假定由 ASn 宣布的前缀, 通过客户运营商关系或对等关系与 ASn 连接的 ASn 邻居 AS 的集合. 由于任播前缀在全球各地进行宣告, 那么任播前缀的上游 AS 数量会多余正常单播前缀上游 AS 数量.
- P1: 距离大于 1 的上游 AS 对的百分比: AS 对之间的距离代表两个 AS 之间存在的 AS 跳数 (任播较大).
- P2: 距离大于 2 的上游 AS 对的百分比 (任播较大).
- MD: 上游 AS 之间的最大距离 (任播较大).
- ML:AS 路径的最大长度 (任播较小).

对于上述 5 个特征, 直观上任播和单播之间存在明显的差异 (文中也通过实验证明了两者在这些特征上的差异), 因此选择这 5 个特征属性训练分类器, 他们分别使用决策树和随机森林模型训练分类器并在检测任播前缀时达到了 90% 的精度.

4.1.2 任播站点的定位研究

对于任播站点的研究, 就方法论而言, 已经采用如下的技术: (1) 发送特殊类 (CHAOS), 类型 (TXT) 和名称 (host-name.bind 或 id.server) 的特殊 DNS 查询请求. (2) 使用 traceroute, ping 等基于 ICMP 协议的工具测量.

文献 [9,30,53] 通过发送特殊的 DNS 查询, 将查询类型更改为“TXT”查询参数修改为“id.server”, 这样返回的查询结果中就会包含 DNS 服务器的地理位置. 然而, 不是所有使用了任播的 DNS 服务器都对 CHAOS 类查询回复自己的标识符 (包含所在地理位置), 即使进行回复, 也不一定总是遵循通用的命名标准 (通常会采用 IATA 机场代码来标记服务器所在位置), 另外, 文献 [40] 指出一些路径上的代理会修改此类回复. 因此, 作者提出修改现有的任播服务器以答复特殊的 DNS IN TXT 查询. 尽管这是一种有效的方法, 但这仅针对一些使用了任播的 DNS 服务器, 适用范围比较狭窄.

文献 [1,54] 提出了一种通用的任播站点定位方法, 该方法首先根据光速违规理论检测任播, 并采用最大独立集算法确定各个任播服务器所在地理范围, 最后根据得到的地理范围, 文章采用两个指标来确定任播服务器所在城市, 分别是 (1) 范围内各城市的人口数量以及 (2) 各城市距离范围边界的距离. 具体公式如下:

$$pi = \alpha \frac{ci}{\sum_j cj} + (1-\alpha) \frac{d(p,t) - d(p,Ai)}{\sum_j d(p,t) - d(p,Aj)},$$

其中, 参数 α 用于调整人口与距离在决策中的重要性, pi 表示站点部署在城市 i 的可能性, c 表示一个城市的人口, $d(p, t)$ 表示地理范围的半径, $d(p, Ai)$ 表示城市 i 距离范围中心的距离.

该方法基于的理论基础是文献 [57] 中认为的站点主要设立于人口密集的区域, 即大城市更有可能部署站点. 该方法最终可以达到 78% 的定位准确率, 其主要局限性在于探测点的数量有限, 学者们使用 RIPE 平台进行实验, 而 RIPE 提供的探测点在全球分布很不均匀, 具体体现在欧美多, 非洲以及南美洲的探测点较少, 而本文提出的方法需要全球各地的探测点用以发现更多的任播站点位置, 因此导致最终结果的不完善.

4.1.3 总 结

我们将对于任播推断的研究总结见表 2. 其中可以看出, 对于任播前缀的测量已经比较完善, 而对于任播站点位置的测量目前的方法并不多, 尽管大多数提供任播服务的公司会分享自己的站点分布图, 但想要进行任播的全面研究, 主动测量得到任播站点的位置的方法是必不可少的.

4.2 关于任播性能的研究

4.2.1 任播性能无法保证的原因

关于这个问题, 许多文献已经说明了原因: 文献 [45] 根据 DNS 根服务器请求的路由信息验证了等长 AS-PATH 是导致任播延迟增加的主要原因. 具体地说, 因为路由选择基于最短路径原理, BGP 在遇到等长的 AS-PATH 时会根据其他选路原则进行选路, 最终导致选择一条实际距离较长的路由, 从而增加往返时延. 同时, 文

© 中国科学院软件研究所 http://www.jos.org.cn

献 [56] 发现远程对等可能会对任播路由产生影响. 远程对等网络[58,59]是一种应用程序体系结构, 是在链路层进行互连的一种方式, 它使网络更平坦, 但会影响 BGP 路由, 导致 BGP 选择较短 AS 路径但实际距离较长的路由. 文献 [53] 从任播 IP 宣告的角度来看, 他们指出任播的性能与其上层运营商的数量有关, 并且不同的运营商具有不同的路由公告选项, 这使得网络中的路由信息非常复杂, 给 BGP 路由带来很大的干扰. 综上所述, 导致任播性能无法保证的原因有两个: (1) BGP 缺乏对网络基本拓扑的了解, 导致任播做出次优选择. (2) 远程对等会影响 ISP 的域内路由策略选择.

