生带着便携机从宿舍到图书馆再到教室。很有可能在每个位置这个学生将连接到一个新的子网，因此在每个位置都需要一个新的 IP 地址。DHCP 是适合这种情形的理想方法，因为有许多用户来来往往，并且仅在有限的时间内需要地址。DHCP 的即插即用能力的价值是显然的，因为下列情况是不可想象的：系统管理员在每个位置能够重新配置便携机，并且少数学生（除了那些上过计算机网络课程的学生）让专家人工地配置他们的便携机。

DHCP 是一个客户 - 服务器协议。客户通常是新到达的主机，它要获得包括自身使用的 IP 地址在内的网络配置信息。在最简单场合下，每个子网（在图 4-20 的编址意义下）将具有一台 DHCP 服务器。如果在某子网中没有服务器，则需要一个 DHCP 中继代理（通常是一台路由器），这个代理知道用于该网络的 DHCP 服务器的地址。图 4-23 显示了连接到子网 223.1.2/24 的一台 DHCP 服务器，具有一台提供中继代理服务的路由器，它为连接到子网 223.1.1/24 和 223.1.3/24 的到达客户提供 DHCP 服务。在我们下面的讨论中，将假定 DHCP 服务器在该子网上是可供使用的。

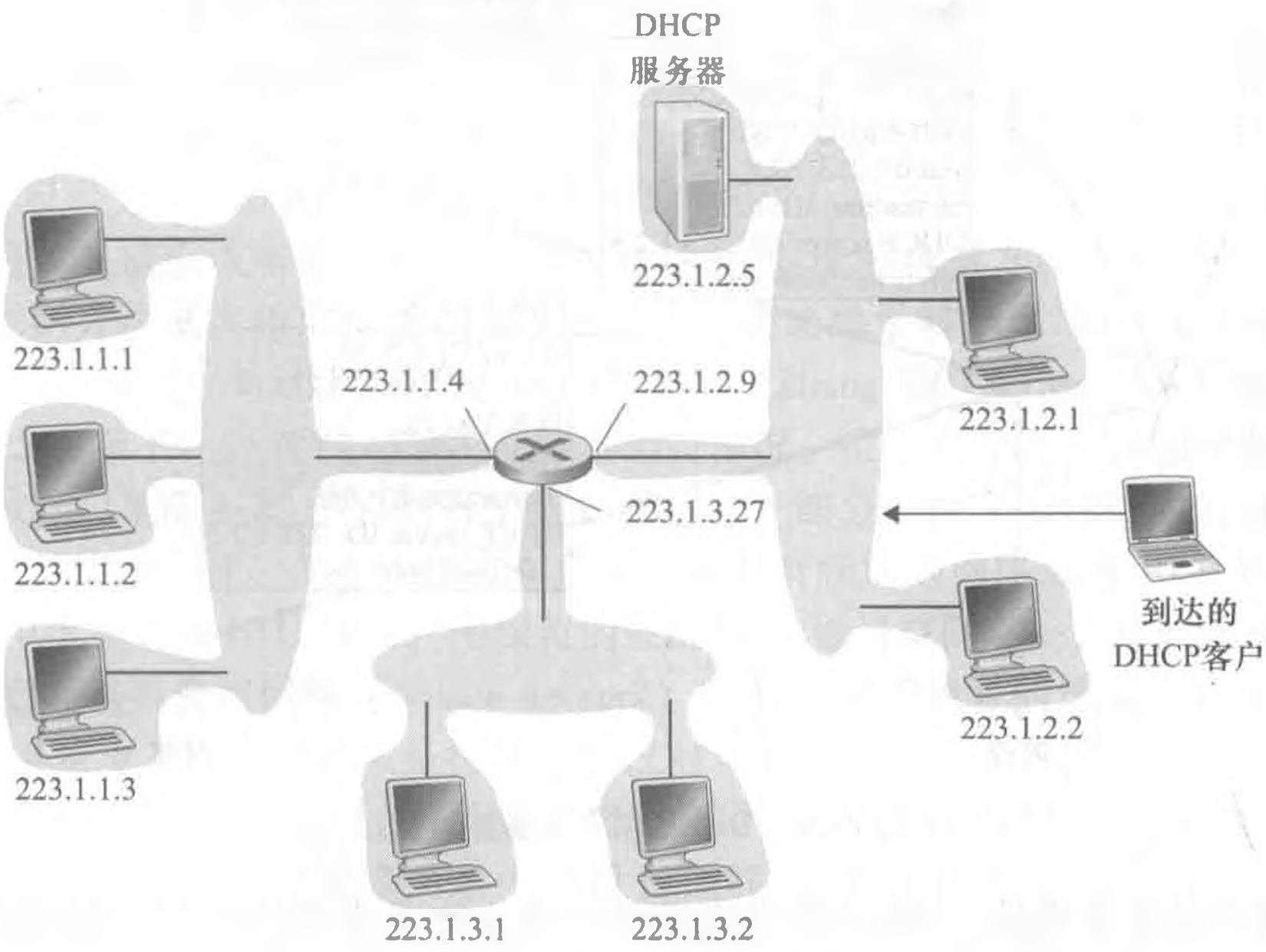

图 4-23　DHCP 客户和服务器

对于一台新到达的主机而言，针对图 4-23 所示的网络设置，DHCP 协议是一个 4 个步骤的过程，如图 4-24 中所示。在这幅图中，yiaddr（表示“你的因特网地址”之意）指示分配给该新到达客户的地址。

这 4 个步骤是：

- DHCP 服务器发现。一台新到达的主机的首要任务是发现一个要与其交互的 DHCP 服务器。这可通过使用 **DHCP 发现报文**（DHCP discover message）来完成，客户在 UDP 分组中向端口 67 发送该发现报文。该 UDP 分组封装在一个 IP 数据报中。但是这个数据报应发给谁呢？主机甚至不知道它所连接网络的 IP 地址，更不用说用于该网络的 DHCP 服务器地址了。在这种情况下，DHCP 客户生成包含 DHCP 发现报文的 IP 数据报，其中使用广播目的地址 255.255.255.255 并且使用“本主机”源 IP 地址 0.0.0.0。DHCP 客户将该 IP 数据报传递给链路层，链路层然后将该帧广播到所有与

