第 3 章　IPv6 过渡基本原理

IPv4 地址短缺已成为全球性问题，而 IP 地址是下一代互联网发展的基础资源。为在下一代互联网发展中抢占先机，国内外互联网运营商在 IPv6 网络的建设部署、IPv6 服务应用拓展等方面的投入在不断加大。IPv6 基础网络建设正如火如荼地进行，而 IPv4 与 IPv6 网络不兼容的问题也日益受到国内外运营商、设备制造商的关注。IP 网络由 IPv4 向 IPv6 过渡技术的研究成为推进 IPv6 部署、发展下一代互联网的关键之一。本章将深入分析互联网由 IPv4 向 IPv6 过渡面临的基本问题，探讨 IPv6 过渡过程中涉及的关键要素，对当前的三类主流 IPv6 过渡技术——双栈技术、翻译技术和隧道技术进行原理以及优缺点剖析，从而对 IPv6 过渡有基本的理解。

3.1　IPv6 过渡概述

随着全球 IPv4 互联网普及率的大幅度提升，以及移动互联网的快速发展，IPv4 终端数量爆炸性增长导致 IPv4 地址空间的消耗日益加快。目前全球公有 IPv4 地址已经分配完毕，IPv4 地址资源正式宣告枯竭。为解决 IP 地址短缺的问题，互联网各方参与者越来越多地关注 IPv6。IPv6 具有巨大的地址空间，可为互联网提供 2^{128} 个 IP 地址，是目前最好的 IPv4 的替代协议。但是 IPv6 与 IPv4 并不兼容，为两个异构的 IP 网络。若通信两端处在不同协议的网络中，二者之间无法进行通信。然而互联网用户、互联网服务提供商（Internet Service Provider，ISP）、互联网内容提供商（Internet Content Provider，ICP）以及网络设备提供商等网络参与者由 IPv4 向 IPv6 的迁移在短时间内无法完成，因此 IPv6 完全取代 IPv4 需要相当长的时间。IPv4 与 IPv6 网络、服务及应用将长期共存，但互联网整体需逐步向 IPv6 过渡[1]。我们称互联网的这个时期为 IPv6 过渡时期。

互联网 IPv6 过渡时期的主要任务是，通过新型过渡技术沟通异构的 IPv4 网络与 IPv6 网络，从而在保护现有 IPv4 网络投资的同时，充分利用 IPv6 网络设施，促进全网向 IPv6 迁移，并最终实现全网的 IPv6 升级。IPv6 过渡技术的发展，将促使互联网获得更大的发展空间和更快的发展速度，并为移动互联网的发展开拓空间。然而 IPv6 过渡涉及互联网发展的各个部分，它们之间紧密的依存关系，致使全向 IPv6 过渡并非易事。

3.1.1　IPv6 过渡的相互依存关系

过去 30 年 IPv4 互联网的巨大繁荣，为用户提供了大量的服务和应用。在 IPv6 过渡时期，用户对丰富的 IPv4 服务仍会保持较强的依赖性，而随着 IPv6 服务类型的日益多样化，用户对 IPv6 的需求会逐渐增长。因此过渡时期需要确保用户的 IPv4 和 IPv6 使用均不受影响。

作为互联网的主要参与者，互联网服务提供商（ISP）、互联网内容提供商（ICP）和终端用户之间的相互依赖和关联，构成了当前 IP 网络发展的主体生态系统：ISP 主要向 ICP 和用户提供网络接入和传输服务，ICP 向用户提供丰富的服务以及各种创新型应用，用户通过

购买 ISP 和 ICP 提供的服务享受互联网带来的便利。IPv6 过渡是对互联网的框架性演进，涉及互联网各参与者设备和服务的更新升级。由于各方向 IPv6 过渡的代价不同，以及互联网服务提供商、互联网内容提供商以及用户等之间的相互依存关系，IPv6 过渡面临多个方面的压力。

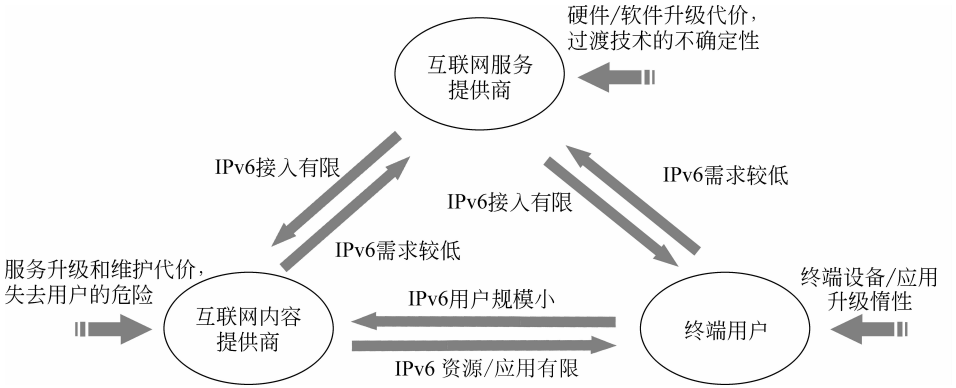

图 3.1　IPv6 过渡参与者的依存关系

硬件设备由 IPv4 向 IPv6 的升级，以及新 IPv6 接入网、主干网等网络的建设，对于运营商来说是巨大的投入。而相应的软件支撑系统系统，如认证系统、计费系统以及防火墙等，也需要与硬件系统的升级相匹配，这也是很大的成本投入。另一方面，目前过渡场景多种多样[2]，网络未来的发展趋势也难以预测，运营商无法确定在 IPv6 部署过程中会遇到何种场景；同时，针对不同场景的 IPv6 过渡技术层出不穷，尚未出现可以令运营商一劳永逸的 IPv6 过渡技术。IPv6 过渡场景的多变性和过渡技术的多样性导致运营商过渡策略难以短期唯一确定，这也变相地增加了运营商向 IPv6 过渡的成本。运营商向 IPv6 过渡的动力不足、IPv6 网络建设和部署的局限，致使运营商向内容提供商、终端用户提供 IPv6 的接入范围有限。

新浪、腾讯、百度等内容提供商，也需要向 IPv6 迁移。内容提供商实现向 IPv6 过渡则需要涉及原有 IPv4 服务的全面、平滑升级，以及数据中心网络设施的 IPv6 升级。内容提供商还需开拓、推广新 IPv6 业务，在保持当前用户群的基础上吸引新用户。而在其数据中心、内部网络中新增或升级网络设备的巨大投入，拓展新业务带来的不确定性，新型 IPv6 服务相对于已存在的 IPv4 服务的滞后性，都有可能导致用户暂时性甚至永久性的流失。因此内容提供商 ICP 对于数据中心、内部网络的 IPv6 升级改造以及对 IPv6 服务的研发投入和业务拓展处理较为谨慎。

终端用户更关注互联网中各种应用服务的实际体验，而对于底层传输为 IPv4 还是 IPv6 并没有特别的偏好。但由于内容提供商提供的 IPv6 服务种类有限，而 IPv4 新型服务层出不穷、用户体验良好，导致用户对 IPv4 服务具有较强的依赖性。在过去十年 IPv6 的发展历程中，并未出现所谓“杀手级”应用对用户产生巨大的吸引力，促使用户主动向 IPv6 迁移。另外，用户侧的 IPv6 过渡会需要用户升级设备的协议栈、升级客户端软件，甚至更换接入设备(如家庭网关)，导致用户侧向 IPv6 的过渡缓慢。用户侧的 IPv6 需求不强烈，也成为阻碍运营商和服务提供商 IPv6 过渡的原因之一。

3.1.2 IPv6 平滑过渡的原则

尽管全网向 IPv6 的过渡面临诸多阻力，但全球互联网用户日益增长，服务商服务器集群效应越来越明显，对 IP 地址的需求量爆炸性增长已耗尽 IPv4 地址空间。互联网顺利过渡到 IPv6，是其进一步高速、稳定的发展的基础。IPv6 过渡符合下一代互联网的发展方向，是当今国内外网络技术研究的重点和热点。以下几方面考虑需要贯穿 IPv6 过渡的全过程，以确保互联网向 IPv6 平滑过渡，并促进 IPv6 未来的发展。

1. 提供 IPv4 和 IPv6 服务

在过渡时期，随着运营商网络设备的逐步升级、服务商新型 IPv6 业务的提供以及用户侧操作系统、设备对 IPv6 支持的逐步普及，同时支持 IPv4 和 IPv6 的双栈终端将逐步增多，但 IPv4 单栈用户将仍然存在。因此 IPv4 与 IPv6 用户在过渡时期会同时存在，处于不同网络环境中的用户都会有访问各类 IPv4 和 IPv6 应用服务的需求，为确保用户体验不受影响，运营商的单栈和双栈网络均需要具有同时具备提供 IPv4 和 IPv6 的接入和传输服务的能力。