表 2 关于任播推断的文献总结

文献	研究对象	测量方法	测量平台	准确率 (%)
[1]	站点定位	被动测量	—	—
[9]	站点定位	被动测量	—	—
[30]	站点定位	被动测量	—	—
[54]	前缀枚举, 站点定位	主动测量	RIPE Atlas	78
[55]	前缀枚举	主动测量	PlanetLab	—
[57]	前缀枚举	被动测量	—	90

4.2.2 任播性能无法保证的解决方案

第一个被证明可行的解决方案是在文献 [41]. 作者探讨了任播配置的含义, 并为将来的任播部署提供指导, 最终提出所有任播服务应共享单个服务运营商. 这个猜想在文献 [45] 中得到了验证. 他们发现 C-root 根服务器采用了这种单个服务运营商的部署模式, 并且没有出现任何任播路由问题, 但是这种方法并不可能在实践中进行应用. 之后, 许多相关工作致力于解决任播问题.

在文献 [9] 中, 作者确定了实现合理延迟所需的任播站点的数量, 并提出了部署新站点的收益递减的问题. 在文献 [8,60] 中, 作者提出了一种基于 DNS 的解决方案来解决以下问题: 由于 CDN 具有两种不同的流量重定向机制——基于 DNS 的重定向和任播, 作者提出这两种方法虽然都有表现不佳的情况, 但两者并不同时发生且导致这两种方案出错的原因并不相关, 所以, 通过简单的预测方案, 在任播表现不佳时使用 DNS 重定向, 以提高任播性能.

最近的文献 [45] 通过实验表明: 如果 AS 在不违反通用路由选择策略的情况下更智能地选择路由, 则可以避免很多性能损失, 因此作者提出添加静态“提示”到 BGP 中, 用以辅助 BGP 做出路由选择. 一种简单, 具体的实现方式是指定 community 标签, 使用前 16 位来表示 AS 编号, 后 16 位对粗略的纬度和经度进行编码, 具体的: 纬度在–90°到 90°之间变化, 但人类居住地的纬度则大部分在–50°到 74°之间, 因此可以按 7 位编码. 经度在–180°到 180°之间变化, 因此可以轻松地在剩余的 9 位中进行编码. 但是, 这样一来, 需要为 BGP 路由器配置自身其纬度和经度, 并进行解码 BGP 社区标签中编码的纬度和经度, 最后通过计算才能获得到路由目的地的距离. 这无疑会引入计算开销, 有可能会对选路速度产生影响. 尽管该方法有一些局限性, 但是为 BGP 添加提示的想法我们认为是有发展空间的, 一个主要优点是, 无论提示类型如何, 它都可以增量部署, 与现有的 BGP 策略兼容, 另外, 该架构具有足够的灵活性, 可以允许不同的任播服务添加不同类型的提示, 并且 AS 可以使用自己的机制来评估提示并选择最佳路由.

文献 [53] 中展示了另一种解决方案. 他们首先提出任播网络无法直接控制入站路由: 控制入站路由很大程度上取决于上游供应商的策略. 如图 6, 任播网络可能具有少数 (左侧) 上游网络运营商, 也可能有多数 (右侧).

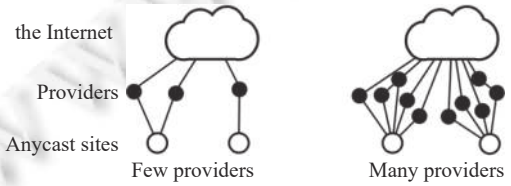


图 6 任播网络以及上游供应商[53]

© 中国科学院软件研究所 http://www.jos.org.cn

作者又进一步认证了在具有广泛而多样的对等链接的多运营商网络的情况下, 影响路由决策的正确机制是宣告本身, 通过改变接收它们的供应商的集合可以优化任播性能, 但是简单的添加或移除对运营商的宣告并不总是可以提高 RTT 性能, 实际上, 不加选择地向新的运营商宣告会导致近 40% 的网络性能下降. 所以作者发明一个名为 DaliyCatch 的方法, 这是一种用于测试和验证宣告配置更改的经验性度量方法, 该方法可以帮助站点在宣告任播 IP 时能够选择性得向部分供应商网络进行宣告, 从而使网络中的路由信息易于 BGP 做出最优选择.