该子网连接的节点（我们将在6.4节中涉及链路层广播的细节）。

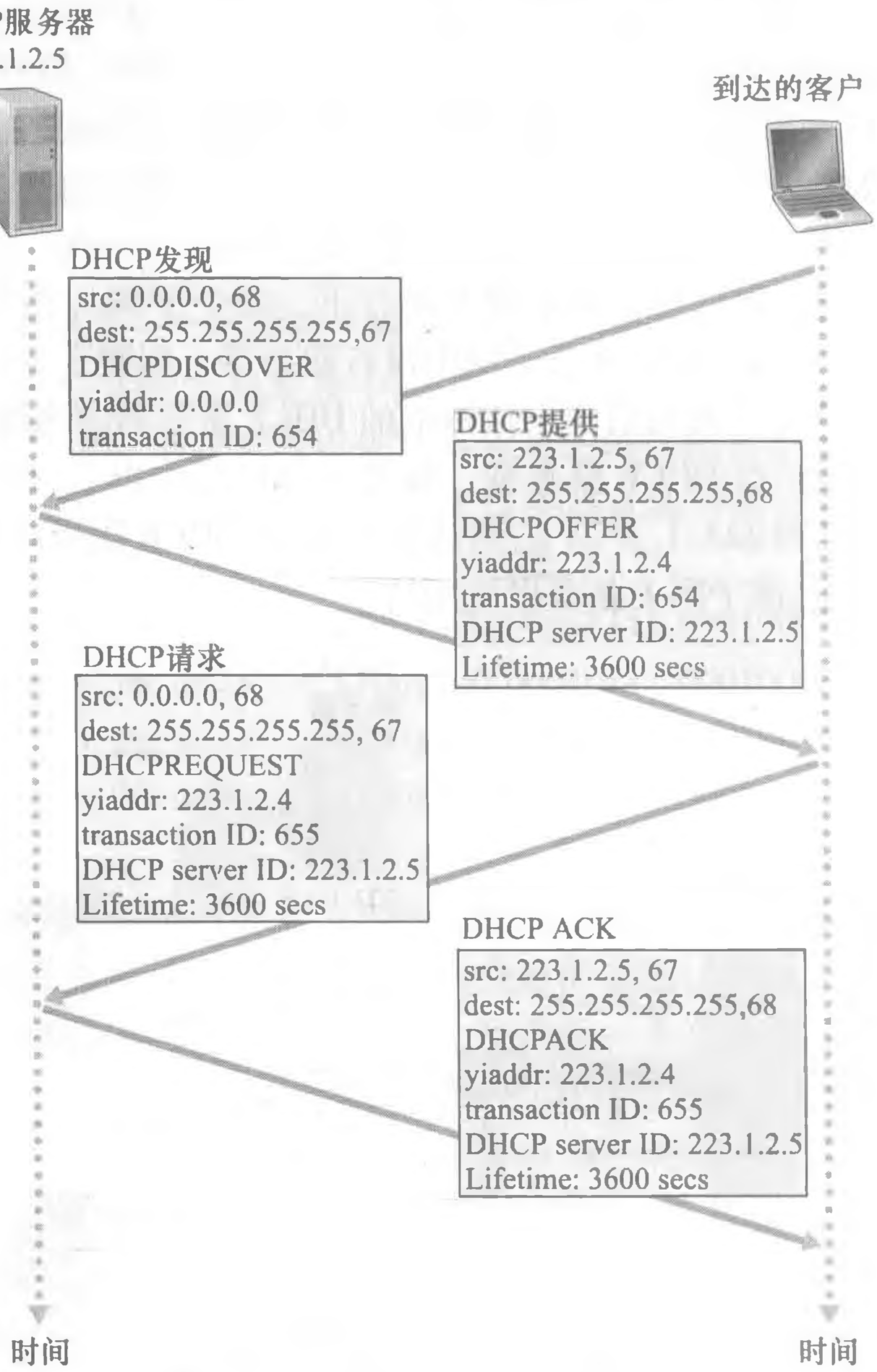

图4-24　DHCP客户－服务器交互

- DHCP服务器提供。DHCP服务器收到一个DHCP发现报文时，用**DHCP提供报文**（DHCP offer message）向客户做出响应，该报文向该子网的所有节点广播，仍然使用IP广播地址255.255.255.255（你也许要思考一下这个服务器为何也必须采用广播）。因为在子网中可能存在几个DHCP服务器，该客户也许会发现它处于能在几个提供者之间进行选择的优越位置。每台服务器提供的报文包含有收到的发现报文的事务ID、向客户推荐的IP地址、网络掩码以及IP**地址租用期**（address lease time），即IP地址有效的时间量。服务器租用期通常设置为几小时或几天［Droms 2002］。
- DHCP请求。新到达的客户从一个或多个服务器提供中选择一个，并向选中的服务器提供用**DHCP请求报文**（DHCP request message）进行响应，回显配置的参数。
- DHCP ACK。服务器用**DHCP ACK报文**（DHCP ACK message）对DHCP请求报文进行响应，证实所要求的参数。

一旦客户收到DHCP ACK后，交互便完成了，并且该客户能够在租用期内使用DHCP

分配的 IP 地址。因为客户可能在该租用期超时后还希望使用这个地址，所以 DHCP 还提供了一种机制以允许客户更新它对一个 IP 地址的租用。

从移动性角度看，DHCP 确实有非常严重的缺陷。因为每当节点连到一个新子网，要从 DHCP 得到一个新的 IP 地址，当一个移动节点在子网之间移动时，就不能维持与远程应用之间的 TCP 连接。在第 6 章中，我们将研究移动 IP，它是一种对 IP 基础设施的扩展，允许移动节点在网络之间移动时使用其单一永久的地址。有关 DHCP 的其他细节可在［Droms 2002］与［dhc 2016］中找到。一个 DHCP 的开放源码参考实现可从因特网系统协会［ISC 2016］得到。

4.3.4　网络地址转换

讨论了有关因特网地址和 IPv4 数据报格式后，我们现在可清楚地认识到每个 IP 使能的设备都需要一个 IP 地址。随着所谓小型办公室、家庭办公室（Small Office，Home Office，SOHO）子网的大量出现，看起来意味着每当一个 SOHO 想安装一个 LAN 以互联多台机器时，需要 ISP 分配一组地址以供该 SOHO 的所有 IP 设备（包括电话、平板电脑、游戏设备、IP TV、打印机等）使用。如果该子网变大了，则需要分配一块较大的地址。但如果 ISP 已经为 SOHO 网络的当前地址范围分配过一块连续地址该怎么办呢？并且，家庭主人一般要（或应该需要）首先知道的管理 IP 地址的典型方法有哪些呢？幸运的是，有一种简单的方法越来越广泛地用在这些场合：**网络地址转换**（Network Address Translation，NAT）［RFC 2663；RFC 3022；Huston 2004；Zhang 2007；Cisco NAT 2016］。

图 4-25 显示了一台 NAT 使能路由器的运行情况。位于家中的 NAT 使能的路由器有一个接口，该接口是图 4-25 中右侧所示家庭网络的一部分。在家庭网络内的编址就像我们在上面看到的完全一样，其中的所有 4 个接口都具有相同的网络地址 10.0.0/24。地址空间 10.0.0.0/8 是在［RFC 1918］中保留的三部分 IP 地址空间之一，这些地址用于如图 4-25 中的家庭网络等**专用网络**（private network）或**具有专用地址的地域**（realm with private address）。具有专用地址的地域是指其地址仅对该网络中的设备有意义的网络。为了明白它为什么重要，考虑有数十万家庭网络这样的事实，许多使用了相同的地址空间 10.0.0.0/24。在一个给定家庭网络中的设备能够使用 10.0.0.0/24 编址彼此发送分组。然而，转发到家庭网络之外进入更大的全球因特网的分组显然不能使用这些地址（或作为源地址，或作为目的地址），因为有数十万的网络使用着这块地址。这就是说，10.0.0.0/24 地址仅在给定的网络中才有意义。但是如果专用地址仅在给定的网络中才有意义的话，当向或从全球因特网发送或接收分组时如何处理编址问题呢，地址在何处才必须是唯一的呢？答案在于理解 NAT。

NAT 使能路由器对于外部世界来说甚至不像一台路由器。相反 NAT 路由器对外界的行为就如同一个具有单一 IP 地址的单一设备。在图 4-25 中，所有离开家庭路由器流向更大因特网的报文都拥有一个源 IP 地址 138.76.29.7，且所有进入家庭的报文都拥有同一个目的 IP 地址 138.76.29.7。从本质上讲，NAT 使能路由器对外界隐藏了家庭网络的细节。（另外，你也许想知道家庭网络计算机是从哪儿得到其地址，路由器又是从哪儿得到它的单一 IP 地址的。在通常的情况下，答案是相同的，即 DHCP！路由器从 ISP 的 DHCP 服务器得到它的地址，并且路由器运行一个 DHCP 服务器，为位于 NAT-DHCP 路由器控制的家庭网络地址空间中的计算机提供地址。）