2. 确保可持续发展

运营商在选择 IPv6 过渡技术方案时，需要从可持续性的角度出发，所部署的方案不能对现有网络产生限制，也不能对 IPv6 未来的发展产生潜在的限制甚至阻碍 IPv6 的进一步发展。IPv6 过渡场景多种多样，各种场景下所对应的过渡需求也不尽相同，但其目标都应是促使网络最终演进为 IPv6 单栈网络。因此运营商所部署的 IPv6 过渡方案既要满足过渡场景的需求，又要促进原生 IPv6 的部署和使用，从而促使全网逐步向 IPv6 切换。

3. 易于运营和管理

较为简易的网络运维管理有利于过渡技术方案的部署和推广。因此过渡技术需易于理解、便于维护，从而使得网络运维人员能迅速掌握技术核心，便于网络监控、故障检测排除等。IPv6 过渡方案也需要具有较强的健壮性（例如支持冗余备份等），以确保营商在实际运营管理时，可以有效减轻配置、维护和排障的负担。

4. 保证处理效率

选择和部署过渡技术时必须有性能考虑，过渡技术方案要从设计角度保证应用数据传输、路由转发的效率，保证用户体验，避免因为技术方案漏洞造成网络拥塞。过渡设备也应能在较低投入的基础上保证数据的处理效率。

5. 逐步部署和最小更新

过渡机制的部署可能会发生在运营商网络中的各个位置，且不同网络中的设备也可能同时部署不同的过渡机制，因此要求过渡机制可在部分部署的情况下运行，且不会对其他部分产生不利的影响。另外，部署的过渡机制应该可以促进 IPv6 网络和用户的规模逐步增加。过渡机制对网络设备的更新应该尽量简单，降低对现有网络的影响，避免大规模升级；从用户体验的角度考虑，用户端也应该避免复杂的更新。

3.2 IPv6 过渡关键技术要素

IPv6 过渡涉及互联网的各个部分，涉及 IPv6 过渡方案时需要从多个维度进行考虑，以保证过渡方案能符合上述原则。这些“维度”就是 IPv6 过渡技术设计时需要关注的各个关

键技术要素，主要包括端到端透明性、编址及地址规划、路由可扩展性、状态维护以及 IPv4 地址资源复用等。这些关键要素可以作为运营商判断 IPv6 过渡机制适用性的依据。

3.2.1 端到端透明性

端到端特性是计算机网络的经典设计原则[3]。在网络中，如果要进行通信，那么通信的两端就需要通过某种方式连接起来。这种连接从物理层看来，是以各种网络设备，如路由器、交换机等，通过传输线缆的连接形成的。而端到端通信的两端建立连接时并不关心它们之间物理链路的情况，因此，当两端建立连接后就认为它们之间是端到端传输了，直到通信结束这种状态会一直保持。用于建立端到端连接的典型协议如传输控制协议（Transport Control Protocol，TCP）[4]和顺序包交换协议（Sequences Packet Exchange，SPX）[5]。

端到端特性要求针对应用程序的功能应该实现在终端主机上而不是网络中的传输节点。链路建立后，发送端知道接收设备一定能收到数据，而且经过中间交换设备时不需要进行存储，因此传输延迟小。对于两个相互通信的节点，从所采取的网络侧通信方式获得的可靠性可能无法与两个通信节点需要的可靠性完全匹配。在终端主机使用某些机制以确保通信可靠性，要比在传输网络采取措施更容易且便于管理和操作，尤其是通常终端无法对传输网络进行控制。端到端特性使得网络核心设备主要负责传输功能，而在端系统实现各种创新型应用。

端到端透明性是指在互联网 IP 协议簇（TCP/IP 模型）的设计中，将互联网系统中与通信相关的部分（IP 网络）与高层应用（端点）分离，最大限度地简化网络的设计，将尽可能多的复杂性和控制放在用户终端上。亦即应尽可能地将与特定应用相关的状态信息维护在端系统处，而网络内部不维护该类状态信息，从而简化网络内部的功能而使其专注于网络传输。这样才能在网络中某部分发生故障时，不至于通信中断，除非端系统发生故障。

在 IPv4 网络中，由于 IPv4 地址短缺，网络地址翻译技术（Network Address Translation，NAT）大量使用，终端用户使用的私有 IPv4 地址需要在网关处翻译成公有 IPv4 地址才能顺利访问 IPv4 互联网资源。网络地址翻译技术通过位于网络高层的网关设备隐藏了发起连接的真正终端主机，导致逆向地址溯源困难。某些新型服务和应用在 NAT 环境下无法建立双向连接，以致无法正常工作，如 P2P 应用（Peer-to-Peer）、涉及网络定位的服务等。对端到端透明性的破坏致使端系统服务创新受到限制。

IPv6 具有巨大的地址空间，可以保证网络的端到端特性。而在 IPv6 过渡的过程中，也需要考虑如何保持端到端透明性，从而在保证对现有上层应用的透明支持，并催生更多在端系统的创新，吸引用户向 IPv6 迁移，进而推进全网向 IPv6 过渡。

3.2.2 编址及地址规划

在 IPv6 过渡时期，IPv4 与 IPv6 网络服务及用户将长期共存，对 IPv4 地址及 IPv6 地址均保持相应的使用需求。因此运营商需要根据所管理的 IP 地址资源情况、网络部署情况以及用户需求，对 IPv4 地址、IPv6 地址进行宏观规划。

IPv4 地址与 IPv6 地址长度不同，二者并不兼容，无法直接通信。为实现 IPv4 节点与 IPv6 节点的互访问，一种方案是将 IPv4 地址信息无失真地嵌入 IPv6 前缀或者接口 ID（Interface ID）中，构造特定格式的 IPv6 地址[6,7]，即对 IPv4 与 IPv6 地址进行紧耦合。

通信双方通过某种协议约定 IPv4 地址信息在 IPv6 地址中的位置，以对这种特定格式的 IPv6 地址进行识别和解析。内嵌 IPv4 地址信息构成 IPv6 地址的方式，要求在 IPv4 地址与 IPv6 地址之间建立稳定的映射关系，以便通信双方能够从 IPv6 地址中获取 IPv4 地址信息。因此这种 IPv6 地址结构要求运营商对 IPv4 地址与 IPv6 地址同时规划。

IPv4 地址与 IPv6 地址耦合对运营商的网络规划和运维提出一定的要求，运营商在进行 IPv6 地址规划过程中，必须同时考虑用于与 IPv6 地址空间耦合的 IPv4 地址空间的规划和耦合操作。而两种地址的分配过程也需要进行耦合，包括地址资源管理设备（如 DHCP 服务器等）。另外，将 IPv4 地址信息嵌入 IPv6 地址中，会占用 IPv6 地址中的比特位，而运营商可能使用 IPv6 地址中的某些比特位以区分业务。这导致 IPv4 地址的嵌入可能与运营商的实际运营需求产生冲突，因此 IPv4-IPv6 地址耦合并非万全之策。

另一种方案是保持 IPv4 地址与 IPv6 地址独立，而非将两种异构地址耦合。这种方案简单易行，运营商可以分别对 IPv4 地址空间和 IPv6 地址空间进行规划，以及实现地址的动态分配，从而简化运营商网络规划的复杂度，也降低 IPv6 过渡对运营商的运维管理的要求。由于 IPv4 地址与 IPv6 地址相对独立，为了实现跨异构网络的正确寻址，某些特定网络节点需要维护 IPv4 地址与 IPv6 地址的动态或静态映射。

3.2.3 路由可扩展性

互联网的蓬勃发展促使网络规模迅速膨胀，出现了多宿主（Multihoming）的场景、域间流量工程（Traffic Engineering）等应用需求，也导致域间路由策略日益复杂，全球核心路由器的路由表规模急剧膨胀。现有路由结构正面临着巨大的挑战。据统计[8]，截至 2013 年 9 月，全球主干网核心 BGP 路由条目已经超过 49 万条。这种爆炸式的增长给网络的运营维护带来了沉重负担。提高互联网路由系统的效率和降低路由器制造成本成为互联网技术的研究重点之一。而 IPv4 地址消耗殆尽，下一代互联网将采用 IPv6 作为核心协议。IPv6 具备足够的地址空间，能够满足未来网络的海量地址需求。在这种趋势下，如果继续沿用现行的路由机制，IPv4 和 IPv6 路由的共存则势必导致路由表极度膨胀和路由更新大幅增加，最终可能成为下一代互联网顺利演进的主要障碍之一。

路由表膨胀导致全球 BGP 路由更新持续增加，大量路由更新消耗了大量的网络带宽和路由计算资源。网络监测数据显示，BGP 路由表尺寸和更新报文规模保持加速增长的总体趋势。随着网络规模的扩大，BGP 路由表维护成本将持续增加。近年来 IPv4 互联网发展迅速，但是 IPv4 地址空间却继续被碎片化，导致主干网核心路由器路由信息表（Routing Information Base，RIB）和转发信息表（Forwarding Information Base，FIB）条目成指数级增长。巨大的路由表规模，导致路由查询效率降低，而且对核心路由器的硬件提出了较高要求，同时增加了运营商网络日常网络运行维护的开销。