文献 [23] 更细粒度的, 通过预测任播集水区挑选出任播站点子集以达到最小化客户端延迟的目的. 该文章的作者发现, 对于一个客户端网络, 当在任意两个 (可能是多个) 任播站点之间进行选择时, 将始终选择其中一个. 并且在考虑所有任播站点对偏好时, 客户端网络的成对偏好集通常可以形成一个全序. 而在已知了客户端网络对于所有站点的偏好顺序之后, 将任播优化问题映射到具有偏好排序的简单工厂选址问题, 通过解决该问题以找到实现最低总延迟的任播站点子集. 最终实验表明该方法可以达到 94.7% 的集水区预测准确率并且降低客户端平均 RTT 33 ms.

针对任播的种种问题, 有学者提出了区域任播的概念, 与普通的全球任播不同, 区域任播将全球网络划分为多个虚拟集群, 每个集群属于一个特定的地理区域. 通过限制任播服务区域来限制候选站点的数量, 以防止潜在的路径膨胀[61]. 关于区域任播的研究暂时并不多, 就我们所知, 仅在文献 [60] 中提及了如何进行区域划分, 文中考虑如下情况: 当用户与其 LDNS 划分到不同区域, 那么用户流量会被路由到另一区域的站点上, 这无疑会降低区域任播的性能. 所以该文章通过寻找用户到 LDNS 的映射, 最小化用户和他们的 LDNS 由不同区域的站点服务的可能性. 尽管区域任播相关研究不多, 但该技术已经被广泛地应用于 CDN 公司如 EdgeCast, Incapsula 等.

4.2.3 总 结

通过上述对现有解决任播性能问题方案的分析, 我们认为区域任播的方案最具可行性, 尽管前面的几种方案经过实验证明了其有效性, 在改善任播性能方面都有各自的优势. 然而, 这些方法不是对中间路由器添加额外的负担, 就是很难在广域网中部署实现, 因此至今这些方法仅在实验中验证可行. 对于区域任播的研究尚不全面, 我们列举了几个未来可能的研究方向.

(1) 区域任播的区域划分: 在之前讨论任播性能无法保证的原因时提到, 远程对等是主要原因之一. 但随着互联网扁平化趋势的逐步推进, 远程对等将是未来网络通信的重要组成部分. 因此, 如何优化区域划分方法将是“切断”远程对等, 提高区域任播性能的研究方向之一.

(2) 从全球任播向区域任播的转变: 从已经部署的全球任播转移到区域任播需要进行哪些操作, 注意哪些问题, 是否有通用的方法等.

(3) 区域的动态变化等: 动态的网络环境存在许多不可知因素, 比如一条链路的失效可能会导致单个区域内站点负载的上升. 而动态的调整区域划分可以有效平衡区域内站点负载, 提升区域任播的鲁棒性.

4.3 关于任播负载均衡的研究

首先提出有关任播负载均衡方案的是文献 [38], 该文中指出, 可以通过管理员手动修改各站点的路由公告来控制路由到各自站点的用户数量, 具体的方法是预置 AS-PATH 属性, 将超负载站点的 BGP 公告中预置多跳的 AS-PATH, 由此一些客户会因为 AS-PATH 的增加而不选择本该选择的服务站点. 这种方法实际上是任播负载均衡与拓扑邻近的权衡, 使用预置 AS-PATH 可能会使客户无法选择一个拓扑最优站点, 但避免了站点的过载情况.

Yamamoto 等人[10]提出了一种新的网络服务模式“主动任播”, 通过主动路由器从负载平衡的角度选择合适的服务器转发客户的任播请求; 紧接着, Yamamoto 等人[62]又在此基础上进行改进, 使得任播能够在负载均衡与 RTT 之间取得平衡, 即主动路由器而从负载平衡以及 RTT 的角度进行路由选择.

在 2008 年, Alzoubi 等人[63]发明了一种具有负载感知特性的任播 CDN 模型, 该模型主要基于路由控制[64,65], 如图 7.

首先图 7 中心位置存在一个路由控制器 (route controller), 在每个 AS 边缘都分布着边缘路由器 (provider edge

© 中国科学院软件研究所 http://www.jos.org.cn

router, PE), 该模型的前提是路由控制器可以与 CDN 提供商网络中的边缘路由器交换路由, 由此路由控制器可以影响这些边缘路由器的路由选择. 首先, CDN 节点 A, B 通过 BGP (分别通过 PE0 和 PE5) 宣告相同的任播地址, PE0 和 PE5 依次向路由控制器通告任播地址, 该路由控制器负责向网络中所有其他 PE (PE1 至 PE4) 通告 (适当的) 路由. 这些 PE 依次通过 eBGP 会话与邻居网络中的对等路由器 (PEa 到 PEd) 通告路由, 从而使任播地址可以到达整个互联网, 而 CDN 节点上的内容请求流量将遵循相反的路径到达站点. 基于返回给路由控制器的负载反馈 (入口 PE 负载和服务器负载), 它可以决定将哪个入口 PE (PE1 至 PE4) 定向到哪个出口 PE (PE0 或 PE5). 路由控制器可以操纵入口负载的请求通信量, 以实现平衡服务器负载. 在作者的方法中存在一个隐含的假设, 即请求 (进入) 流量与结果响应服务器流量之间存在直接关联, 尽管在直觉上这种关联成立, 但是也存在例外, 仅根据请求流量的大小估算服务器响应流量会影响整个架构的性能.