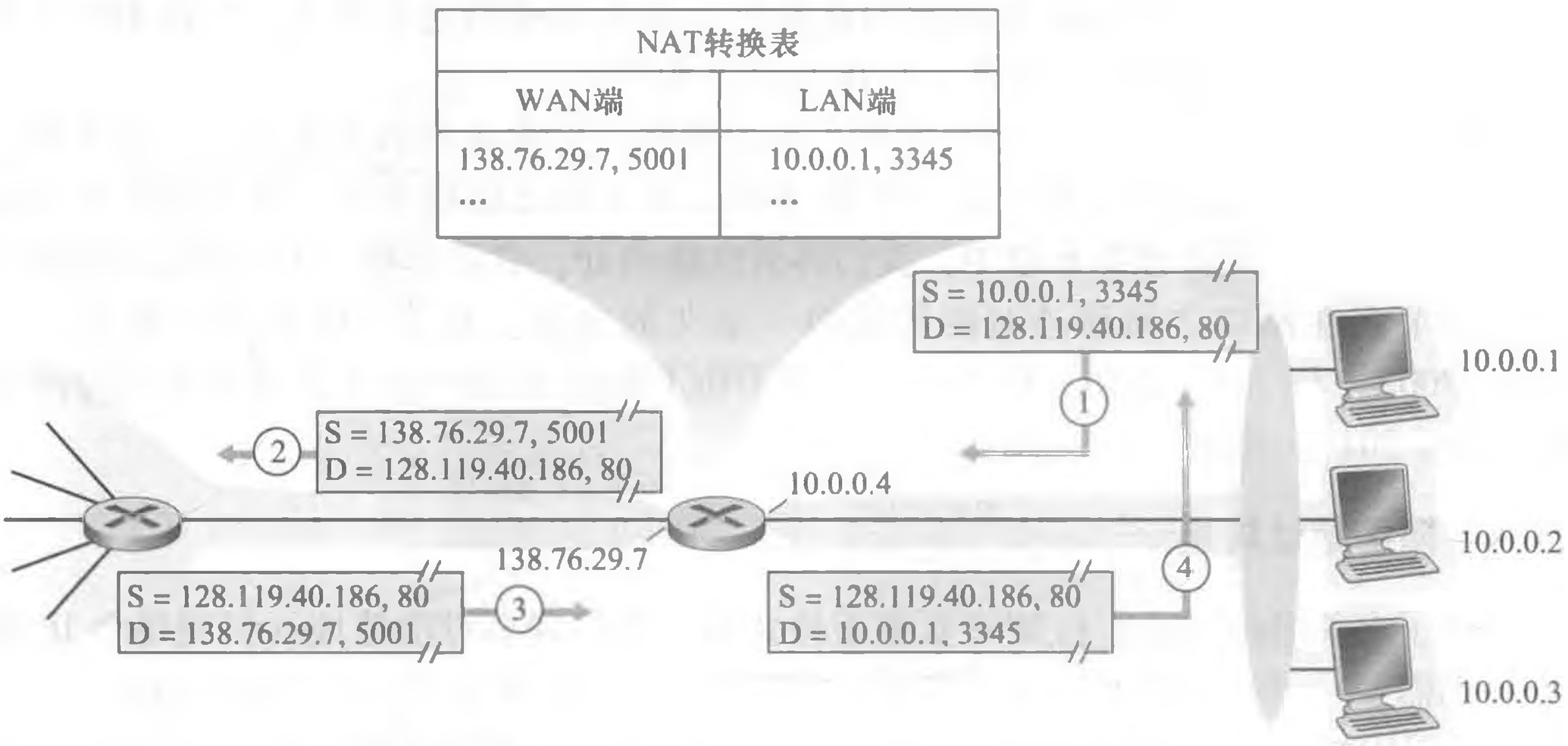

图 4-25　网络地址转换

如果从广域网到达 NAT 路由器的所有数据报都有相同的目的 IP 地址（特别是对 NAT 路由器广域网一侧的接口），那么该路由器怎样知道它应将某个分组转发给哪个内部主机呢？技巧就是使用 NAT 路由器上的一张 **NAT 转换表**（NAT translation table），并且在表项中包含了端口号及其 IP 地址。

考虑图 4-25 中的例子。假设一个用户坐在家庭网络主机 10.0.0.1 后，请求 IP 地址为 128.119.40.186 的某台 Web 服务器（端口 80）上的一个 Web 页面。主机 10.0.0.1 为其指派了（任意）源端口号 3345 并将该数据报发送到 LAN 中。NAT 路由器收到该数据报，为该数据报生成一个新的源端口号 5001，将源 IP 替代为其广域网一侧接口的 IP 地址 138.76.29.7，且将源端口 3345 更换为新端口 5001。当生成一个新的源端口号时，NAT 路由器可选择任意一个当前未在 NAT 转换表中的源端口号。（注意到因为端口号字段为 16 比特长，NAT 协议可支持超过 60 000 个并行使用路由器广域网一侧单个 IP 地址的连接！）路由器中的 NAT 也在它的 NAT 转换表中增加一表项。Web 服务器并不知道刚到达的包含 HTTP请求的数据报已被 NAT 路由器进行了改装，它会发回一个响应报文，其目的地址是 NAT 路由器的 IP 地址，其目的端口是 5001。当该报文到达 NAT 路由器时，路由器使用目的 IP 地址与目的端口号从 NAT 转换表中检索出家庭网络浏览器使用的适当 IP 地址（10.0.0.1）和目的端口号（3345）。于是，路由器重写该数据报的目的 IP 地址与目的端口号，并向家庭网络转发该数据报。

NAT 在近年来已得到了广泛的应用。但是 NAT 并非没有贬低者。首先，有人认为端口号是用于进程寻址的，而不是用于主机寻址的。这种违规用法对于运行在家庭网络中的服务器来说确实会引起问题，因为正如我们在第 2 章所见，服务器进程在周知端口号上等待入请求，并且 P2P 协议中的对等方在充当服务器时需要接受入连接。对这些问题的技术解决方案包括 **NAT 穿越**（NAT traversal）工具［RFC 5389］和通用即插即用（Universal Plug and Play，UPnP）。UPnP 是一种允许主机发现和配置邻近 NAT 的协议［UPnP Forum 2016］。

其次，纯粹的体系结构者提出了更为“原理性的”反对 NAT 的意见。这时，关注焦点在于路由器是指第三层（即网络层）设备，并且应当处理只能达到网络层的分组。NAT

违反主机应当直接彼此对话这个原则，没有干涉节点修改 IP 地址，更不用说端口号。但不管喜欢与否，NAT 已成为因特网的一个重要组件，成为所谓**中间盒**［Sekar 2011］，它运行在网络层并具有与路由器十分不同的功能。中间盒并不执行传统的数据报转发，而是执行诸如 NAT、流量流的负载均衡、流量防火墙（参见下面的插入框内容）等功能。我们将在随后的 4.4 节学习的通用转发范例，除了传统的路由器转发外，还允许一些这样的中间盒功能，从而以通用、综合的方式完成转发。

关注安全性

检查数据报：防火墙和入侵检测系统

假定你被赋予了管理家庭网络、部门网络、大学网络或公司网络的任务。知道你网络 IP 地址范围的攻击者，能够方便地在此范围中发送 IP 数据报进行寻址。这些数据报能够做各种不正当的事情，包括用 ping 搜索和端口扫描形成你的网络图，用恶意分组使易受攻击的主机崩溃，扫描你网络中服务器上的开放 TCP/UDP 端口，并且通过在分组中带有恶意软件来感染主机。作为网络管理员，你准备做些什么来将这些能够在你的网络中发送恶意分组的坏家伙拒之门外呢？对抗恶意分组攻击的两种流行的防御措施是防火墙和入侵检测系统（IDS）。

作为一名网络管理员，你可能首先尝试在你的网络和因特网之间安装一台防火墙。（今天大多数接入路由器具有防火墙能力。）防火墙检查数据报和报文段首部字段，拒绝可疑的数据报进入内部网络。例如，一台防火墙可以被配置为阻挡所有的 ICMP 回显请求分组（参见 5.6 节），从而防止了攻击者横跨你的 IP 地址范围进行传统的端口扫描。防火墙也能基于源和目的 IP 地址以及端口号阻挡分组。此外，防火墙能够配置为跟踪 TCP 连接，仅许可属于批准连接的数据报进入。

IDS 能够提供另一种保护措施。IDS 通常位于网络的边界，执行“深度分组检查”，不仅检查数据报（包括应用层数据）中的首部字段，而且检查其有效载荷。IDS 具有一个分组特征数据库，这些特征是已知攻击的一部分。随着新攻击的发现，该数据库自动更新特征。当分组通过 IDS 时，IDS 试图将分组的首部字段和有效载荷与其特征数据库中的特征相匹配。如果发现了这样的一种匹配，就产生一个告警。入侵防止系统（IPS）与 IDS 类似，只是除了产生告警外还实际阻挡分组。在第 8 章中，我们将更为详细地研究防火墙和 IDS。

防火墙和 IDS 能够全面保护你的网络免受所有攻击吗？答案显然是否定的，因为攻击者继续寻找特征还不能匹配的新攻击方法。但是防火墙和传统的基于特征的 IDS 在保护你的网络不受已知攻击入侵方面是有用的。

4.3.5　IPv6

在 20 世纪 90 年代早期，因特网工程任务组就开始致力于开发一种替代 IPv4 的协议。该努力的首要动机是以下现实：由于新的子网和 IP 节点以惊人的增长率连到因特网上（并被分配唯一的 IP 地址），32 比特的 IP 地址空间即将用尽。为了应对这种对大 IP 地址空间的需求，开发了一种新的 IP 协议，即 IPv6。IPv6 的设计者还利用这次机会，在 IPv4 积累的运行经验基础上加进和强化了 IPv4 的其他方面。

IPv4 地址在什么时候会被完全分配完（因此没有新的网络再能与因特网相连）是一个相当有争议的问题。IETF 的地址寿命期望工作组的两位负责人分别估计地址将于 2008 年和 2018 年用完［Solensky 1996］。在 2011 年 2 月，IANA 向一个区域注册机构分配完了未分配 IPv4 地址的最后剩余地址池。这些注册机构在它们的地址池中还有可用的 IPv4 地址，一旦用完这些地址，从中央池中将再也分配不出更多的可用地址块了［Huston 2011a］。IPv4 地址空间耗尽的近期调研以及延长该地址空间的寿命所采取的步骤见［Richter 2015］。