路由表膨胀的成因是多方面的，其中得到业界比较认同主要有 3 个方面[9]。一是多数企业用户选择（Provider Independent，PI）地址，这样做可以保证在更换运营商时避免更换网络地址，从而保证系统管理员不必重新配置防火墙等 IP 地址相关的设备。但这一策略导致运营商必须单独宣告客户使用的 PI 地址，以致在网络中形成不可聚合的路由前缀。PI 地址的使用在一定程度上影响了 IPv4 地址的聚合性。二是网络中的多宿主连接。多宿主提高了连接的可靠性，还带来了优化接入成本、增加网络可靠性和流量均衡等好处。但这种

连接方式仍然需要运营商在 BGP 中宣告不可聚合的客户侧地址，同样破坏了路由地址的可聚合性。三是域内的流量工程，由于路由采用最长前缀匹配的基本策略，为了实现自治域内的流量工程，影响自治域间引入流量的路径，往往需要向外广播掩码更长的子前缀，这种方式明显地会导致路由前缀碎片的进一步增多，不利于路由聚合。按照目前的路由机制，域间前缀策略表达必须以牺牲路由可聚合性为代价，而且这一影响会扩散到全局 BGP 的范围，最终导致了互联网路由严重的可扩展性问题。

未来互联网的路由规模决定于网络应用的泛在性，特别是物联网应用的极大需求将极大地推进网络规模的扩张。在这样的应用背景下，路由系统要应对更大的地址空间，必须保证核心路由系统的路由和转发效率，并减少对带宽的消耗。路由系统的可扩展性是网络体系结构的基本要求之一。

IPv4 互联网面临着严重的路由可扩展性问题，在以 IPv6 为核心的下一代互联网的发展中，通过研究新型路由体系结构、改进路由机制，是解决问题的根本途径。IPv6 主干网核心路由聚合程度较高，路由表规模比 IPv4 小得多，路由可扩展性较高。而在互联网由 IPv4 向 IPv6 过渡的阶段，新型路由体系研究尚处在方案探索和理论研究阶段，无法彻底解决路由爆炸问题。为避免对 IPv6 未来的发展产生负面的限制性影响，在 IPv6 过渡阶段应保持 IPv6 路由系统与 IPv4 路由系统的独立，避免将 IPv4 庞大的路由表引入 IPv6 网络中，以致引发 IPv6 路由表膨胀的问题。这就要求 IPv6 过渡机制在设计过程中，保证 IPv6 网络只处理纯 IPv6 相关的路由，保证 IPv4 与 IPv6 路由隔离，在 IPv6 核心骨干网中，尽量减少与 IPv4 路由信息相关的 IPv4-IPv6 转换表项或 IPv4-IPv6 映射表项，从而保证路由可扩展性。

3.2.4 状态维护

在 IP 网络中，IP 地址标识了网络服务资源、网络用户等通信参与者的网络位置，也是区分网络参与者的标识。通过对目标 IP 地址的访问操作，就可以实现对目的网络元素的访问，如获取文本、视频资源，与网络用户进行实时通信等。因此 IP 地址即为标识网络节点的 ID。

异构网络中需要使用不同的网络 ID，以进行网络间通信的区分，以及各自特定功能的实现。由于网络 ID 不具有通用性，因此当异构的网络需要进行双向通信时，处于网络中的通信双方无法直接通信，需要在不同网络 ID 之间动态或静态建立映射关系，即维护网络 ID 之间的状态。例如在 IPv4 网络中，由于公有地址短缺引入的私有 IPv4 地址即为另一种网络 ID。两种地址分别作为公有 IPv4 网络和私有 IPv4 网络的标识，无法直接进行通信。NAT 技术对两种网络 ID 进行状态映射和维护，将私有 IPv4 地址映射为公有 IPv4 地址，从而实现了私有网络与公有网络之间的正常通信。

IPv4 网络与 IPv6 网络作为异构网络，分别使用 IPv4 地址和 IPv6 地址作为网络标识 ID。在 IPv6 过渡时期，为保证用户能获取 IPv4 以及 IPv6 互联网中的丰富服务，运营商网络需要支持 IPv4 与 IPv6 的互访问。这种服务需求，要求过渡技术维护两种网络 ID 之间的状态映射，即在 IPv6 网络中表示 IPv4 地址以及在 IPv4 网络中表示 IPv6 地址。通过对状态的维护，可以确保特定网络节点（如边界网关等）正确查找到异构网络中的对应的网络 ID，从而实现跨异构网络通信。

由于映射对象的类型可能不同，状态维护的粒度和复杂度也会不同，因此状态可分为不同种类。根据状态维护的粒度，可将状态维护分为以下四类。

(1) 每流映射状态。例如针对每个 IPv4 数据流，维护由 IPv4 流到 IPv6 流的映射关系。这种状态维护方式与 NAT 技术的机理类似，来自相同 IPv4 地址的数据流可能被映射为不同 IPv6 数据流，可动态变化，而网络设备记录了这种映射关系。

(2) 每用户状态。例如针对某个 IPv4 用户发起的所有网络连接，维护由该 IPv4 地址到某 IPv6 地址的状态映射关系，这种映射关系可以动态变化，也可以预先设定。

(3) 每前缀状态。例如针对来自某个 IPv6 子网的数据包(相同 IPv6 前缀)，全部映射为使用某一个 IPv4 地址为源地址的数据包。

(4) 网络侧无状态。在网络侧通过预先确定映射关系，使得网络侧设备自动实现 IPv4 与 IPv6 数据包的转换，而不需要维护显性状态，此时网络侧为无状态。在这种情况下，状态维护变为由 IPv4 网络与 IPv6 网络之间特定的映射关系实现，而状态维护的复杂度由网络侧完全下放至用户侧。

不同的网络环境和过渡场景，对状态维护的要求不同；运营商根据自身网络状况的评估，对可承受的状态维护的复杂度也不同。因此从状态维护的角度，会产生多种 IPv6 过渡技术方案供运营商评估和选择。

3.2.5 IPv4 地址资源复用

在 IPv6 过渡时期，须在 IPv4 网络与 IPv6 网络共存甚至重叠部署的情况下，保证 IPv4 服务的连续性。因此用户侧仍需具有合法 IPv4 地址，以支撑上层纯 IPv4 业务、应用的运行。随着全球 IPv4 地址的耗竭，为用户侧分配新的 IPv4 地址已越来越困难。充分利用有限的 IPv4 地址资源，在保证服务质量和用户体验的前提下实现对现有 IPv4 地址的复用，也是在 IPv6 过渡时期需要解决的重要难题。

在传统 IPv4 网络中，地址复用通常是通过使用网络地址翻译技术 NAT 实现的。通过边界网关连接私有 IPv4 网络和公有 IPv4 网络，用户侧使用私有 IPv4 地址，通过边界网关可以将多个私有 IPv4 地址翻译为同一个公有 IPv4 地址，从而实现地址复用。而不同的私有网络可以使用相同的私有地址，只需边界网关获取一个或多个公有地址。

在 IPv6 过渡时期，可以对网络地址翻译技术进行扩展，用户侧使用私有 IPv4 地址、网关处维护公有地址与 IPv6 地址的映射，以 NAT 的方式实现 IPv4 地址的复用。这种方式 IPv4 地址复用率高，可以最大限度地利用 IPv4 地址资源；但是复杂度和对网关设备的性能要求较高。另一种方式是将传输层端口作为 IPv4 地址资源的一部分，将一个 IPv4 地址对应的 65 536 个端口分为不重叠的“端口段”，网络侧为每个用户分配一个 IPv4 地址和可用传输层端口段。用户发起 IPv4 连接时，所使用的源地址为获取的 IPv4 地址，源端口必须为所获取的可用端口段中的端口号。这样可以使得同一个 IPv4 地址供多个用户使用，从而实现 IPv4 地址的复用。运营商可根据网络用户规模划定 IPv4 地址复用率，从而确定端口段的大小。但这种地址复用方式不能用在 IPv4 单栈网络中，否则会造成网络连接建立失败。

国际各大运营商、设备商等在进行过渡技术研究的过程中，都对上述关键要素进行了考虑与分析，以便能在尽量少的影响当前互联网发展的基础上，推动网络向 IPv6 迁移。根据实现上述关键要素的方式的不同，目前主流过渡技术可分为双栈技术、翻译技术和隧道

技术。

3.3 双栈技术

IPv4 协议栈与 IPv6 协议栈是功能相近的网络层协议，二者作为 TCP/IP 协议簇框架的重要组成部分，基于相同的数据链路层和物理层平台，其上层的传输层所使用的传输层协议也没有区别。因此可以通过技术升级，使得网络节点同时支持网络层 IPv4 和 IPv6。

3.3.1 IPv4 单栈和 IPv6 单栈

在介绍双栈技术之前，先对 IPv4 单栈和 IPv6 单栈进行定义[10]。

IPv4 单栈是指某个系统支持 IPv4 的系统协议栈，在此系统上运行的应用直接或者间接地使用 IPv4 协议栈进行通信，并通过网络中的 IPv4 路由系统进行路由转发。两个应用实体可以通过 IPv4 进行通信，而不是 IPv6。两个实体具有 IPv4 协议栈、IPv4 地址，所连接的网络支持 IPv4 路由，并可能支持 IPv4/IPv4 翻译，但是由于功能元素缺失导致其不支持与 IPv6 实体进行通信。

