

层，完成了人和行李的离港登机口到到港登机口的转移。在起飞/着陆层，完成了一个人及其行李的跑道到跑道的转移。每个层次通过以下方式提供服务：①在这层中执行了某些动作（例如，在登机口层，某飞机的乘客登机和离机）；②使用直接下层的服务（例如，在登机口层，使用起飞/着陆层的跑道到跑道的旅客转移服务）。



图 1-22 航线功能的水平分层

利用分层的体系结构，我们可以讨论一个定义良好的、大而复杂系统的特定部分。这种简化本身由于提供模块化而具有很高价值，这使实现由层所提供的服务易于改变。只要该层对其上面的层提供相同的服务，并且使用来自下面层次的相同服务，当某层的实现变化时，该系统的其余部分保持不变。（值得注意的是，改变一个服务的实现与改变服务本身是极为不同的！）例如，如果登机口功能被改变了（例如让人们按身高登机和离机），航线系统的其余部分将保持不变，因为登机口仍然提供相同的功能（人们登机和离机）；改变后，它仅是以不同的方式实现了该功能。对于大而复杂且需要不断更新的系统，改变服务的实现而不影响该系统其他组件是分层的另一个重要优点。

1. 协议分层

我们对航线已经进行了充分讨论，现将注意力转向网络协议。为了给网络协议的设计提供一个结构，网络设计者以分层（layer）的方式组织协议以及实现这些协议的网络硬件和软件。每个协议属于这些层次之一，就像图 1-22 所示的航线体系结构中的每种功能属于某一层一样。我们再次关注某层向它的上一层提供的服务（service），即所谓一层的服务模型（service model）。就像前面航线例子中的情况一样，每层通过在该层中执行某些动作或使用直接下层的服务来提供服务。例如，由第 n 层提供的服务可能包括报文从网络的一边到另一边的可靠传送。这可能是通过使用第 $n - 1$ 层的边缘到边缘的不可靠报文传送服务，加上第 n 层的检测和重传丢失报文的功能来实现的。

一个协议层能够用软件、硬件或两者的结合来实现。诸如 HTTP 和 SMTP 这样的应用层协议几乎总是在端系统中用软件实现的，运输层协议也是如此。因为物理层和数据链路层负责处理跨越特定链路的通信，它们通常是实现在与给定链路相联系的网络接口卡（例如以太网或 WiFi 接口卡）中。网络层经常是硬件和软件实现的混合体。还要注意的，如同分层的航线体系结构中的功能分布在构成该系统的各机场和飞行控制中心中一样，一个第 n 层协议也分布在构成该网络的端系统、分组交换机和其他组件中。这就是说，第 n 层协议的不同部分常常位于这些网络组件的各部分中。

协议分层具有概念化和结构化的优点 [RFC 3439]。如我们看到的那样，分层提供了