尽管在 20 世纪 90 年代中期对 IPv4 地址耗尽的估计表明，IPv4 地址空间耗尽的期限还有可观的时间，但人们认识到，如此大规模地部署一项新技术将需要可观的时间，因此研发 IP 版本 6（IPv6）［RFC 2460］的工作开始了［RFC 1752］。（一个经常问的问题是：IPv5 出了什么情况？人们最初预想 ST-2 协议将成为 IPv5，但 ST-2 后来被舍弃了。）有关 IPv6 的优秀信息来源见［Huitema 1998］。

1. IPv6 数据报格式

IPv6 数据报的格式如图 4-26 所示。

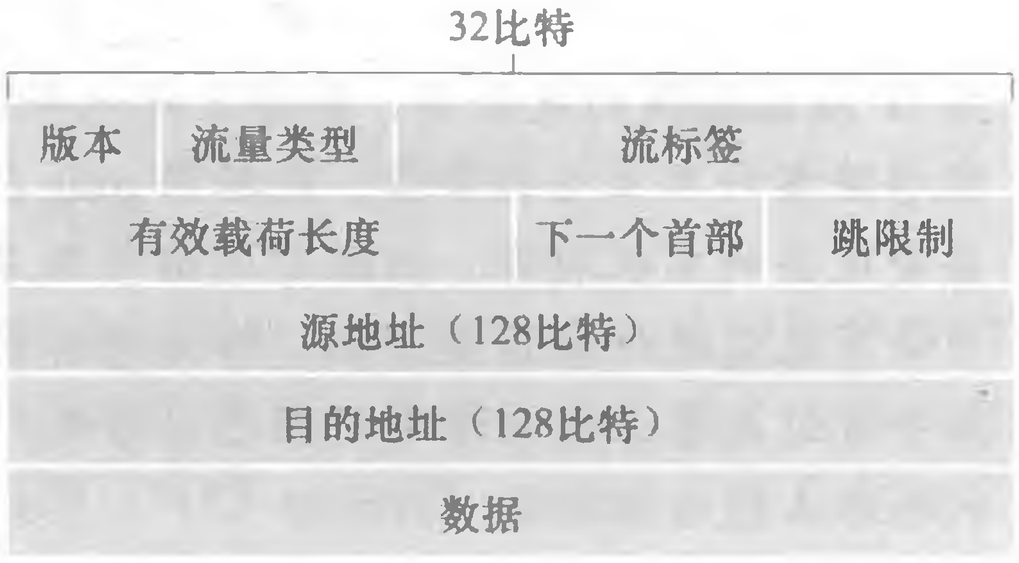

图 4-26　IPv6 数据报格式

IPv6 中引入的最重要的变化显示在其数据报格式中：

- 扩大的地址容量。IPv6 将 IP 地址长度从 32 比特增加到 128 比特。这就确保全世界将不会用尽 IP 地址。现在，地球上的每个沙砾都可以用 IP 地址寻址了。除了单播与多播地址以外，IPv6 还引入了一种称为**任播地址**（anycast address）的新型地址，这种地址可以使数据报交付给一组主机中的任意一个。（例如，这种特性可用于向一组包含给定文档的镜像站点中的最近一个发送 HTTP GET 报文。）
- 简化高效的 40 字节首部。如下面讨论的那样，许多 IPv4 字段已被舍弃或作为选项。因而所形成的 40 字节定长首部允许路由器更快地处理 IP 数据报。一种新的选项编码允许进行更灵活的选项处理。
- 流标签。IPv6 有一个难以捉摸的**流**（flow）定义。RFC 2460 中描述道，该字段可用于“给属于特殊流的分组加上标签，这些特殊流是发送方要求进行特殊处理的流，如一种非默认服务质量或需要实时服务的流”。例如，音频与视频传输就可能被当作一个流。另一方面，更为传统的应用（如文件传输和电子邮件）就不可能被当作流。由高优先权用户（如某些为使其流量得到更好服务而付费的用户）承载的流量也有可能被当作一个流。然而，IPv6 的设计者显然已预见到最终需要能够区分这些流，即使流的确切含义还未完全确定。

如上所述，比较图 4-26 与图 4-16 就可看出，IPv6 数据报的结构更简单、更高效。以下是在 IPv6 中定义的字段。

- 版本。该 4 比特字段用于标识 IP 版本号。毫不奇怪，IPv6 将该字段值设为 6。注意到将该字段值置为 4 并不能创建一个合法的 IPv4 数据报。（如果这样的话，事情就简单多了，参见下面有关从 IPv4 向 IPv6 迁移的讨论。）
- 流量类型。该 8 比特字段与我们在 IPv4 中看到的 TOS 字段的含义相似。
- 流标签。如上面讨论过的那样，该 20 比特的字段用于标识一条数据报的流，能够

对一条流中的某些数据报给出优先权，或者它能够用来对来自某些应用（例如 IP 话音）的数据报给出更高的优先权，以优于来自其他应用（例如 SMTP 电子邮件）的数据报。

- 有效载荷长度。该 16 比特值作为一个无符号整数，给出了 IPv6 数据报中跟在定长的 40 字节数据报首部后面的字节数量。
- 下一个首部。该字段标识数据报中的内容（数据字段）需要交付给哪个协议（如 TCP 或 UDP）。该字段使用与 IPv4 首部中协议字段相同的值。
- 跳限制。转发数据报的每台路由器将对该字段的内容减 1。如果跳限制计数达到 0，则该数据报将被丢弃。
- 源地址和目的地址。IPv6 128 比特地址的各种格式在 RFC 4291 中进行了描述。
- 数据。这是 IPv6 数据报的有效载荷部分。当数据报到达目的地时，该有效载荷就从 IP 数据报中移出，并交给在下一个首部字段中指定的协议处理。

以上讨论说明了 IPv6 数据报中包括的各字段的用途。将图 4-26 中的 IPv6 数据报格式与图 4-16 中的 IPv4 数据报格式进行比较，我们就会注意到，在 IPv4 数据报中出现的几个字段在 IPv6 数据报中已不复存在：

- 分片/重新组装。IPv6 不允许在中间路由器上进行分片与重新组装。这种操作只能在源与目的地执行。如果路由器收到的 IPv6 数据报因太大而不能转发到出链路上的话，则路由器只需丢掉该数据报，并向发送方发回一个“分组太大”的 ICMP 差错报文即可（见 5.6 节）。于是发送方能够使用较小长度的 IP 数据报重发数据。分片与重新组装是一个耗时的操作，将该功能从路由器中删除并放到端系统中，大大加快了网络中的 IP 转发速度。
- 首部检验和。因为因特网层中的运输层（如 TCP 与 UDP）和数据链路层（如以太网）协议执行了检验操作，IP 设计者大概觉得在网络层中具有该项功能实属多余，所以将其去除。再次强调的是，快速处理 IP 分组是关注的重点。在 4.3.1 节中我们讨论 IPv4 时讲过，由于 IPv4 首部中包含有一个TTL字段（类似于 IPv6 中的跳限制字段），所以在每台路由器上都需要重新计算 IPv4 首部检验和。就像分片与重新组装一样，在 IPv4 中这也是一项耗时的操作。
- 选项。选项字段不再是标准 IP 首部的一部分了。但它并没有消失，而是可能出现在 IPv6 首部中由“下一个首部”指出的位置上。这就是说，就像 TCP 或 UDP 协议首部能够是 IP 分组中的“下一个首部”一样，选项字段也能是“下一个首部”。删除选项字段使得 IP 首部成为定长的 40 字节。

2. 从 IPv4 到 IPv6 的迁移

既然我们已了解了 IPv6 的技术细节，那么我们考虑一个非常实际的问题：基于 IPv4 的公共因特网如何迁移到 IPv6 呢？问题是，虽然新型 IPv6 使能系统可做成向后兼容，即能发送、路由和接收 IPv4 数据报，但已部署的具有 IPv4 能力的系统却不能够处理 IPv6 数据报。可以采用以下几种方法［Huston 2011b；RFC 4213］。

一种可选的方法是宣布一个标志日，即指定某个日期和时间，届时因特网的所有机器都关机并从 IPv4 升级到 IPv6。上次重大的技术迁移（为得到可靠的运输服务，从使用 NCP 迁移到使用 TCP）出现在差不多 35 年以前。即使回到那时［RFC 801］——因特网很小且仍然由少数“奇才”管理着，人们也会认识到选择这样一个标志日是不可行的。一