IPv6 单栈是指某个系统支持 IPv6 协议栈，实体之间直接或间接使用 IPv6 进行通信，而非 IPv4。两个实体具有 IPv6 协议栈、IPv6 地址，所连接的网络支持 IPv6 路由，但是由于某些功能元素缺失，导致其不支持与 IPv4 实体进行通信。

3.3.2 双栈技术原理

双协议栈技术是指 IP 网络中的节点同时支持 IPv4 和 IPv6 两种协议栈，两种协议栈之间没有互操作或相互影响。这种 IPv4/IPv6 节点具有收发 IPv4 报文和 IPv6 报文的能力。这种节点可以直接通过 IPv4 连接与 IPv4 节点进行通信，也可以直接通过 IPv6 连接与 IPv6 节点进行通信[11]。网络中的终端设备、路由器、三层交换机等具有网络层功能的设备，都可以升级为能够同时处理 IPv4 报文和 IPv6 报文的设备，从而可以分别使用原生 IPv4、IPv6 协议栈与 IPv4 节点、IPv6 节点建立通信。

双栈节点需通过现有的网络配置协议从运营商处获取相关配置资源，以便接入互联网，其中最重要的是 IP 地址配置。双栈节点可以通过静态配置获取 IPv4/IPv6 地址，也可以分别使用 IPv4 地址分配机制获取 IPv4 地址，例如 IPv4 动态主机配置协议(DHCPv4)[12]，以及通过 IPv6 地址分配机制获取 IPv6 地址，例如无状态地址自动配置机制(SLAAC)[13]或者 IPv6 动态主机配置协议(DHCPv6)[14]。

在 IPv4 和 IPv6 网络中，都有域名解析系统 DNS 来将 IP 地址与主机名进行映射[11]。在 IPv4 网络中，使用 A 记录携带 IPv4 地址，在 IPv6 网络中使用 AAAA 记录携带 IPv6 地址(详见第 2.9.2 节)。由于双栈节点需要与其他 IPv4/IPv6 节点通信，因此双栈节点的域名解析库应能够处理 A 记录和 AAAA 记录。但是 DNS 系统返回 A 记录或 AAAA 记录，与 DNS 请求报文是由 IPv4 承载还是 IPv6 承载无关，而且 DNS 系统也并不限制发送 DNS 请求的节点对 IPv4 和 IPv6 的支持情况。

双栈技术是实现向 IPv6 过渡的最简单、最直接策略。对现有的 IPv4 网络进行软硬件升级，使其能够支持 IPv6 传输，从而实现同一网络对基于 IPv4 和 IPv6 的应用同时提供服

务，而内容提供商可以继续提供 IPv4 内容服务，同时开拓 IPv6 业务。这种过渡策略适用于大部分网络，如家庭网、企业网、服务提供商网络和内容提供商网络。

双栈技术能够支持各式各样的设备和通信方式，并能良好地保证网络的端到端特性。处于网络中的终端设备只需要具备最基本的功能和最简单的配置，即可实现过渡。同时，采用原生连接可以有效地避免在传输中遇到的 MTU 配置等问题。因此，这样的网络具有稳定可靠的优势。

3.3.3 公网双栈和私网双栈

在实际网络运营中，双栈技术又分为公网双栈技术和私网双栈技术。

所谓公网双栈技术，即运营商将网络设备升级为双栈后，分配给网络设备以及终端用户的 IPv4 地址仍为全球可路由的公有 IPv4 地址。使用双栈技术后，用户体验与升级前没有差别，所有用户仍可以运行各种新型应用，如 P2P(Peer-to-Peer)应用、作为应用服务器对外提供服务等。在此基础上，用户还可以使用 IPv6 的新型应用和服务。然而公网双栈技术并未对公网地址短缺的现状有所缓解，反而加剧了公网地址的消耗。

为应对公网地址短缺的现状，私网双栈技术(见图 3.2)受到部分运营商的青睐。私网双栈技术是指终端用户和接入网路由器均使用私有 IPv4 地址，在运营商网络的公网出口处放置运营商级别的 NAT 设备 CGN(Carrier-Grade NAT)，对私有 IPv4 地址和公有 IPv4 地址进行双向转换。私网双栈技术可以缓解 IPv4 地址短缺的压力，但是由于 CGN 地址池可用的公网地址数有限，以及 CGN 设备性能的限制，用户体验会有所下降。

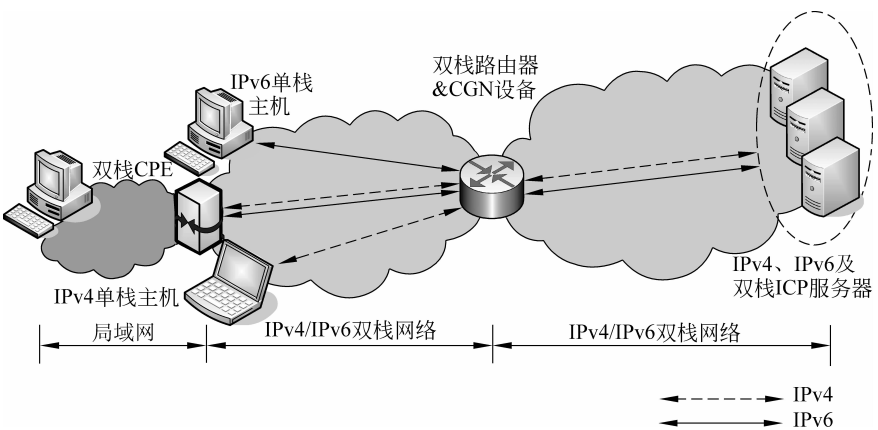

图 3.2 私网双栈原理图

3.3.4 双栈技术面临的挑战

虽然双栈技术简单直接，但是在 IPv6 过渡过程中实际网络的运营也面临着一些挑战[15]。首先，由于目前并未实现全网的 IPv6 部署，尤其是互联网内容提供商未完成双栈部署，导致网络中 IPv6 流量较小。网络连接的两端节点都应该具有 IPv6 全球可达性，拥有自己的 IPv6 地址，并将 IPv6 地址通告相应的域名服务器。由于网络中可能存在较多 IPv4 单栈目的节点(如目前的大部分网站只支持 IPv4)，虽然网络中的 IPv6 传输服务已经开启，但是流量集中在 IPv4 域内，实际的 IPv6 流量很少，难以达到向 IPv6 过渡的效果。比较直接

的解决方案是对网络目的节点进行 IPv6 升级，保证大部分节点都能够接收 IPv6 数据。目前互联网的流量高度集中在一些特定的内容提供商网络中，即使是对其中的一小部分网络进行双栈升级，也能够带来非常明显的效果。例如 Google 由只提供 IPv4 服务升级为提供 IPv4/IPv6 双栈服务后，全球 IPv6 流量大幅攀升。

双栈技术所面临的另一个挑战是，当目的 IPv4 和 IPv6 地址中有一个不可达时，某些应用可能无法迅速完成连接切换。例如，如果 IPv6 连接是不可达的，则应用程序需要花费较长时间确认 IPv6 连接失败，并重新发起 IPv4 连接。这将导致内容提供商尽量避免向 DNS 服务器宣告服务器的 IPv6 地址。缺少支持 IPv6 的内容提供商的接入，会加剧上文提到的 IPv6 流量不足的压力。

3.3.5 双栈技术小结

双栈技术的初衷是在 IPv4 地址耗尽之前部署同时支持两种 IP 的网络。然而，目前互联网发展的形势是在 IPv6 尚未实现全球性的大规模使用时，IPv4 地址已经迅速消耗完毕。双栈技术要求全网升级为 IPv4/IPv6 双栈，需要对接入网，甚至主干网的网络设备进行升级改造，势必会要求大量资源投入。由于公有地址的耗竭，运营商能支持的新增用户有限，公网双栈技术无法大规模部署。私网双栈虽然可以在一定程度上缓解运营商压力，但是私有地址的大量使用破坏了端到端特性，导致有些应用需要某些辅助技术才能正常运行，如应用层网关技术等。另外，CGN 设备的性能会成为网络服务质量的瓶颈，日志和溯源的困难导致私网双栈也无法大规模部署。

如前文所述，从 IPv6 过渡的角度考虑，双栈技术难以有效推动网络和用户向 IPv6 过渡。虽然网络设备和终端用户都支持双栈，但是用户的网络访问需求仍将停留在 IPv4，互联网内容提供商会针对用户需求，提供更丰富更有吸引力的 IPv4 服务，导致运营商耗费巨大投资建设的 IPv6 网络流量小、利用率低。

3.4 翻 译 技 术

实现 IPv4 与 IPv6 两种异构网络的直接通信，较为直观的方式是将通信发起端所使用的 IP 报文，直接转换为目标网络可识别的另一种结构的 IP 报文。IPv4/IPv6 翻译技术，通过特定算法对 IPv4 和 IPv6 报文的 IP 报头进行转换，经 IPv4/IPv6 翻译器完成 IPv4 和 IPv6 报文之间的语义翻译，从而实现两种异构网络的直接双向通信[10]。

