已在第2章看到，服务器进程在周知端口号上等待入请求。）第二，他们认为路由器通常仅应当处理高达第三层的分组。第三，他们认为NAT协议违反了所谓端到端原则，即主机彼此应相互直接对话，结点不应介入修改IP地址与端口号。第四，他们认为应使用IPv6（参见4.4.4节）来解决IP地址短缺问题，而不是不计后果地用一种如NAT之类的权宜之计来修补存在的问题。但不管喜欢与否，NAT已成为因特网的一个重要组件。

NAT的另一个重要问题是它妨碍P2P应用程序，包括P2P文件共享应用和P2PIP语音应用。第2章讲过在一个P2P应用程序中，任何参与对等方A应当能够对任何其他参与对等方B发起一条TCP连接。该问题的实质在于如果对等方B在一个NAT后面，它不能充当服务器并接收TCP连接。如我们将在课后习题中所见，如果对等方A不在一个NAT的后面，则该NAT问题能够绕过去。在这种情况下，对等方A能够首先通过一个中间对等方C与对等方B联系，其中C不位于NAT之后并与B已经创建了一条进行中的TCP连接。对等方A则能够经对等方C请求对等方B，发起直接返回对等方A的一条TCP连接。一旦对等方A和B之间创建一条直接的P2P TCP连接，这两个对等方就能够交换报文或文件。这种雇佣关系被称为**连接反转**（connection reversal,，实际上被许多P2P应用程序用于**NAT穿越**（NAT traversal）。如果对等方A和对等方B都在它们自己的NAT后面，这种情况有些棘手，但是能够使用应用程序进行中继处理，正如我们在第2章中对于Skype中继所见到的那样。

4. UPnP

NAT穿越正越来越多地由通用即插即用（UPnP）提供，UPnP是一种允许主机发现并配置邻近NAT的协议［UPnP Forum 2012］。UPnP要求主机和NAT都是UPnP兼容的。使用UPnP，在主机上运行的应用程序能够为某些请求的公共端口号请求一个NAT映射，该映射位于其（专用IP地址，专用端口号）和（公共IP地址，公共端口号）之间。如果某NAT接受该请求并生成映射，则来自外部的结点能够发起到（公共IP地址，公共端口号）的TCP连接。此外，UPnP让该应用程序知道（公共IP地址，公共端口号），因此该应用程序能够向外部世界通告它。

举一个例子，假定你的主机位于一个UPnP使能的NAT背后，具有专用地址10.0.0.1并且在端口3345上运行BitTorrent。另外假定该NAT的公共IP地址是138.76.29.7。你的BitTorrent应用程序自然要能够接受来自其他主机的连接，使得它能够同其他主机对换块。此后，你主机上的BitTorrent应用程序请求NAT产生一个“洞”，将（10.0.0.1，3345）映射到（138.76.29.7，5001）。（该应用程序选择了公共端口号5001。）在你主机中的BitTorrent应用程序也能够向它的追踪器通告它在（138.76.29.7，5001）可供使用。以这种方式，运行BitTorrent的一台外部主机能够联系该追踪器，并知道你的BitTorrent应用程序正运行在（138.76.29.7，5001）。该外部主机能够向（138.76.29.7，5001）发送TCP SYN分组。当NAT接收到该SYN分组，它将改变分组中的目的IP地址和端口号，并通过NAT转发分组。

总而言之，UPnP允许外部主机使用TCP或UDP向NAT化的主机发起通信会话。长期以来NAT一直对P2P应用程序十分不利；UPnP由于提供了有效和健壮的NAT穿越解决方案，可能成为了P2P应用程序的救世主。这里我们对NAT和UPnP的讨论十分简要，对于NAT更为详细的讨论参见［Huston 2004；Cisco NAT 2012］。