个涉及数十亿台机器的标志日现在更是不可想象的。

在实践中已经得到广泛采用的 IPv4 到 IPv6 迁移的方法包括**建隧道**（tunneling）[RFC 4213]。除了 IPv4 到 IPv6 迁移之外的许多其他场合的应用都具有建隧道的关键概念，包括在第 7 章将涉及的全 IP 蜂窝网络中也得到广泛使用。建隧道依据的基本思想如下：假定两个 IPv6 节点（如图 4-27 中的 B 和 E）要使用 IPv6 数据报进行交互，但它们是经由中间 IPv4 路由器互联的。我们将两台 IPv6 路由器之间的中间 IPv4 路由器的集合称为一个**隧道**（tunnel），如图 4-27 所示。借助于隧道，在隧道发送端的 IPv6 节点（如 B）可将整个 IPv6 数据报放到一个 IPv4 数据报的数据（有效载荷）字段中。于是，该 IPv4 数据报的地址设为指向隧道接收端的 IPv6 节点（在此例中为 E），再发送给隧道中的第一个节点（在此例中为 C）。隧道中的中间 IPv4 路由器在它们之间为该数据报提供路由，就像对待其他数据报一样，完全不知道该 IPv4 数据报自身就含有一个完整的 IPv6 数据报。隧道接收端的 IPv6 节点最终收到该 IPv4 数据报（它是该 IPv4 数据报的目的地），并确定该 IPv4 数据报含有一个 IPv6 数据报（通过观察在 IPv4 数据报中的协议号字段是 41［RFC 4213］，指示该 IPv4 有效载荷是 IPv6 数据报），从中取出 IPv6 数据报，然后再为该 IPv6 数据报提供路由，就好像它是从一个直接相连的 IPv6 邻居那里接收到该 IPv6 数据报一样。

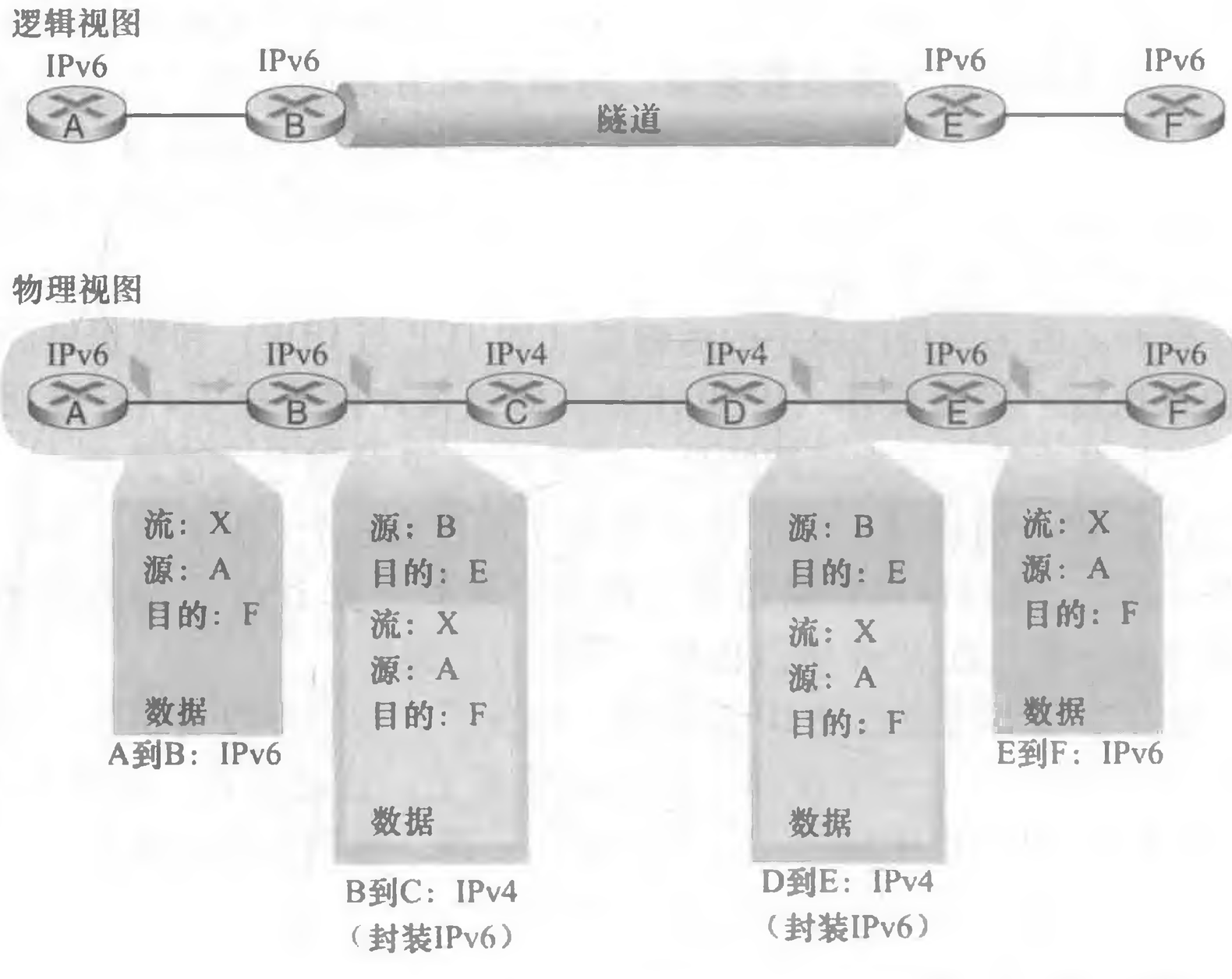

图 4-27　建隧道

在结束本节前需要说明的是，尽管采用 IPv6 最初表现为一个缓慢启动的过程［Lawton 2001；Huston 2008b］，但势头已经有了。NIST［NIST IPv6 2015］报告称，超过三分之一的美国政府二级域名是支持 IPv6 的。在客户端，谷歌报告称访问谷歌服务的客户仅有 8% 使用了 IPv6［Google IPv6 2015］。但其他最近统计结果指出［Czyz 2014］，IPv6 的采用正在加速。诸如 IP 使能的电话和其他便携式设备的激增，为 IPv6 的更广泛部署提供了新的推动力。欧洲的第三代合作计划［3GPP 2016］已规定了 IPv6 为移动多媒体的标准编址方案。

我们能从 IPv6 经验中学到的重要一课是，要改变网络层协议是极其困难的。自从 20 世纪 90 年代早期以来，有许多新的网络层协议被鼓吹为因特网的下一次重大革命，但这些协议中的大多数至今为止只取得了有限突破。这些协议包括 IPv6、多播协议、资源预留协议，其中后面两个协议的讨论可在本书的在线补充材料中找到。在网络层中引入新的协议的确如同替换一幢房子的基石，即在不拆掉整幢房子（或至少临时重新安置房屋住户）的情况下是很难完成上述工作的。另一方面，因特网却已见证了在应用层中新协议的快速部署。典型的例子当然有 Web、即时讯息、流媒体、分布式游戏和各种形式的社交媒体。引入新的应用层协议就像给一幢房子重新刷一层漆，这是相对容易做的事，如果你选择了一个好看的颜色，邻居将会照搬你的选择。总之，未来我们肯定会看到因特网网络层发生改变，但这种改变将比应用层慢得多。

4.4　通用转发和 SDN

在 4.2.1 节中，我们注意到因特网路由器的转发决定传统上仅仅基于分组的目的地址。然而，在前一节中，我们也已经看到执行许多第三层功能的中间盒有了大量发展。NAT 盒重写首部 IP 地址和端口号；防火墙基于首部字段值阻拦流量或重定向分组以进行其他处理，如深度分组检测（DPI）。负载均衡器将请求某种给定服务（例如一个 HTTP 请求）的分组转发到提供该服务的服务器集合中的一个。[RFC 3234] 列出了许多常用中间盒功能。

第二层交换机和第三层路由器等中间盒 [Qazi 2013] 的剧增，而且每种都有自己特殊的硬件、软件和管理界面，无疑给许多网络操作员带来了十分头疼的大麻烦。然而，近期软件定义网络的进展已经预示并且正在提出一种统一的方法，以一种现代、简洁和综合方式，提供多种网络层功能以及某些链路层功能。

回顾 4.2.1 节将基于目的地转发的特征总结为两个步骤：查找目的 IP 地址（“匹配”），然后将分组发送到有特定输出端口的交换结构（“动作”）。我们现在考虑一种更有意义的通用“匹配加动作”范式，其中能够对协议栈的多个首部字段进行“匹配”，这些首部字段是与不同层次的不同协议相关联的。“动作”能够包括：将分组转发到一个或多个输出端口（就像在基于目的地转发中一样），跨越多个通向服务的离开接口进行负载均衡分组（就像在负载均衡中一样），重写首部值（就像在 NAT 中一样），有意识地阻挡/丢弃某个分组（就像在防火墙中一样），为进一步处理和动作而向某个特定的服务器发送一个分组（就像在 DPI 一样），等等。