3.4.1 翻译技术原理

经过近几年的发展，翻译技术已经基本形成了较为完整的体系。这个体系由多种翻译技术组成，并已经以国际标准的形式发布。本节从整体介绍翻译技术的核心技术。为更通用地解释 IPv4-IPv6 翻译，本节使用 IPvX 表示 IPv4 与 IPv6 网络中的一种，用 IPvY 表示另一种。

翻译技术的基本思路是对 IPvX 网络和 IPvY 网络进行语义转换。过渡网络主要由通信发起点、通信接收点和 IPv4/IPv6 翻译器构成。通信发起点和接收点分别位于 IPvX 网络和 IPvY 网络中，IPv4/IPv6 翻译器位于网络边界处，通常为地址簇边缘路由器（Address

Family Border Router，AFBR)。翻译技术原理图如图 3.3 所示。

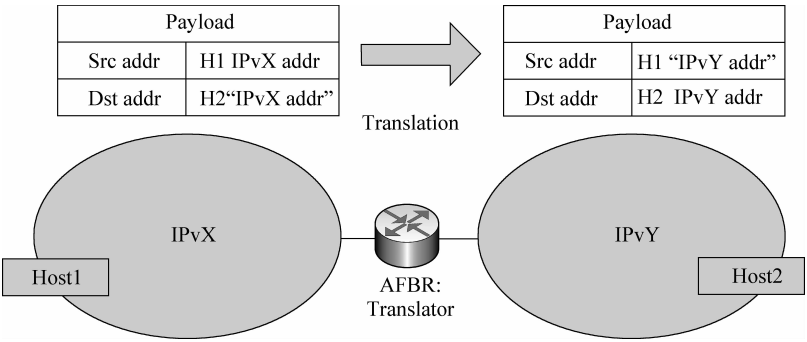

图 3.3　翻译技术原理图

IPv4/IPv6 翻译器通常具有两个网络接口，分别连接 IPvX 网络和 IPvY 网络。当位于 IPvX 网络中的主机需要访问位于 IPvY 网络中的节点时，它需要预先获知目的 IPvY 网络节点的地址在 IPvX 网络中的相应表示，并以此作为目的地址。报文经正常的 IPvX 路由系统，到达 IPv4/IPv6 翻译器的 IPvX 网络接口，翻译器根据配置情况以及相关算法对报文头进行翻译，使之成为 IPvY 网络报文，并从 IPvY 网络接口转发到 IPvY 网络。该报文最终经 IPvY 路由系统到达目的节点。反向通信过程与此类似。

假设 IPvX 网络中的主机 1(Host1)是通信的发起者，IPvY 网络中的主机 2(Host2)是通信的目的端。在通信发起前，Host1 必须通过某种方式(例如 DNS64 等)获得 Host2 在 IPvX 网络中的相应地址。Host1 会以本身接口的 IPvX 地址作为源地址，以获取的 Host2 在 IPvX 网络中表示的地址为目的地址，将数据包发送出去。经 IPvX 网络路由到达 IPv4/IPv6 翻译器后，翻译器根据相关算法从此数据包的目的 IPvX 地址中获取目的 IPvY 地址，同时需要将 H1 的源 IPvX 地址表示为 IPvY 地址。此 IPvY 地址在翻译过程中由翻译器分配或翻译获得。翻译器以 Host1 的 IPvY 网络中所表示的地址作为源地址，以翻译获得的 Host2 的 IPvY 地址为目的地址，将数据包转发入 IPvY 网络，经过正常路由至 Host2。除了相关的编址操作，网络路由需要保证以 Host2 所使用的 IPvX 地址为目的地址的 IPvX 包，和以 Host1 使用的 IPvY 地址为目的地址的 IPvY 包被路由至翻译器。在一定程度上，IPv4-IPv6 翻译机制与 IPv4 的网络地址翻译技术 NAT 类似，通过对两种异构网络的地址进行翻译，实现处在两种网络中的用户的直接通信。

翻译技术基本的数据层面操作是 IPv4-IPv6 数据报文的翻译，涉及网络层、传输层以及应用层。数据层面的行为包括地址和端口转换，IP/TCP/UDP 协议字段的翻译，以及应用层翻译(详见第 4 章)。翻译技术为弥合 IPv4 与 IPv6 定义上的差异，需要对两种协议报文的分片和重组、路径最大传输单元(Maximum Transmission Unit，MTU)、ICMP 等相关部分进行翻译处理。在控制层面，翻译技术根据 IPv4-IPv6 地址转换规则进行报文的翻译。可以通过预先部署特定的地址规划方案，或者动态建立地址绑定关系来实现。异构编址和相关的路由机制需要基于地址转换规则做相应的调整。

3.4.2　翻译技术的主要组成元素

翻译技术的主要组成部分包括[10]：地址翻译、IP/ICMP 翻译、翻译技术的状态维护、

DNS64 和 DNS46 以及其他应用层协议的翻译。

1. 地址翻译

翻译技术采用特定的 IPv6 地址格式，一般将 32b(位)IPv4 地址被嵌入到 IPv6 地址中。如果翻译技术使用由运营商指定的网络 IPv6 前缀，则 IPv4 地址需要与其他信息构成 IPv6 地址后缀从而形成 128b 的地址；如果采用 IANA 定义的翻译技术 IPv6 专用前缀，则将 32b IPv4 地址与前缀连接起来就可以形成所需的 IPv6 地址。所谓地址翻译，即翻译节点从 IPv6 地址中特定位置获取 IPv4 地址，或通过 IPv4 地址组建 IPv6 地址的过程，进而实现 IPv6 网络与 IPv4 网络的互通。

2. 其他字段翻译

IP 报文以及 ICMP 报文中，除了源地址与目的地址外，还有其他字段用于携带报文特定的信息，例如分片 ID、校验和、禁止分片标记[16, 17]等。在 IPv4 地址与 IPv6 地址的双向翻译过程中，IPv4 报文头中携带的信息不能丢失。因此翻译技术所采用的算法需要保证 IPv4 信息的完整性，以便接收端协议栈进行相应的操作。

3. 翻译技术的状态维护

翻译技术按照网络侧翻译器是否需要维护动态的 IPv4 与 IPv6 映射，分为无状态翻译技术和有状态翻译技术。无状态翻译技术通过翻译算法，使得 IPv4 地址与 IPv6 地址之间具有确定的映射关系。这种技术适用于目的 IPv4 地址可通过配置的“IPv4 无状态翻译前缀”转换成特定地址区间的可翻译 IPv6 地址的情况。而有状态翻译技术则支持动态建立 IPv4 地址与 IPv6 地址之间的映射关系，针对每流数据进行映射状态的维护。当目的 IPv4 地址无法翻译为特定区间的 IPv6 地址时，需要使用有状态翻译技术。

4. DNS64 和 DNS46

DNS64[18]与 DNS46 的主要功能是实现域名查询的 A 记录(IPv4 回复)与 AAAA 记录(IPv6 回复)的双向翻译。通常有两种实现方式。一种是静态配置 DNS 记录，将 DNS 服务器的解析记录都配置 A 记录和 AAAA 记录。另一种方式是动态翻译，即当用户发起 A 记录查询时，如果 DNS 服务器没有相应的 A 记录，则会回复相应的 AAAA 记录。对于 AAAA 记录查询时，如果 DNS 服务器没有相应的 AAAA 记录，则会回复相应的 A 记录。通过域名解析的过程，告知用户目的站点对应的异构 IP 地址。

5. 其他应用层协议的翻译

由于某些应用无法在 NAT 环境下正常工作，也就无法在 IPv4/IPv6 翻译的环境下正常工作，如 FTP 的主动模式。一种解决方案是使用应用层网关协助这类应用实现翻译工作。第 4 章第 4.3 节将做详细讨论。

3.4.3 翻译技术的两种模型

目前的主要翻译技术可以分为无状态翻译和有状态翻译[10]。对于无状态翻译技术，翻译所需的信息被嵌在 IPv6 地址中，并在 IPv4/IPv6 翻译器上有相关配置，支持由 IPv4 网络发起的访问 IPv6 网络的连接和由 IPv6 网络发起的访问 IPv4 网络的连接。无状态翻译通常会对配置到 IPv6 节点的 IPv6 地址格式有一定的限制，因为 IPv6 节点需要使用特定的算法实现与目标 IPv4 的通信。有状态翻译技术需要在翻译器处维护状态，在翻译过程中动态建立地址映射状态。状态由 IPv4 地址/传输层端口对与 IPv6 地址/传输层端口对组成，可

以支持 IPv6 系统发起的与 IPv4 系统的通信连接。两类翻译技术可适用于不同的场景。

翻译技术可以实现 IP 数据包在 IPv4 和 IPv6 两种协议簇之间的转换。无论客户端处在传输网络还是接入网络，翻译技术都能使之访问网络服务，并且无论其他网络中的用户使用的是何种网络协议，翻译技术都能实现它们之间的互通。然而，翻译技术并非网络发展的长期支持策略，但作为中远期的一种过渡策略，翻译技术可以作为互联网向 IPv6 演进过程中的重要技术。