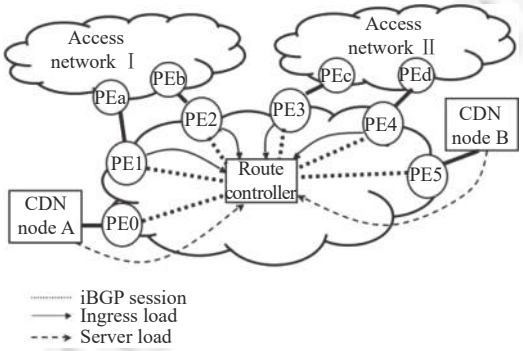


图 7 负载感知任播 CDN 模型[63]

文献 [66] 给出了另外一种名为 FastRoute 的解决方案, FastRoute 构造了一个分层体系结构, 每一层拥有各自的任播 IP, 当某一层中的节点超载时, 流量会重新定向到下一层节点 (如图 8). 其中圆柱体表示一个 FastRoute 节点, 该节点中包含任播服务站点, 由外向内节点容量逐渐增大, 当某一层的节点检测到身负载过大时会将流量转移到内层节点上.

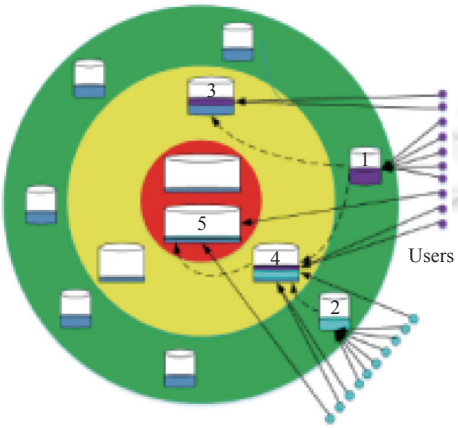


图 8 3 层任播配置图例[66]

该方案具体实现的关键在于将权威 DNS 服务器与任播站点并置于同一个 FastRoute 节点中, 并且基于假设: 可控流的数量要远大于不可控流 (客户 DNS 查询和随后的用户流量代理都位于同一 FastRoute 节点上, 该用户流量即称为可控流). 当一个 FastRoute 节点中的负载管理服务器 (load manager server) 检测到代理服务器的负载超过某一阈值时, 该 DNS 服务器会返回下一层站点的 IP 从而实现分流. 尽管该方法已被应用于微软 CDN 中并受到

© 中国科学院软件研究所 http://www.jos.org.cn

了一定成效, 然而该方法需要很强的应用条件即 DNS 服务器与任播站点并置在一起, 这使得 FastRoute 很难得到广泛应用.

另外, Cloudflare 在其边缘数据中心内部使用任播并提出了一种针对任播的负载均衡方法, 该数据中心并没有采用集中式的负载均衡器[67], 具体方法如下: 数据中心中的每一台服务器都通过 BIRD 对外宣告自己的 IP 地址(该 IP 地址都是相同的), 每个服务器都将自己的负载情况附加到所属 IP 路由, 当数据中心选择服务器时会根据每条路由上附加的信息进行选择[22]. 然而这种方法只能够在数据中心的局域网上部署实施, 在广域网中因为无法对 IP 宣告内容进行修改, 因此无法实现.

5 总结与展望

本文首先介绍了任播的工作机制以及任播的发展历史, 随后阐述了任播在当前互联网中的主要应用, 最后列举任播发展过程中存在的问题, 主要分为: 任播推断的不完善, 任播性能无法保证, 难以控制任播负载均衡 3 个方面. 总结了关于任播问题的相关工作和最新研究进展, 同时针对不同研究发明的解决方案进行总结并提出需要改进的方向. 具体来说, 已有工作研究还存在以下几个问题.

(1) 已有的任播前缀检测和任播站点定位的精度有待提高. 无论是被动测量还是主动测量, 所能达到的最高精度也只是在 90% 左右, 另外, 由于一些特殊原因, 任播地址的不响应也导致最终枚举结果不完全, 定位不准确的问题.

(2) 区域任播的区域划分太随意. 目前市面上采用的区域任播基本以大洲为单位进行区域划分, 我们认为这种划分方式过于粗糙导致无法解决全球任播中存在的问题. 对于区域任播的测量中我们发现, 尽管能够略微提升任播的性能, 但是因为各大洲的部署情况不同, 各区域的表现也参差不齐, 具体表现为欧洲北美洲站点多性能好, 非洲南美洲站点少性能差.