在通用转发中，一张匹配加动作表将我们在 4.2.1 节中看到的基于目的地的转发表一般化了。因为能够使用网络层和/或链路层源和目的地址做出转发决定，所以显示在图 4-28 中的转发设备更为准确地描述为“分组交换机”而不是第三层“路由器”或第二层“交换机”。因此，在本节后面部分以及 5.5 节中，我们将这些设备称为分组交换机，这是在 SDN 文献中被广泛采用的术语。

图 4-28 显示了位于每台分组交换机中的一张匹配加动作表，该表由远程控制器计算、安装和更新。我们注意到虽然在各台分组交换机中的控制组件可以相互作用（例如以类似于图 4-2 中的方式），但实践中通用匹配加动作能力是通过计算、安装和更新这些表的远程控制器实现的。花几分钟比较图 4-2、图 4-3 和图 4-28，你能看出图 4-2 和图 4-3 中显示的基于目的地转发与图 4-28 中显示的通用转发有什么相似和差异吗？

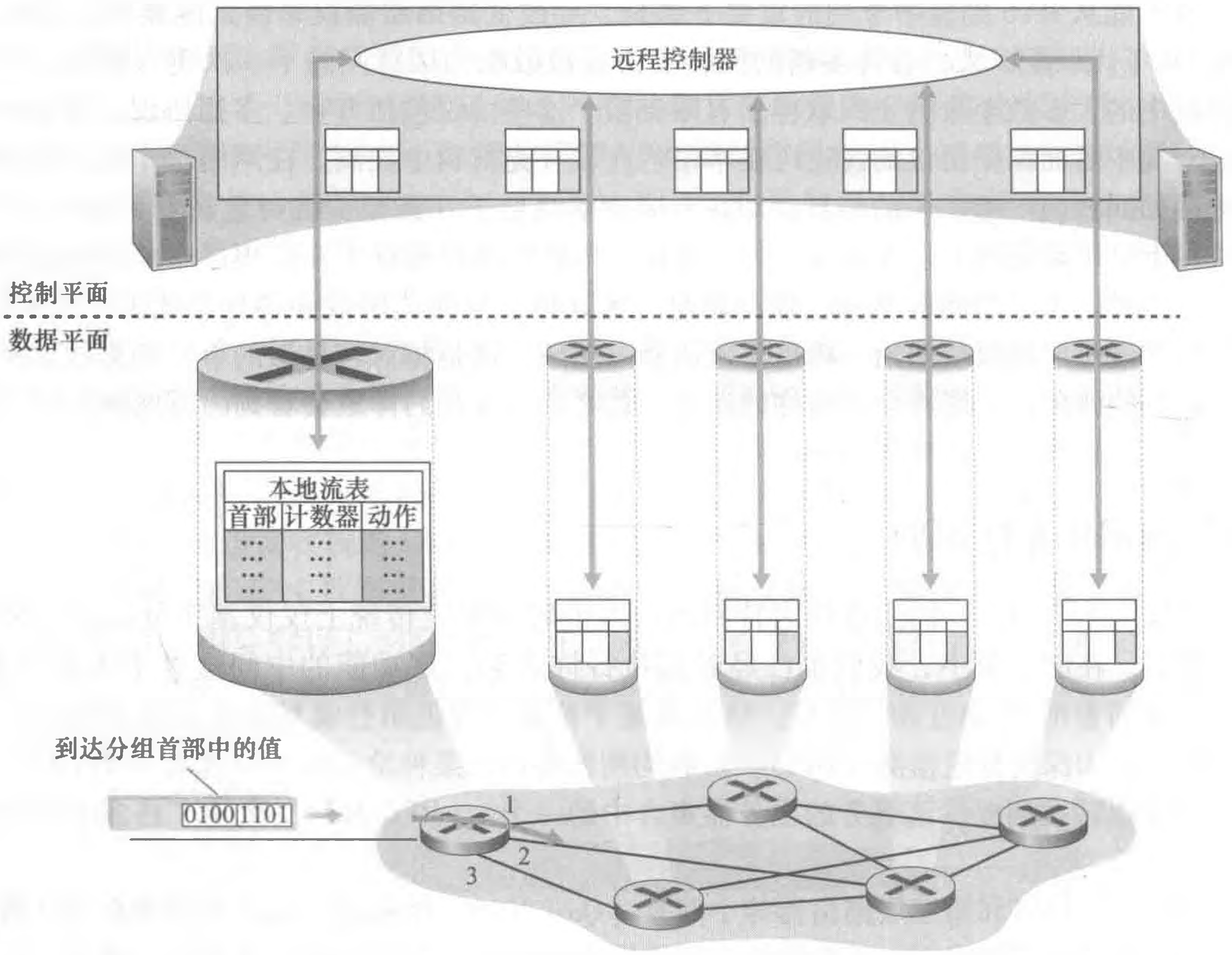

图4-28　通用转发：每台分组交换机包含一张匹配加动作表，该表是由远程控制器计算和分发的

我们后续对通用转发的讨论将基于 OpenFlow［McKeown 2008；OpenFlow 2009；Casado 2014；Tourrilhes 2014］，OpenFlow 是一个得到高度认可和成功的标准，它已经成为匹配加动作转发抽象、控制器以及更为一般的 SDN 革命等概念的先驱［Feamster 2013］。我们将主要考虑 OpenFlow 1.0，该标准以特别清晰和简明的方式引入了关键的 SDN 抽象和功能。OpenFlow 的后继版本根据实现和使用获得的经验引入了其他能力；OpenFlow 标准的当前和早期版本能在［ONF 2016］中找到。

匹配加动作转发表在 OpenFlow 中称为**流表**（flow table），它的每个表项包括：

- 首部字段值的集合，入分组将与之匹配。与基于目的地转发的情况一样，基于硬件匹配在 TCAM 内存中执行得最为迅速（TCAM 内存中可能有上百万条地址表项）［Bosshart 2013］。匹配不上流表项的分组将被丢弃或发送到远程控制器做更多处理。在实践中，为了性能或成本原因，一个流表可以由多个流表实现［Bosshart 2013］，但我们这里只关注单一流表的抽象。
- 计数器集合（当分组与流表项匹配时更新计数器）。这些计数器可以包括已经与该表项匹配的分组数量，以及自从该表项上次更新以来的时间。
- 当分组匹配流表项时所采取的动作集合。这些动作可能将分组转发到给定的输出端口，丢弃该分组、复制该分组和将它们发送到多个输出端口，和/或重写所选的首部字段。

我们将在 4.4.1 节和 4.4.2 节中分别更为详细地探讨匹配和动作。我们将学习每台分组交换机网络范围的匹配规则集合是如何用来实现多种多样的功能的，包括 4.4.3 节中的

路由选择、第二层交换路由、防火墙、负载均衡、虚拟网络等等。在结束时，我们注意到流表本质上是一个 API，通过这种抽象每台分组交换机的行为能被编程；我们将在 4.4.3 节中看到，通过在网络分组交换机的集合中适当地编程/配置这些表，网络范围的行为能被类似地编程［Casado 2014］。

4.4.1　匹配

图 4-29 显示了 11 个分组首部字段和入端口 ID，该 ID 能被 OpenFlow 1.0 中的匹配加动作规则所匹配。前面 1.5.2 节讲过，到达一台分组交换机的一个链路层（第二层）帧将包含一个网络层（第三层）数据报作为其有效载荷，该载荷通常依次将包含一个运输层（第四层）报文段。第一个观察是，OpenFlow 的匹配抽象允许对来自三个层次的协议首部所选择的字段进行匹配（因此相当勇敢地违反了我们在 1.5 节中学习的分层原则）。因为我们还没有涉及链路层，用如下的说法也就足够了：显示在图 4-29 中的源和目的 MAC 地址是与帧的发送和接收接口相关联的链路层地址；通过基于以太网地址而不是 IP 地址进行转发，我们看到 OpenFlow 使能的设备能够等价于路由器（第三层设备）转发数据报以及交换机（第二层设备）转发帧。以太网类型字段对应于较高层协议（例如 IP），利用该字段分解该帧的载荷，并且 VLAN 字段与所谓虚拟局域网相关联，我们将在第 6 章中学习 VLAN。OpenFlow 1.0 规范中匹配的 12 个值在最近的 OpenFlow 规范中已经增加到 41 个［Bosshart 2014］。