3.4.4 翻译技术的优势与缺陷

翻译技术可以实现 IPv4 与 IPv6 网络的直接互通，在单栈传输网络的情况下，实现一种 IP 地址在另一种 IP 网络中的表示。因此翻译技术可以在单栈情况下为用户提供 IPv4 和 IPv6 的服务。基于翻译技术实现的 46 互通特性，也可以有更多的创新和扩展。

但翻译技术机制存在着一些缺陷。

1. 可扩展性问题

翻译技术可以使用 IANA 分配的专有 IPv6 前缀，或使用运营商确定的网络特定前缀。由于知名 IPv6 前缀较长(96b)，因此如果将专用 IPv6 前缀用于自治域间的路由，会导致 IPv6 路由聚合相对困难[7]。而且广播此前缀的自治域必须支持翻译技术。如果运营商使用网络特定前缀提供翻译服务，则需要对网络进行谨慎的规划，以免将 IPv4 的路由信息引入 IPv6 路由中，造成 IPv6 路由聚合困难。

2. 破坏端到端特性

翻译技术通过翻译器实现 IPv4 与 IPv6 地址转换，终端只能获取对端在本网络中的地址表示，而无法直接与对端通信：翻译器将通信终端“隐藏”了。翻译技术破坏了互联网的端到端特性，导致某些应用无法正常运行。

3. 异构地址寻址问题

为了进行通信，发起端必须知道目的端所对应的翻译地址。通信的双方中至少有一方能够感知翻译技术：发起端根据翻译技术去构造自己的翻译地址，或是目的端以某种方式将自己的地址通告给发起端。一般来讲可以采用 DNS 应用层网关或 DNS64 等方法来达到上述目的。但是当 DNS 应用层网关和 DNS64 不可用时，且通信双方的域名并没有在 DNS 服务器上注册，则需要对应用程序行为进行修改才能解决。因此翻译技术会破坏对上层应用的透明性。

4. 应用层翻译问题

理论上讲，翻译器应该支持应用层翻译的功能。但是在实际使用中，实时的应用层翻译是不可能做到的，因为在大量的网络设备上实现应用层翻译会造成巨大的成本和效率等方面的消耗，并且由于上层应用不尽相同、多种多样，也几乎不可能完全满足这样的需求。

根本上讲，IPv4/IPv6 翻译技术与当前广泛应用的 IPv4 NAT 有一定相似性。然而由于 IPv6 地址空间(2^{128})与 IPv4 地址空间(2^{32})的严重不对等，致使 IPv6 地址向 IPv4 地址的翻译会有语义信息的丢失。因此将翻译技术运用到大规模的网络上，其技术难度比 IPv4 NAT 大很多。

3.4.5 翻译技术小结

翻译技术对 IPv4 和 IPv6 报文的信息直接进行翻译，支持 IPv4 和 IPv6 网络的互操作。

但是翻译技术会涉及网络层、传输层和应用层等相关协议字段的翻译。此外，为协调 IPv4 和 IPv6 协议本身的一些区别，保证跨协议的语义正确性，还需要特殊处理分片重组与路径 MTU 检测、ICMP 协议等。从根本上说，这是由于翻译打破端到端特性，以及 IPv4-IPv6 地址空间不对称导致的。目前也有相关技术应对这些问题，如端口控制协议 PCP[19]。

3.5 隧道技术

在 IPv6 过渡时期，无论网络使用何种网络协议，都应同时支持 IPv4 服务和 IPv6 服务，并保证传输协议对上层应用透明。双栈技术可以较好地支持 IPv4 与 IPv6，但是实现全网的双栈部署需要巨额投资，各个网络参与者也存在较大的惰性。翻译技术直接实现 IPv4 节点与 IPv6 节点的通信，但是翻译过程涉及元素较多，机制较为复杂，而且会破坏端到端特性。隧道技术机制相对简单灵活，能够保持端到端特性，具有较好的可扩展性。

隧道过渡技术采用封装机制，将完整的 IP 报文作为负载进行传输，以穿越单栈传输网络。对于被封装的 IP 的通信两端来说，就像在异构网络中为其报文建立了一条虚拟隧道，从而能充分利用已有的路由系统，实现跨越异构网络的通信，并保证对上层应用的透明性。

3.5.1 隧道技术原理

隧道技术可以跨越 IPv4 网络提供 IPv6 服务(IPv6-over-IPv4)，也可以支持跨越 IPv6 网络提供 IPv4 服务(IPv4-over-IPv6)。隧道技术的基本操作包括封装、解封装以及隧道端点之间的发现机制，只涉及网络层，对其他层的行为没有影响。本节使用 IPvX 表示 IPv4 与 IPv6 网络中的一种，用 IPvY 表示另一种，具体介绍隧道技术的原理。

隧道技术基本原理如图 3.4 所示。处于 IPvY 网络中的主机需要与处在另一个 IPvY 网络中的主机进行通信，但是二者中间的传输网络为 IPvX 网络。为了实现 IPvY 网络跨越中间的 IPvX 网络进行互访，需要在 IPvX 连接两侧 IPvY 网络处部署隧道端点设备，用以建立 IPvY-over-IPvX 隧道。隧道端点设备支持 IPvX/IPvY 双栈，可以为 AFBR 或者主机设备。隧道端点设备负责对到达的报文进行封装和解封装操作，并将其转发入相应的网络。

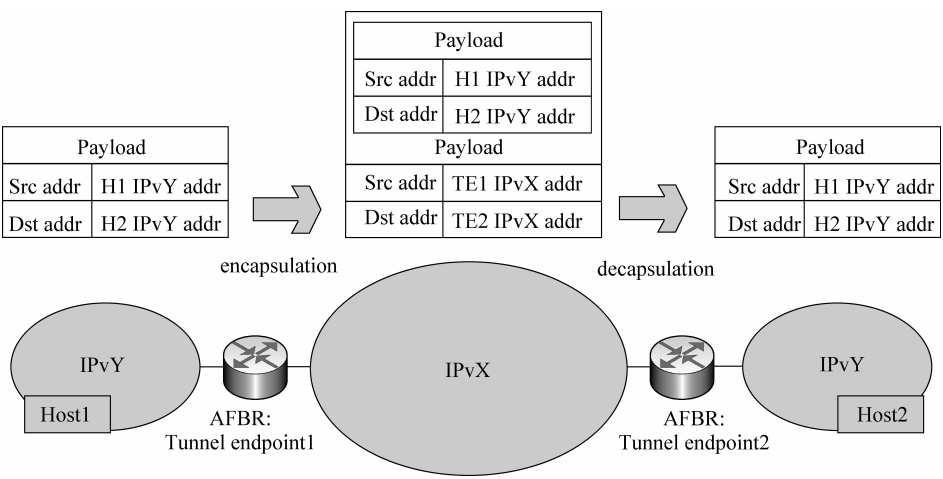

图 3.4 隧道技术原理图

假设位于 IPvY 网络中主机 1(Host1)为通信发起者,需要与主机 2(Host2)建立连接进行通信。Host1 以自己的 IPvY 地址为源地址、以 Host2 的 IPvY 地址为目的地址发起建立连接的请求。经 IPvY 网络的路由系统,Host1 发送的数据包到达隧道入口(隧道端点 1),隧道端点 1 会将整个 IPvY 报文作为 IPvX 报文的部分负载。该 IPvX 报文的源地址为隧道端点 1 的 IPvX 地址,目的地址为隧道出口(隧道端点 2)的 IPvX 地址。然后,封装后的报文会被转发进入 IPvX 网络,经 IPvX 路由系统到达隧道出口。隧道出口的隧道端点 2 接收该 IPvX 报文后,进行解封装得到原 IPvY 报文,将其转发进入 IPvY 网络中,经 IPvY 路由到达 Host2。由 Host2 向 Host1 发起连接的过程与此类似。

隧道技术数据层面主要是对报文进行封装和解封装操作。这种封装和解封装操作是将整个 IPvY 报文作为另一种协议的负载,因此内部封装的 IP 报文信息的完整性得以保留。在 IPv6 过渡场景下,可以采用多种封装方式,如 IP-IP[20]、通用路由封装(Generic Routing Encapsulation,GRE)[21]、二层隧道协议(Layer Two Tunneling Protocol,L2TP)[22]、多协议标签交换协议(Multiple protocol Label Switching,MPLS)[23]、IPsec[24] 等。这为运营商部署隧道技术时提供了较大的选择空间,运营商可根据实际情况选择相应的方案。

为保证数据层面的正确封装转发,在控制层面上,需要实现跨越 IPvX 网络的 IPvY 路由交互,以及隧道端点对封装地址映射状态的维护。在较为简单的网络环境中(如运营商内部网络或者两个运营商间的指定路径),可以采取传统的配置隧道实现异构穿越。通常配置隧道适用于隧道端点均为路由器的场景,网络管理者需要在隧道端点预先静态配置相关参数,例如每隧道所使用的 IP 地址、目的 IP 地址、路径 MTU 等。而对于更通用的过渡场景,尤其是在穿越 NAT、动态 IP 地址分配等场景下,可能涉及在隧道两端进行隧道端点地址的选择、动态/静态配置、状态维护等问题,需要其他更灵活的隧道机制。针对不同场景下的过渡需求,出现了多种新型隧道机制。

