

DNS 接收到一个提供所希望映射的 DNS 回答报文。这个映射结果则被传递到调用 DNS 的应用程序。因此，从用户主机上调用应用程序的角度看，DNS 是一个提供简单、直接的转换服务的黑盒子。但事实上，实现这个服务的黑盒子非常复杂，它由分布于全球的大量 DNS 服务器以及定义了 DNS 服务器与查询主机通信方式的应用层协议组成。

DNS 的一种简单设计是在因特网上只使用一个 DNS 服务器，该服务器包含所有的映射。在这种集中式设计中，客户直接将所有查询直接发往单一的 DNS 服务器，同时该 DNS 服务器直接对所有的查询客户做出响应。尽管这种设计的简单性非常具有吸引力，但它不适用于当今的因特网，因为因特网有着数量巨大（并持续增长）的主机。这种集中式设计的问题包括：

- **单点故障**（a single point of failure）。如果该 DNS 服务器崩溃，整个因特网随之瘫痪！
- **通信容量**（traffic volume）。单个 DNS 服务器不得不处理所有的 DNS 查询（用于为上亿台主机产生的所有 HTTP 请求报文和电子邮件报文服务）。
- **远距离的集中式数据库**（distant centralized database）。单个 DNS 服务器不可能“邻近”所有查询客户。如果我们将单台 DNS 服务器放在纽约市，那么所有来自澳大利亚的查询必须传播到地球的另一边，中间也许还要经过低速和拥塞的链路。这将导致严重的时延。
- **维护**（maintenance）。单个 DNS 服务器将不得不为所有的因特网主机保留记录。这不仅将使这个中央数据库非常庞大，而且它还不得不为解决每个新添加的主机而频繁更新。

总的来说，在单一 DNS 服务器上运行集中式数据库完全没有可扩展能力。因此，DNS 采用了分布式的设计方案。事实上，DNS 是一个在因特网上实现分布式数据库的精彩范例。

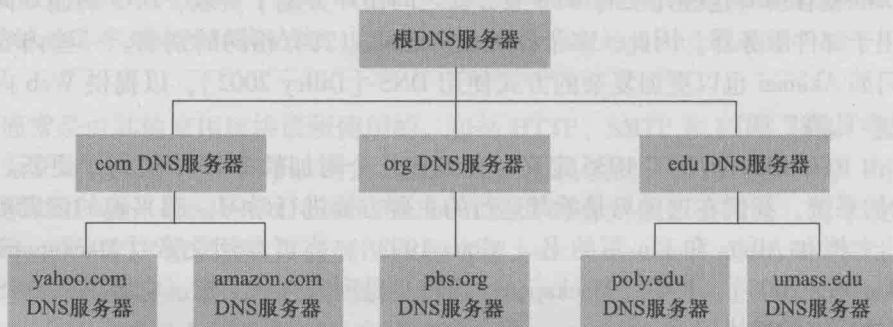


图 2-19 DNS 服务器的部分层次结构

1. 分布式、层次数据库

为了处理扩展性问题，DNS 使用了大量的 DNS 服务器，它们以层次方式组织，并且分布在全世界范围内。没有一台 DNS 服务器拥有因特网上所有主机的映射。相反，该映射分布在所有的 DNS 服务器上。大致说来，有 3 种类型的 DNS 服务器：根 DNS 服务器、顶级域（Top-Level Domain, TLD）DNS 服务器和权威 DNS 服务器。这些服务器以图 2-19 中所示的层次结构组织起来。为了理解这 3 种类型的 DNS 服务器交互的方式，假定一个 DNS 客户要决定主机名 `www. amazon. com` 的 IP 地址。粗略说来，将发生下列事件。客户首