图 4-29　OpenFlow 1.0 流表的分组匹配字段

入端口是指分组交换机上接收分组的输入端口。在 4.3.1 节中，我们已经讨论过该分组的 IP 源地址、IP 目的地址、IP 协议字段和 IP 服务类型字段。运输层源和目的端口号字段也能匹配。

流表项也可以有通配符。例如，在一个流表中 IP 地址 128.119.*.* 将匹配其地址的前 16 比特为 128.119 的任何数据报所对应的地址字段。每个流表项也具有相应的优先权。如果一个分组匹配多个流表项，选定的匹配和对应的动作将是其中有最高优先权的那个。

最后，我们观察到并非一个 IP 首部中的所有字段都能被匹配。例如 OpenFlow 并不允许基于 TTL 字段或数据报长度字段的匹配。为什么有些字段允许匹配，而有些字段不允许呢？毫无疑问，与功能和复杂性有关。选择一种抽象的“艺术”是提供足够的功能来完成某种任务（在这种情况下是实现、配置和管理宽泛的网络层功能，以前这些一直是通过各种各样的网络层设备来实现的），不必用如此详尽和一般性的“超负荷”抽象，这种抽象已经变得臃肿和不可用。Butler Lampson 有过著名的论述［Lampson 1983］：

> 在一个时刻做一件事，将它做好。一个接口应当俘获一个抽象的最低限度的要件。不要进行一般化，一般化通常是错误的。

考虑到 OpenFlow 的成功，人们能够推测它的设计者的确很好地选择了抽象技术。OpenFlow 匹配的更多细节能够在［OpenFlow 2009；ONF 2016］中找到。

4.4.2　动作

如图4-28中所见，每个流表项都有零个或多个动作列表，这些动作决定了应用于与流表项匹配的分组的处理。如果有多个动作，它们以在表中规定的次序执行。

其中最为重要的动作可能是：

- 转发。一个入分组可以转发到一个特定的物理输出端口，广播到所有端口（分组到达的端口除外），或通过所选的端口集合进行多播。该分组可能被封装并发送到用于该设备的远程控制器。该控制器则可能（或可能不）对该分组采取某些动作，包括安装新的流表项，以及可能将该分组返回给该设备以在更新的流表规则集合下进行转发。
- 丢弃。没有动作的流表项表明某个匹配的分组应当被丢弃。
- 修改字段。在分组被转发到所选的输出端口之前，分组首部10个字段（图4-29中显示的除IP协议字段外的所有第二、三、四层的字段）中的值可以重写。

4.4.3　匹配加动作操作中的OpenFlow例子

在已经考虑了通用转发的匹配和动作组件后，我们在图4-30显示的样本网络场景中将这些想法拼装在一起。该网络具有6台主机（h1、h2、h3、h4、h5和h6）以及3台分组交换机（s1、s2和s3），每台交换机具有4个本地接口（编号1到4）。我们将考虑一些希望实现的网络范围的行为，在s1、s2和s3中的流表项需要实现这种行为。

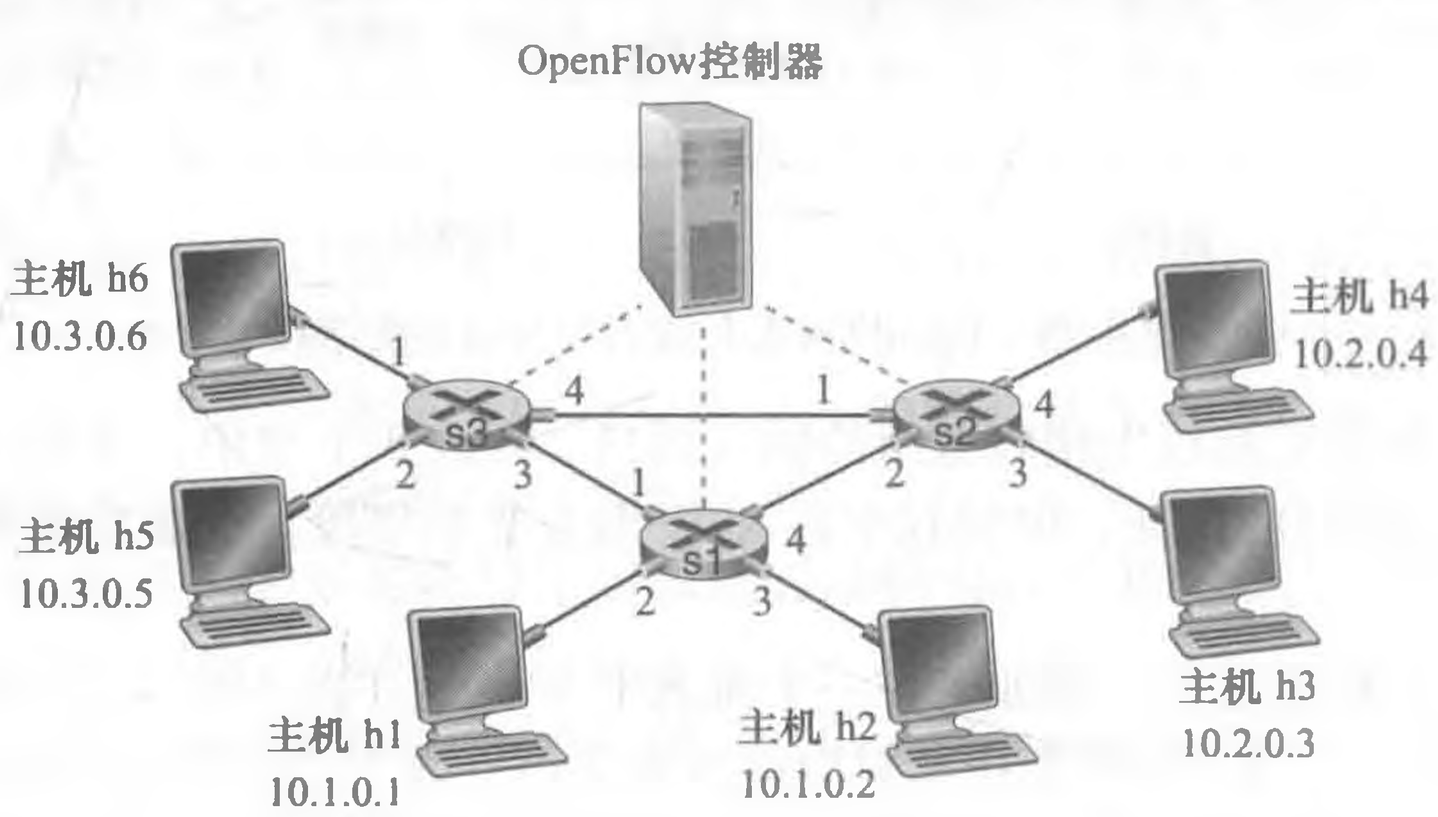

图4-30　具有3台分组交换机、6台主机和1台OpenFlow控制器的OpenFlow匹配加动作网络

第一个例子：简单转发

作为一个非常简单的例子，假定希望的转发行为是：来自h5或h6发往h3或h4的分组从s3转发到s1，然后从s1转发到s2（完全避免使用s3和s2之间的链路）。在s1中的流表项将是：

s1流表（例1）

匹配	动作
Ingress Port = 1；IP Src = 10.3.*.*；IP Dst = 10.2.*.*	Forward（4）
……	……

当然，我们也需要在 s3 中有一个流表项，使得该数据报从 h5 或 h6 经过出接口 3 转发到 s1：

s3 流表（例 1）	
匹配	动作
IP Src = 10. 3. ＊. ＊; IP Dst = 10. 2. ＊. ＊	Forward（3）
……	……

最后，我们也需要在 s2 中有一个流表项来完成第一个例子，使得从 s1 到达的数据报转发到它们的目的主机 h3 或 h4。

s2 流表（例 1）	
匹配	动作
Ingress Port = 2; IP Dst = 10. 2. 0. 3	Forward（3）
Ingress Port = 2; IP Dst = 10. 2. 0. 4	Forward（4）
……	……

第二个例子：负载均衡

作为第二个例子，我们考虑一个负载均衡的场景，其中来自 h3 发往 10. 1. ＊. ＊的数据报经过 s1 和 s2 之间的直接链路转发，与此同时来自 h4 发往 10. 1. ＊. ＊的数据报经过 s2 和 s3（于是从 s3 到 s1）之间的链路转发。注意到这种行为不能通过基于 IP 的目的地转发取得。在这种情况下，在 s2 中的流表项将是：

s2 流表（例 2）	
匹配	动作
Ingress Port = 3; IP Dst = 10. 1. ＊. ＊	Forward（2）
Ingress Port = 4; IP Dst = 10. 1. ＊. ＊	Forward（1）
……	……