(3) 缺乏一种普适性的平衡任播各站点负载的方法. 文献 [63] 采用的集中式负载控制以及文献 [66] 采用的分布式控制方法都需要特定的部署条件 (集中式方法需要网络中实现路由控制机制, 分布式方法需要将权威 DNS 与站点并置), 而这些方法需要在部署任播服务前进行规划, 而对于已部署的任播, 都无法得到应用.

随着任播服务的越来越成熟, 任播的安全性也愈发重要, 因此, 部署一个安全、高效的任播服务, 使其能够适应各种各样的网络环境也变得重要. 同时, 因为任播的特性, 针对任播的研究无法绕开对网络路由的研究, 任播问题最终还是需要在路由协议的研究中寻求解决方法. 未来的研究将继续致力于提高全球范围任播的性能, 并利用任播独特的工作原理寻找新的应用服务.

References:

[1] Cicalese D, Augé J, Joumblatt D, Friedman T, Rossi D. Characterizing IPv4 anycast adoption and deployment. In: Proc. of the 11th ACM Conf. on Emerging Networking Experiments and Technologies. Heidelberg: Association for Computing Machinery, 2015. 16. [doi: 10.1145/2716281.2836101]

[2] Giordano D, Cicalese D, Finamore A, Mellia M, Munafo M, Joumblatt D, Rossi D. A first characterization of anycast traffic from passive traces. In: Proc. of the 2016 IFIP Workshop on Traffic Monitoring and Analysis (TMA). Ouvain La Neuve: IMT, 2016. 30–38.

[3] Partridge C, Mendez T, Milliken W. Host anycasting service. 1993. http://www.rfc-editor.org/rfc/rfc1546.txt

[4] Li J, Lu SW. Research and design of routing protocols for IPv6 anycast communication. Computer Systems & Applications, 2007, (9): 26–30 (in Chinese with English abstract). [doi: 10.3969/j.issn.1003-3254.2007.09.007]

[5] Zhang QL, Jiang CP, Wang JL, Li X. A survey on IPv6 address structure standardization researches. Chinese Journal of Computers, 2019, 42(6): 1384–1405 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2019.01384]

[6] Leyes Z. How anycast works to bring content closer to your visitors. 2017. https://www.imperva.com/blog/how-anycast-works/

[7] Xu L, Tang XW. Research based on IPv6 anycast technology. Computer Science, 2006, 33(S12): 19–23, 70 (in Chinese with English abstract).

[8] Calder M, Flavel A, Katz-Bassett E, Mahajan R, Padhye J. Analyzing the performance of an anycast CDN. In: Proc. of the 2015 Internet Measurement Conf. Tokyo: Association for Computing Machinery, 2015. 531–537. [doi: 10.1145/2815675.2815717]

© 中国科学院软件研究所 http://www.jos.org.cn

[9] de Oliveira Schmidt R, Heidemann J, Kuipers JH. Anycast latency: How many sites are enough? In: Proc. of the 18th Int'l Conf. on Passive and Active Measurement. Sydney: Springer, 2017. 188–200. [doi: 10.1007/978-3-319-54328-4_14]

[10] Yamamoto M, Miura H, Nishimura K. Server load balancing with network support: Active anycast. IEEE Trans. on Communications, 2001, E84-B(6): 1561–1568.

[11] Calder M, Fan X, Hu Z, Katz-Bassett E, Heidemann J, Govindan R. Mapping the expansion of Google's serving infrastructure. In: Proc. of the 2013 Conf. on Internet Measurement Conf. Barcelona: Association for Computing Machinery, 2013. 313–326. [doi: 10.1145/2504730.2504754]

[12] Anonymous. The collateral damage of Internet censorship by DNS injection. ACM SIGCOMM Computer Communication Review, 2012, 42(3): 21–27. [doi: 10.1145/2317307.2317311]

[13] Gill P, Crete-Nishihata M, Dalek J, Goldberg S, Senft A, Wiseman G. Characterizing Web censorship worldwide: Another look at the opennet initiative data. ACM Trans. on the Web, 2015, 9(1): 4. [doi: 10.1145/2700339]

[14] Grubb B. The four digits that could thwart Australia's anti-piracy, website-blocking regime. 2015. https://www.smh.com.au/technology/8888-the-four-digits-that-could-thwart-australias-antipiracy-websiteblocking-regime-20150624-ghw7kc.html

[15] Madory D, Popescu A, Zmijewski E. Accidentally Importing Censorship-The I-root instance in China. San Francisco: Renesys Corporation, 2010.

