- 源主机从目标主机接收到一个 TCP RST 报文段。这意味着该 SYN 报文段到达了目标主机，但目标主机没有运行一个使用 TCP 端口 6789 的应用程序。但攻击者至少知道发向该主机端口 6789 的报文段没有被源和目标主机之间的任何防火墙所阻挡。（将在第 8 章中讨论防火墙。）
- 源什么也没有收到。这很可能表明该 SYN 报文段被中间的防火墙所阻挡，无法到达目标主机。

nmap 是一个功能强大的工具，该工具不仅能“侦察”打开的 TCP 端口，也能“侦察”打开的 UDP 端口，还能“侦察”防火墙及其配置，甚至能“侦察”应用程序的版本和操作系统。其中的大多数都能通过操作 TCP 连接管理报文段完成［Skoudis 2006］。读者能够从 http://www.nmap.org 下载 nmap。

到此，我们介绍完了 TCP 中的差错控制和流量控制。在 3.7 节中，我们将回到 TCP 并更深入地研究 TCP 拥塞控制问题。然而，在此之前，我们先后退一步，在更广泛环境中讨论拥塞控制问题。

3.6 拥塞控制原理

在前面几节中，我们已经分析了面临分组丢失时用于提供可靠数据传输服务的基本原理及特定的 TCP 机制。我们以前讲过，在实践中，这种丢包一般是当网络变得拥塞时由于路由器缓存溢出引起的。分组重传因此作为网络拥塞的征兆（某个特定的运输层报文段的丢失）来对待，但是却无法处理导致网络拥塞的原因，因为有太多的源想以过高的速率发送数据。为了处理网络拥塞原因，需要一些机制以在面临网络拥塞时遏制发送方。

在本节中，我们考虑一般情况下的拥塞控制问题，试图理解为什么网络拥塞是一件坏事情，网络拥塞是如何在上层应用得到的服务性能中明确地显露出来的？如何可用各种方法来避免网络拥塞或对它作出反应？这种对拥塞控制的更一般研究是恰当的，因为就像可靠数据传输一样，它在组网技术中的前 10 个基础性重要问题清单中位居前列。我们通过对**异步传递方式**（ATM）网络中**可用比特率**（ABR）服务中的拥塞控制的讨论来总结本节。下面一节包含了 TCP 的拥塞控制算法的详细研究。

3.6.1 拥塞原因与代价

我们通过分析 3 个复杂性越来越高的发生拥塞的情况，开始对拥塞控制的一般性研究。在每种情况下，我们首先将看看出现拥塞的原因以及拥塞的代价（根据资源未被充分利用以及端系统得到的低劣服务性能来评价）。我们暂不关注如何对拥塞作出反应或避免拥塞，而是重点理解一个较为简单的问题，即随着主机增加其发送速率并使网络变得拥塞，这时会发生的情况。

1. 情况 1：两个发送方和一台具有无穷大缓存的路由器

我们先考虑也许是最简单的拥塞情况：两台主机（A 和 B）都有一条连接，且这两条连接共享源与目的地之间的单跳路由，如图 3-43 所示。

我们假设主机 A 中的应用程序以 λ_{in} 字节/秒的平均速率将数据发送到连接中（例如，通过一个套接字将数据传递给运输层协议）。这些数据是初始数据，这意味着每个数据单元仅向套接字中发送一次。下面的运输层协议是一个简单的协议。数据被封装并发送；不