在 s1 中需要流表项将从 s2 收到的数据报转发到 h1 或 h2；在 s3 中需要流表项将接口 4 上从 s2 收到的数据报经过接口 3 转发到 s1。考虑是否能在 s1 和 s3 中配置这些流表项。

第三个例子：充当防火墙

作为第三个例子，我们考虑一个防火墙场景，其中 s2 仅希望（在它的任何接口上）接收来自与 s3 相连的主机所发送的流量。

s2 流表（例 3）	
匹配	动作
IP Src = 10. 3. ＊. ＊; IP Dst = 10. 2. 0. 3	Forward（3）
IP Src = 10. 3. ＊. ＊; IP Dst = 10. 2. 0. 4	Forward（4）
……	……

如果在 s2 的流表中没有其他表项，则仅有来自 10. 3. ＊. ＊的流量将被转发到与 s2 相连的主机。

尽管我们这里仅考虑了几种基本场景，但通用转发的多样性和优势显而易见。在课后习题中，我们将探讨流表如何用来生成许多不同的逻辑行为，包括使用相同分组交换机和

链路物理集合的虚拟网络，即两个或多个逻辑上分离的网络（每个网络有它们自己的独立和截然不同的转发行为）。在5.5节中，当学习SDN控制器时，我们将再次考察流表，其中SDN控制器计算和分发流表，协议用于在分组交换机和它的控制器之间进行通信。

4.5 小结

在本章中，我们讨论了网络层的**数据平面**（data plane）功能，即每台路由器的如下功能：决定到达路由器的输入链路之一的分组如何转发到该路由器的输出链路之一。

我们从仔细观察路由器的内部操作开始，学习输入和输出端口功能，以及基于目的地的转发、路由器的内部交换机制、分组排队管理等等。我们涉及传统的IP转发（其中转发基于数据报的目的地址进行）和通用转发（其中转发和其他功能可以使用数据报首部中的几个不同的字段值来进行），并且看到了后一种方法的多种用途。我们还详细地学习了IPv4和IPv6协议以及因特网编址，并对此有了更深入、更敏锐和更有趣的发现。

借助于新得到的对网络层数据平面的理解，我们现在准备着手学习第5章中的网络层控制平面。

课后习题和问题

复习题

4.1节

R1. 我们回顾在本书中使用的某些术语。前面讲过运输层的分组名字是报文段，数据链路层的分组名字是帧。网络层的分组名字是什么？前面讲过路由器和链路层交换机都被称为分组交换机。路由器与链路层交换机间的根本区别是什么？

R2. 我们注意到网络层功能可被大体分成数据平面功能和控制平面功能。数据平面的主要功能是什么？控制平面的主要功能呢？

R3. 我们对网络层执行的转发功能和路由选择功能进行区别。路由选择和转发的主要区别是什么？

R4. 路由器中转发表的主要作用是什么？

R5. 我们说过网络层的服务模型“定义发送主机和接收主机之间端到端分组的传送特性”。因特网的网络层的服务模型是什么？就主机到主机数据报的传递而论，因特网的服务模型能够保证什么？

4.2节

R6. 在4.2节中，我们看到路由器通常由输入端口、输出端口、交换结构和路由选择处理器组成。其中哪些是用硬件实现的，哪些是用软件实现的？为什么？转到网络层的数据平面和控制平面的概念，哪些是用硬件实现的，哪些是用软件实现的？为什么？

R7. 讨论为什么在高速路由器的每个输入端口都存储转发表的影子副本。

R8. 基于目的地转发意味着什么？这与通用转发有什么不同（假定你已经阅读4.4节，两种方法中哪种是软件定义网络所采用的）？

R9. 假设一个到达分组匹配了路由器转发表中的两个或更多表项。采用传统的基于目的地转发，路由器用什么原则来确定这条规则可以用于确定输出端口，使得到达的分组能交换到输出端口？

R10. 在4.2节中讨论了三种交换结构。列出并简要讨论每一种交换结构。哪一种（如果有的话）能够跨越交换结构并行发送多个分组？

R11. 描述在输入端口会出现分组丢失的原因。描述在输入端口如何消除分组丢失（不使用无限大缓存区）。

R12. 描述在输出端口会出现分组丢失的原因。通过提高交换结构速率，能够防止这种丢失吗？

R13. 什么是HOL阻塞？它出现在输入端口还是输出端口？

R14. 在 4.2 节我们学习了 FIFO、优先权、循环（RR）和加权公平排队（WFQ）分组调度规则。这些排队规则中，哪个规则确保所有分组是以到达的次序离开的？

R15. 举例说明为什么网络操作员要让一类分组的优先权超过另一类分组的。

R16. RR 和 WFQ 分组调度之间的基本差异是什么？存在 RR 和 WFQ 将表现得完全相同的场合吗？（提示：考虑 WFQ 权重。）

4.3 节

R17. 假定主机 A 向主机 B 发送封装在一个 IP 数据报中的 TCP 报文段。当主机 B 接收到该数据报时，主机 B 中的网络层怎样知道它应当将该报文段（即数据报的有效载荷）交给 TCP 而不是 UDP 或某个其他东西呢？

R18. 在 IP 首部中，哪个字段能用来确保一个分组的转发不超过 N 台路由器？

R19. 前面讲过因特网检验和被用于运输层报文段（分别在图 3-7 和图 3-29 的 UDP 和 TCP 首部中）以及网络层数据报（图 4-16 的 IP 首部中）。现在考虑一个运输层报文段封装在一个 IP 数据报中。在报文段首部和数据报首部中的检验和要遍及 IP 数据报中的任何共同字节进行计算吗？

R20. 什么时候一个大数据报分割成多个较小的数据报？较小的数据报在什么地方装配成一个较大的数据报？

R21. 路由器有 IP 地址吗？如果有，有多少个？

R22. IP 地址 223.1.3.27 的 32 比特二进制等价形式是什么？

R23. 考察使用 DHCP 的主机，获取它的 IP 地址、网络掩码、默认路由器及其本地 DNS 服务器的 IP 地址。列出这些值。

R24. 假设在一个源主机和一个目的主机之间有 3 台路由器。不考虑分片，一个从源主机发送给目的主机的 IP 数据报将通过多少个接口？为了将数据报从源移动到目的地需要检索多少个转发表？

R25. 假设某应用每 20ms 生成一个 40 字节的数据块，每块封装在一个 TCP 报文段中，TCP 报文段再封装在一个 IP 数据报中。每个数据报的开销有多大？应用数据所占百分比是多少？

R26. 假定你购买了一个无线路由器并将其与电缆调制解调器相连。同时假定 ISP 动态地为你连接的设备（即你的无线路由器）分配一个 IP 地址。还假定你家有 5 台 PC，均使用 802.11 以无线方式与该无线路由器相连。怎样为这 5 台 PC 分配 IP 地址？该无线路由器使用 NAT 吗？为什么？

R27. “路由聚合”一词意味着什么？路由器执行路由聚合为什么是有用的？

R28. “即插即用”或“零配置”协议意味着什么？

R29. 什么是专用网络地址？具有专用网络地址的数据报会出现在大型公共因特网中吗？解释理由。

R30. 比较并对照 IPv4 和 IPv6 首部字段。它们有相同的字段吗？

R31. 有人说当 IPv6 以隧道形式通过 IPv4 路由器时，IPv6 将 IPv4 隧道作为链路层协议。你同意这种说法吗？为什么？

4.4 节

R32. 通用转发与基于目的地转发有何不同？

R33. 我们在 4.1 节遇到的基于目的地转发与在 4.4 节遇到的 OpenFlow 流表之间有什么差异？

R34. 路由器或交换机的“匹配加动作”意味着什么？在基于目的地转发的分组交换机场合中，要匹配什么并采取什么动作？在 SDN 的场合中，举出 3 个能够被匹配的字段和 3 个能被采取的动作。

R35. 在 IP 数据报中举出能够在 OpenFlow 1.0 通用转发中“匹配”的 3 个首部字段。不能在 OpenFlow 中“匹配”的 3 个 IP 数据报首部字段是什么？

习题

P1. 考虑下面的网络。

a. 显示路由器 A 中的转发表，使得目的地为主机 H3 的所有流量都通过接口 3 转发。

b. 写出路由器 A 中的转发表，使得从 H1 发往主机 H3 的所有流量都通过接口 3 转发，从 H2 发往主机 H3 的所有流量都通过接口 4 转发。（提示：这是一个技巧性的问题。）