[16] Fan X, Heidemann JS, Govindan R. Evaluating anycast in the domain name system. In: Proc. of the 2013 IEEE INFOCOM. Turin: IEEE, 2013. 1681–1689. [doi: 10.1109/INFCOM.2013.6566965]

[17] Aben E. DNS root server transparency: K-Root, anycast and more. 2017. https://labs.ripe.net/author/emileaben/dns-root-server-transparency-k-root-anycast-and-more/

[18] Bellis R. Researching F-root anycast placement using RIPE Atlas. 2015. https://labs.ripe.net/author/ray_bellis/researching-f-root-anycast-placement-using-ripe-atlas/

[19] Moura GCM, de O. Schmidt R, Heidemann J, de Vries WB, Müller M, Wei L, Hesselman C. Anycast vs. DDoS: Evaluating the November 2015 root DNS event. In: Proc. of the ACM Internet Measurement Conf. Santa Monica: Association for Computing Machinery, 2016. 255–270. [doi: 10.1145/2987443.2987446]

[20] Zhang X, Sen T, Zhang ZY, April T, Chandrasekaran B, Choffnes D, Maggs BM, Shen HY, Sitaraman RK, Yang XW. AnyOpt: Predicting and optimizing IP anycast performance. In: Proc. of the 2021 ACM SIGCOMM Conf. New York: Association for Computing Machinery, 2021. 447–462. [doi: 10.1145/3452296.3472935]

[21] de Vries WB, de O. Schmidt R, Hardaker W, Heidemann J, de Boer PT, Pras A. Broad and load-aware anycast mapping with Verfploeter. In: Proc. of the Internet Measurement Conf. London: ACM, 2017. 477–488. [doi: 10.1145/3131365.3131371]

[22] Zhou DW, Ye HJ, Zhan DC. Learning placeholders for open-set recognition. In Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 4399–4408. [doi: 10.1109/CVPR46437.2021.00438]

[23] Schomp K, Al-Dalky R. Partitioning the internet using anycast catchments. ACM SIGCOMM Computer Communication Review, 2020, 50(4): 3–9. [doi: 10.1145/3431832.3431834]

[24] Wei L, Heidemann J. Does anycast hang up on you? In: Proc. of the 2017 Network Traffic Measurement and Analysis Conf. (TMA). Dublin: IEEE, 2017. 1–9. [doi: 10.23919/TMA.2017.8002905]

[25] Maheshwari R. TCP over IP anycast-pipe dream or reality? 2015. https://engineering.linkedin.com/network-performance/tcp-over-ip-anycast-pipe-dream-or-reality

[26] Levine M, Lyon B, Underwood T. TCP anycast —Don 't believe the FUD. 2008. https://archive.nanog.org/meetings/nanog37/presentations/matt.levine.pdf

[27] Engel R, Peris V, Saha D, Basturk E, Haas R. Using IP anycast for load distribution and server location. In: Proc. of the 3rd Global Internet Mini Conf. 1998. 27–35.

[28] Masafumi OE, Yamaguchi S. Design, implementation and evaluation of routing protocols for IPv6 anycast communication. In: Proc. of the 10th Annual Internet SOC. Conf. 2000.

[29] Colitti L, Romijn E, Uijterwaal H, Robachevsky A. Evaluating the effects of anycast on DNS root nameservers. 2006. https://www.ripe.net/publications/docs/ripe-393

[30] Ballani H, Francis P. Towards a global IP Anycast service. In: Proc. of the 2005 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. Philadelphia: ACM, 2005. 301–312. [doi: 10.1145/1080091.1080127]

[31] Abley J. Hierarchical Anycast for Global Service Distribution. Redwood City: ISC, 2003.

[32] Huffaker B, Fomenkov M, Plummer D, Moore D. Distance metrics in the internet. In: Proc. of the 2022 IEEE Int'l Telecommunications Symp. 2002. 1–6.

© 中国科学院软件研究所 http://www.jos.org.cn

[33] Xu X. Research on anycast routing protocol in IPv6 [Ph.D. Thesis]. Nanjing: Nanjing University of Science and Technology, 2011 (in Chinese with English abstract).

[34] Biliris A, Cranor C, Douglis F, Rabinovich M, Sibal S, Spatscheck O, Sturm W. CDN brokering. Computer Communications, 2002, 25(4): 393–402. [doi: 10.1016/S0140-3664(01)00411-X]

[35] Barbir A, Cain B, Nair R, Spatscheck O. Known Content Network (CN) RequesT-routing Mechanisms. RFC Editor, 2003. [doi: 10.17487/RFC3568]

[36] Pearce P, Ensafi R, Li F, Feamster N, Paxson V. Augur: Internet-wide detection of connectivity disruptions. In: Proc. of the 2017 IEEE Symp. on Security and Privacy (SP). San Jose: IEEE, 2017. 427–443. [doi: 10.1109/SP.2017.55]

[37] Madory D, Cook C, Miao K. Who are the anycasters? https://archive.nanog.org/sites/default/files/wed.general.cowie_.anycasters.37.pdf

[38] Johnson D, Deering S. Reserved IPv6 subnet anycast addresses. 1999. https://www.rfc-editor.org/info/rfc2526 [doi: 10.17487/RFC2526.]