3.5.2 隧道技术的两种模型

根据网络结构,可以将隧道技术分为两种模型:星型模型(Hubs and Spokes)和网状互连模型(Mesh)[25]。二者的主要区别在于 IPv4 或 IPv6 孤岛控制的连接数和相关路由数目。星形模型只涉及单个连接和静态的默认路由,而网状模型则设计多个连接以及子网的路由信息。

星型模型主要由多个隧道发起点和单个隧道汇聚点组成。隧道发起点和汇聚点都是双栈设备。隧道发起点主要负责建立隧道,对数据包进行封装和解封装操作,并通过静态默认路由将数据包路由至隧道汇聚点。隧道汇聚点是隧道的终点,具有边界路由器的功能,连接两种异构网络。隧道汇聚点负责维护两种异构地址的绑定表,以确保对数据包进行正确的封装、解封装。星型模型主要适用于接入网的过渡场景。

网状模型主要适用于主干传输网络的过渡场景。该模型由多个对等节点组成,每个对等节点都可以与其他节点直接建立隧道。这些对等节点作为主干网的边缘路由器,连接接入网络和主干传输网络,主干网的核心路由器只支持某一种网络协议栈,而边缘路由器是双栈路由器。边缘路由器维护两种异构网络的路由信息,基于这些信息以及主干网的路由协议建立与其他对等节点的绑定表,从而实现正确的路由和封装、解封装。

3.5.3 隧道技术的优势与缺陷

隧道技术通过构建“虚拟 IP 连接”,实现跨越异构网络的通信。隧道技术具有以下优势。

1. 确保双栈服务

隧道技术通过对网络层协议数据报文进行异构封装,构建虚拟网络隧道,实现跨异构网络的数据传输。尽管传输网络为单栈网络,通过隧道技术可以向用户提供双栈服务,也可以保证互联网内容商在对网络设备进行少量升级的情况下,为 IPv4/IPv6 用户提供相应的服务。隧道技术可以通过单栈传输网络,实现双栈服务需求。

2. 机制简单易行

在隧道技术中,IP 数据包被完整地封装入异构数据报文中,从而各个字段所携带的信息均无损失。隧道端点设备只需进行封装、解封装操作以及绑定表的维护,而不需要处理复杂的翻译算法,因此隧道技术的设备简单,易于设备厂商实现和运营商的日常维护管理。

3. 保持端到端透明性

隧道技术只涉及网络层的封装、解封装以及隧道端点之间的通信,而不需要应用层进行修改,从而避免了应用层网关的使用。原有的应用程序在不做修改的情况下,可以无缝兼容隧道技术,对底层传输协议并无感知。隧道技术可以保持端到端的透明性,使得新型技术,如 P2P、FTP 等,得以在过渡时期继续得到发展和普及。

4. 保证路由可扩展性

隧道技术将 IPv4 与 IPv6 路由进行隔离,IPv4 网络与 IPv6 网络各自维护路由系统,避免在 IPv6 网络中引入 IPv4 路由爆炸的问题,从而保证了过渡时期网络的路由可扩展性。因此隧道技术可用于大规模部署,有利于网络向 IPv6 过渡。

5. 适用场景丰富

隧道技术的星型模型和网状模型可满足不同过渡场景的需求;根据隧道类型,又可将过渡技术分为 IPv6-over-IPv4 隧道和 IPv4-over-IPv6 隧道,使得隧道技术适用于互联网向 IPv6 过渡的不同时期。

然而隧道技术也有其协议机制方面的缺陷。隧道技术实质上还是使用同种协议栈的通信双方的互通,只是二者之间的传输网络为异构网络。因而隧道技术无法实现 IPv4 用户与 IPv6 用户之间的直接互通,难以支持在此基础上发展的新型应用和服务。另外,由于隧道技术将完整 IP 报文作为另一种报文的负载,导致隧道报文与原始报文大小不同,因此会消耗网络带宽,并且会带来较为严重的分片和重组问题。

3.5.4 隧道技术小结

隧道技术通过对报文进行封装和解封装的方式建立虚拟连接[26],实现跨异构网络的双向通信。隧道技术只需在连接异构网络的边界路由器上维护相应的绑定关系,就可实现对 IP 报文的透明传输,是无状态且轻量级的。隧道技术机制简单,灵活性高,可以保证路由的可扩展性。隧道技术基本能够满足各种场景下的 IPv6 过渡需求。目前国际国内的 IPv6 过渡技术研究重点集中在隧道技术,尤其是 IPv4-over-IPv6 隧道技术。IETF 的重要工作组 Softwire 工作组近年来在 IPv4-over-IPv6 隧道技术的标准化方面做了大量工作[27]。但隧

道技术也存在只能实现相同协议之间的互联，不能解决传统的 IPv4 网络和 IPv6 网络直接互通的问题。

3.6 过渡技术的其他问题

IPv6 过渡技术是对 IP 体系结构的修改和补充。为了能够使得 IPv4 网络和 IPv6 网络共存和通信，需要考虑网络过渡、融合过程中产生的各种问题，例如分片与重组问题、DNS 选择问题以及应用层网关问题等。针对这些过渡技术可能面临的共有问题，过渡技术机制需要着重考虑解决方案，以免对 IPv6 后续发展产生限制。

3.6.1 分片与重组问题

在数据链路建立之前，网络节点需要通过动态或静态的方式确定链路最大传输单元(MTU)，即链路上传输的单个报文的最大容量，以便确定协议栈是否对报文进行分片并对 IP 报文头中相应字段(如 Don't Fregment 字段等)进行设置。接收端则需要根据报文头中与分片相关的字段携带的信息，确定收到的报文是否完整，以进行报文重组。

IPv4 报文头长度为 20B，而 IPv6 报文头长度为 40B。二者报文头部长度的差异，导致过渡方案在进行分片与重组处理时的操作不同。双栈技术不涉及两种 IP 地址转换，不存在地址变换的问题；翻译技术需要将报文的 IPv4 头与 IPv6 头进行双向转换，会导致转换后报文长度的变化；隧道技术则是需要在原来报文基础上，增加一个 IPv4 或 IPv6 报文头，也会导致报文总长度的变化。

在实际操作中可以有 3 种方式来应对：路径 MTU 发现、传输层协商(例如设置 TCP 的最大分段大小选项 MSS Option[28])、对报文进行分片。前两种方式的主要目的是通过协议栈通知上层应用，减小发送报文的总长度，从而避免分片。但是网络不同节点的网络接口之间的 MTU 并不相同，通过这两种方式无法保证报文不分片。因此翻译技术和隧道技术需要有特定的报文分片重组机制，以应对由于报文头转换导致的报文大小变化问题。

对于翻译技术而言，对 IPv4 与 IPv6 报文头中分片相关字段的翻译是其核心之一。IPv6 报文头有分片头字段(Fragment Header)，而在 IPv4 报文头中，相关信息是由标记字段(Flags)和分片位移字段(Fragment Offset)表示。在 IPv4 地址与 IPv6 地址的翻译过程中，需要结合翻译方向对这些字段进行语义翻译。关于翻译技术的分片重组问题，请参见 RFC 6145 和 RFC 6146 对翻译机制的介绍。

隧道技术不需要对这些具体的字段进行处理。但是封装增加了整个数据包的大小，因此，封装后得到的隧道包可能需要进行分片处理。对于 IPv6 隧道[29]，隧道包封装上了 IPv6 头，与正常的 IPv6 报文一样。因此封装后的 IPv6 报文只能在隧道入口节点处被分片，而在 IPv6 网络的传输过程中不能被路由器分片。隧道出口节点接收到隧道报文后，必须正确地完成对隧道包的解封装工作。如果是经过分片处理的隧道包，则应该对分片进行重组后再对封装的内容进行处理。

当隧道出口节点使用的是 IPv6 任意播地址时，分片的隧道报文到达该节点时会出现重组问题。所有的分片报文必须到达同一个使用该任意播地址的节点，这样才能根据报文头中的分片标识信息进行正确重组。但是，由于任意播的机制使报文可能到达使用同一任意

播地址的任意一个节点。这就对报文重组的需求无法在任意播的情况下保证。这个问题在原生 IPv6 分片报文到达使用任意播地址节点所面临的问题是一样的。

在过渡场景下,IPv6 隧道内部是 IPv4 报文。此时如果原始 IPv4 数据包的大小超过隧道 MTU 大小,应该按照以下方法处理。

(1) 如果 IPv4 原始数据包头的 Don't Fragment(DF)位标志是 SET,意味着该报文不允许分片传输,则入口节点应该丢弃该数据包,并回复一个 ICMP 报文,其中 type 为 Unreachable,code 为 Packet Too Big,并建议原始数据包的发起方将 MTU 值设为隧道 MTU 的大小。