[39] Gueye B, Ziviani A, Crovella M, Fdida S. Constraint-based geolocation of Internet hosts. In: Proc. of the 4th ACM SIGCOMM Conf. on Internet Measurement. Taormina: ACM, 2004. 288–293. [doi: 10.1145/1028788.1028828]

[40] Eriksson B, Crovella M. Understanding geolocation accuracy using network geometry. In: Proc. of the 2013 IEEE INFOCOM. Turin: IEEE, 2013. 75–79. [doi: 10.1109/INFCOM.2013.6566738]

[41] Ballani H, Francis P, Ratnasamy S. A measurement-based deployment proposal for ip anycast. In: Proc. of the 6th ACM SIGCOMM Conf. on Internet Measurement. Rio de Janeriro: ACM, 2006. 231–244. [doi: 10.1145/1177080.1177109]

[42] Sarat S, Pappas V, Terzis A. On the use of anycast in DNS. In: Proc. of the 15th IEEE Int'l Conf. on Computer Communications and Networks. Arlington: IEEE, 2006. 71–78. [doi: 10.1109/ICCCN.2006.286248]

[43] Liang JJ, Jiang J, Duan HX, Li K, Wu JP. Measuring query latency of top level dns servers. In: Proc. of the 14th Int'l Conf. on Passive and Active Measurement. Hong Kong: Springer, 2013. 145–154. [doi: 10.1007/978-3-642-36516-4_15]

[44] Lee BS, Tan YS, Sekiya Y, Narishige A, Date S. Availability and effectiveness of root DNS servers: A long term study. In: Proc. of the 2010 IEEE Network Operations and Management Symp. Osaka: IEEE, 2010. 862–865. [doi: 10.1109/NOMS.2010.5488355]

[45] Li ZH, Levin D, Spring N, Bhattacharjee B. Internet anycast: Performance, problems, & potential. In: Proc. of the 2018 Conf. of the ACM Special Interest Group on Data Communication. Budapest: ACM, 2018. 59–73. [doi: 10.1145/3230543.3230547]

[46] Liu ZQ, Huffaker B, Fomenkov M, Brownlee N, Claffy K. Two days in the life of the dns anycast root servers. In: Proc. of the 8th Int'l Conf. on Passive and Active Network Measurement. Berlin: Springer, 2007. 125–134. [doi: 10.1007/978-3-540-71617-4_13]

[47] Kuipers JH. Analysing the K-root anycast infrastructure. 2015. https://labs.ripe.net/Members/jh_kuipers/analyzing-the-k-root-anycast-infrastructure

[48] Koch T, Li K, Ardi C, Katz-Bassett E, Calder M, Heidemann J. Anycast in context: A tale of two systems. In: Proc. of the 2021 ACM SIGCOMM. New York: ACM, 2021. 398–417.

[49] Spring N, Mahajan R, Anderson T. The causes of path inflation. In: Proc. of the 2003 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. Karlsruhe: ACM, 2003. 113–124. [doi: 10.1145/863955.863970]

[50] Barber P, Larson M, Kosters M, Toscano P. Life and times of J-ROOT. 2004. https://archive.nanog.org/meetings/nanog32/presentations/kosters.pdf

[51] Colitti L. Effects of anycast on K-root. 2005. https://meetings.ripe.net/ripe-51/presentations/ripe51-anycast_k-root.pdf

[52] Boothe P, Bush R. Anycast measurements used to highlight routing instabilities. NANOG 34 meeting, 2005. https://www.youtube.com/watch?v=0eWIJ56s580\&ab_channel=NANOG

[53] McQuistin S, Uppu SP, Flores M. Taming anycast in the wild internet. In: Proc. of the 2019 ACM Internet Measurement Conf. Amsterdam: ACM, 2019. 165–178. [doi: 10.1145/3355369.3355573]

[54] Cicalese D, Joumblatt D, Rossi D, Buob MO, Augé J, Friedman T. A fistful of pings: Accurate and lightweight anycast enumeration and geolocation. In: Proc. of the 2015 IEEE Conf. on Computer Communications. Hong Kong: IEEE, 2015. 2776–2784. [doi: 10.1109/INFOCOM.2015.7218670]

[55] Sommese R, Bertholdo L, Akiwate G, Jonker M, Van Rijswijk-Deij R, Dainotti A, Claffy KC, Sperotto A, MAnycast^2: Using anycast to measure anycast. In: Proc. of the 2020 ACM Internet Measurement Conf. ACM, 2020. 456–463. [doi: 10.1145/3419394.3423646]

[56] Bian R, Hao S, Wang HN, Dhamdere A, Dainotti A, Cotton C. Towards passive analysis of anycast in global routing: Unintended impact of remote peering. ACM SIGCOMM Computer Communication Review, 2019, 49(3): 18–25. [doi: 10.1145/3371927.3371930]

[57] Eriksson B, Barford P, Sommers J, Nowak R. A learning-based approach for IP geolocation. In: Proc. of the 11th Int'l Conf. on Passive and Active Measurement. Zurich: Springer, 2010. 171–180. [doi: 10.1007/978-3-642-12334-4_18]

[58] Castro I, Cardona JC, Gorinsky S, Francois P. Remote peering: More peering without internet flattening. In: Proc. of the 10th ACM Int'l on Conf. on Emerging Networking Experiments and Technologies. Sydney: ACM, 2014. 185–198. [doi: 10.1145/2674005.2675013]

© 中国科学院软件研究所 http://www.jos.org.cn

[59] Nomikos G, Kotronis V, Sermpezis P, Gigis P, Manassakis L, Dietzel C, Konstantaras S, Dimitropoulos X, Giotsas V. O peer, where art thou?: Uncovering remote peering interconnections at IXPs. In: Proc. of ACM Internet Measurement Conf. Boston: ACM, 2018. 265–278. [doi: 10.1145/3278532.3278556]

[60] Calder M, Schröder M, Gao R, Stewart R, Padhye J, Mahajan R, Ananthanarayanan G, Katz-Bassett E. Odin: Microsoft's scalable fault-tolerant CDN measurement system. In: Proc. of the 15th USENIX Conf. on Networked Systems Design and Implementation. Renton: ACM, 2018. 501–517. [doi: 10.5555/3307441.3307484]

[61] Route optimization. 2019. https://www.imperva.com/learn/performance/route-optimization-anycast/

[62] Miura H, Yamamoto M. Server selection policy in active anycast. IEICE Trans. on Communications, 2001, E84-B(10): 1–4.

[63] Alzoubi HA, Lee S, Rabinovich M, Spatscheck O, Van Der Merwe J. Anycast CDNs revisited. In: Proc. of the 17th Int'l Conf. on World Wide Web. Beijing: ACM, 2008. 277–286. [doi: 10.1145/1367497.1367536]

[64] Van Der Merwe J, Cepleanu A, D'Souza K, *et al*. Dynamic connectivity management with an intelligent route service control point. In: Proc. of the 2006 SIGCOMM Workshop on Internet Network Management. Pisa: ACM, 2006. 29–34. [doi: 10.1145/1162638.1162643]

[65] Verkaik P, Pei D, Scholl T, Shaikh A, Snoeren AC, Van Der Merwe JE. Wresting control from BGP: Scalable fine-grained route control. In: Proc. of the 2007 USENIX Annual Technical Conf. on Proc. of the USENIX Annual Technical Conf. Santa Clara: ACM, 2007. 23. [doi: 10.5555/1364385.1364408]

[66] Flavel A, Mani P, Maltz DA, Holt N, Liu J, Chen YY, Surmachev O. FastRoute: A scalable load-aware anycast routing architecture for modern CDNs. In: Proc. of the 12th USENIX Conf. on Networked Systems Design and Implementation. Oakland: ACM, 2015. 391–394. [doi: 10.5555/2789770.2789797]

[67] Load balancing without load balancers. 2013. https://blog.cloudflare.com/cloudflares-architecture-eliminating-single-p/

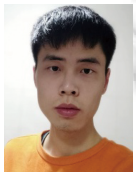
附中文参考文献:

[4] 李锦, 鲁士文. 基于IPv6的任播路由协议的研究和设计. 计算机系统应用, 2007, (9): 26–30. [doi: 10.3969/j.issn.1003-3254.2007.09.007]

[5] 张千里, 姜彩萍, 王继龙, 李星. IPv6地址结构标准化研究综述. 计算机学报, 2019, 42(6): 1384–1405. [doi: 10.11897/SP.J.1016.2019.01384]

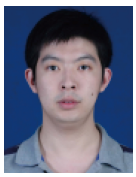
[7] 许靓, 唐学文. 基于IPv6任播技术的研究. 计算机科学, 2006, 33(S12): 19–23, 70.

[33] 徐昕. IPv6中任播路由协议的研究 [博士学位论文]. 南京: 南京理工大学, 2011.



周敏苑(1997－), 男, 博士生, 主要研究领域为任播协议, 内容分发网络.

窦万春(1971－), 男, 博士, 教授, CCF 高级会员, 主要研究领域为大数据, 云计算, 边缘计算, 群智计算.



郑嘉琦(1986－), 男, 博士, 副研究员, CCF 专业会员, 主要研究领域为网络协议优化, 新型网络体系结构, 高性能数据结构, 在线优化.

陈贵海(1963－), 男, 博士, 教授, CCF 会士, 主要研究领域为未来网络系统与协议, 无线网络结构与优化, 物联网与传感网, 新型计算机体系结构, 数据中心核心技术, 数据分析与处理.

© 中国科学院软件研究所 http://www.jos.org.cn