(2) 如果原始数据包的 Don't Fragment(DF)位标志是 CLEAR,意味着该报文可以被分片,则隧道入口节点将原始数据包封装为隧道包,然后将封装后的隧道包进行 IPv6 分片处理,以保证其不超过隧道 MTU 的大小。这里由于是在隧道的入口节点,因此可以对封装后的隧道报文进行分片。

3.6.2 DNS 选择问题

用户侧客户端通过向 DNS 服务器进行查询获取目的节点的 IPv4 或 IPv6 地址,从而发起连接。但是无论是原生的双栈环境,还是通过过渡技术构建的虚拟双栈环境,用户侧都会面临 DNS 选择问题:DNS 服务器会同时返回 IPv4 应答和 IPv6 应答。当从 DNS 服务器返回的目的 IPv4 和 IPv6 地址中有一个不可达时,某些应用可能无法迅速完成连接切换。当前双栈节点的 DNS 处理流程如图 3.5 所示。双栈节点收到从 DNS 服务器返回的 IPv4 地址和 IPv6 地址后,对网络的 IPv6 连接情况并不感知,因此仍试图使用 IPv6 与服务器建立连接。连接失败后,客户端需要一段时间确认 IPv6 确实不能使用,这时,客户端才会尝试使用 IPv4 建立连接。内容提供商可以通过避免向 DNS 服务器宣告服务器的 IPv6 地址的方式,减少这种连接切换的情况,以保证用户体验。但是缺少支持 IPv6 的内容提供商的接入,会加剧上文提到的 IPv6 流量不足的压力,不利于推动 IPv6 发展。

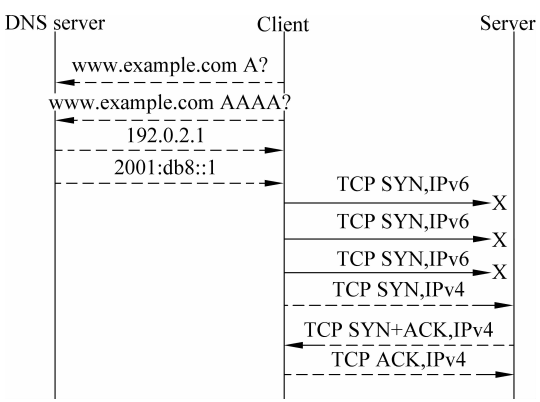

图 3.5 当前双栈节点 DNS 报文交互过程

RFC 6555[30]提出了一种称为 Happy Eyeballs 的算法,在网络中的 IPv6 连接不可达的情况下,从 DNS 客户端的处理过程入手,尽量减少双栈节点从一种网络协议簇向另一种网络协议簇转换的时延。该算法还要求客户端优先使用 IPv6 建立连接,从而推动 IPv6 的

过渡。

Happy Eyeballs 算法的思路是，无论 IPv6 是否可用，在收到从 DNS 服务器返回的 IPv4 地址和 IPv6 地址后，客户端会同时使用 IPv4 和 IPv6 向服务器发送 TCP SYN 信息，试图建立连接。如果 IPv6 无法使用(见图 3.6)，则服务器侧只会对 IPv4 的 TCP 连接产生应答，因此 IPv4 连接可以建立，应用程序可使用 IPv4 进行通信。如果网络中的 IPv6 可以使用(见图 3.7)，则客户端与服务器之间的 IPv4 连接和 IPv6 连接都可以建立，但是机制要求客户端优先使用 IPv6 通信，则客户端会发送 TCP RST 报文，从而中断 IPv4 连接。这种算法可以有效解决双栈节点 TCP 建立连接时延过长的问题，并使得 IPv6 得到更好的应用。

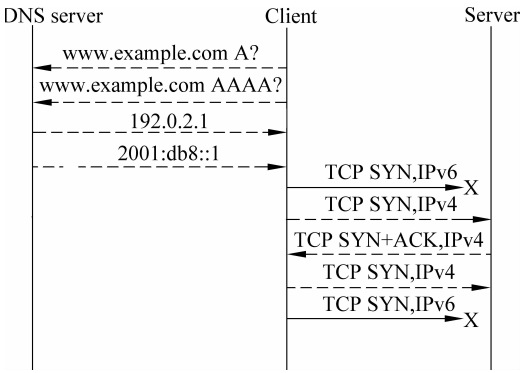

图 3.6　Happy Eyeballs 机制报文交互过程(IPv6 无法使用)

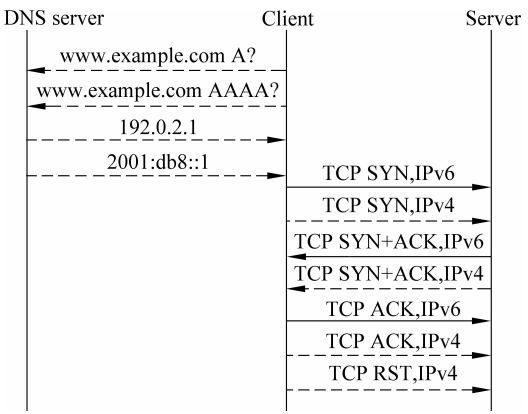

图 3.7　Happy Eyeballs 机制报文交互过程(IPv6 正常工作)

3.6.3　应用层翻译问题

在某些应用层协议建立连接的过程中，通信双方需在应用层报文中指定目的端的 IP 地址及端口信息，才能正常建立连接。这使得应用层协议出现了“跨层”的现象：在应用层报文中包含了网络层和传输层的信息，而且该信息必须与报文的真正网络层、传输层信息相匹配。跨层协议和应用目前很普遍，例如 FTP、SIP 等。应用层出现的跨层协议可以使得应用程序对网络连接有一定的控制力，但是会对网络的分层模型产生一定的影响。在 IPv6 过渡的场景下，这种影响更加明显，尤其对于翻译技术。

翻译技术需要将数据包的网络层信息进行转换，即把源 IP 地址和目的 IP 地址翻译为目标网络支持的 IP 地址类型。这种翻译操作会导致网络层 IP 地址信息与应用层报文中的 IP 地址信息不匹配，造成应用层协议控制层面无法正常工作。为了能支持这类应用，翻译技术既要能对网络层数据进行翻译，也能对应用层中携带 IP 信息的报文进行相应的翻译，即应用层翻译。

对于双栈技术和隧道技术，IPv4 与 IPv6 隔离不会存在 IPv4 地址与 IPv6 地址的直接转换，因此不需要在应用层进行翻译操作。跨层应用可以在双栈和隧道环境下正常工作。

3.7 本章小结

互联网的高速发展，导致 IPv4 地址急剧消耗，全网向 IPv6 过渡已成为下一代互联网发展的趋势。由于互联网运营商、内容提供商、设备制造商以及用户面临的 IPv6 过渡的问题不同，导致 IPv6 过渡具有一定的难度。为保证互联网向 IPv6 平滑过渡，在选择过渡策略时，需要遵循相应的原则。而在具体过渡技术设计过程中，则需要对多个重要的技术因素进行全面考量。双栈技术、翻译技术和隧道技术适用于不同的场景，从不同的角度解决 IPv6 过渡问题。通过对技术原理、部署模型以及优劣势的介绍，本章分析了当前互联网过渡的主要研究内容。随着过渡场景和各方需求的逐渐清晰，目前 IETF 对 IPv6 过渡技术研究重点已转移到新型隧道技术方案的制定和完善。

参考文献

[1] P. Wu, Y. Cui, J. Wu, et al. Transition from IPv4 to IPv6: A State-of-the-Art Survey. Communications Surveys & Tutorials of IEEE, 2013, 15 (3): 1407-1424.

[2] 崔勇，董江，徐明伟，等. IPv6 过渡场景分析. CCSA 行业标准，2012，7.

[3] J. Saltzer, D. Reed, D. Clark. End-to-end arguments in system design. ACM Transactions on Computer Systems (TOCS), 1984, 2 (4): 277-288.

[4] Information Sciences Institute University of Southern California. Transmission Control Protocol. IETF RFC 793, September 1981.

[5] K. Alagappan. Telnet Authentication: SPX. IETF RFC 1412, January 1993.

[6] E. Nordmark. Stateless IP/ICMP Translation Algorithm (SIIT). IETF RFC 2765, February 2000.

[7] C. Bao, C. Huitema, M. Bagnulo, et al. IPv6 Addressing of IPv4/IPv6 Translators. IETF RFC 6052, October 2010.

[8] http://bgp.potaroo.net/as6447/.

[9] 张威，毕军，吴建平. 互联网域间路由可扩展性. 软件学报，2011，22 (1): 84-100.

[10] F. Baker, X. Li, C. Bao, et al. Framework for IPv4/IPv6 Translation. IETF RFC 6144, April 2011.

[11] E. Nordmark, R. Gilligan. Basic Transition Mechanisms for IPv6 Hosts and Routers. IETF RFC 4213, October 2005.

[12] R. Droms. Dynamic Host Configuration Protocol. IETF RFC 2131, March 1997.

[13] S. Thomson, T. Narten. IPv6 Stateless Address Auto-configuration. IETF RFC 2462, December 1998.