

么我们就知道该分组中已经出现了差错。

你可能想知道为什么 UDP 首先提供了检验和, 就像许多链路层协议 (包括流行的以太网协议) 也提供了差错检测那样。其原因是由于不能保证源和目的之间的所有链路都提供差错检测; 这就是说, 也许这些链路中的一条可能使用没有差错检测的协议。此外, 即使报文段经链路正确地传输, 当报文段存储在某台路由器的内存中时, 也可能引入比特差错。在既无法确保逐链路的可靠性, 又无法确保内存中的差错检测的情况下, 如果端到端数据传输服务要提供差错检测, UDP 就必须在端到端基础上在运输层提供差错检测。这是一个在系统设计中被称颂的端到端原则 (end-end principle) 的例子 [Saltzer 1984], 该原则表述为因为某种功能 (在此时为差错检测) 必须基于端到端实现: “与在较高级别提供这些功能的代价相比, 在较低级别上设置的功能可能是冗余的或几乎没有价值的。”

因为假定 IP 是可以运行在任何第二层协议之上的, 运输层提供差错检测作为一种保险措施是非常有用的。虽然 UDP 提供差错检测, 但它对差错恢复无能为力。UDP 的某种实现只是丢弃受损的报文段; 其他实现是将受损的报文段交给应用程序并给出警告。

至此结束了关于 UDP 的讨论。我们将很快看到 TCP 为应用提供了可靠数据传输及 UDP 所不能提供的其他服务。TCP 自然要比 UDP 复杂得多。然而, 在讨论 TCP 之前, 我们后退一步, 先来讨论一下可靠数据传输的基本原理是有用的。

3.4 可靠数据传输原理

在本节中, 我们在通常情况下考虑可靠数据传输的问题。因为可靠数据传输的实现问题不仅在运输层出现, 也会在链路层以及应用层出现, 这时讨论它是恰当的。因此, 一般性问题对网络来说更为重要。如果的确要将所有网络中最为重要的“前 10 个”问题排名的话, 可靠数据传输将是名列榜首的候选者。在下一节中, 我们将学习 TCP, 尤其要说明 TCP 所采用的许多原理, 而这些正是我们打算描述的内容。

图 3-8 图示说明了我们学习可靠数据传输的框架。为上层实体提供的服务抽象是: 数据可以通过一条可靠的信道进行传输。借助于可靠信道, 传输数据比特就不会受到损坏 (由 0 变为 1, 或者相反) 或丢失, 而且所有数据都是按照其发送顺序进行交付。这恰好就是 TCP 向调用它的因特网应用所提供的服务模型。

实现这种服务抽象是可靠数据传输协议 (reliable data transfer protocol) 的责任。由于可靠数据传输协议的下层协议也许是不可靠的, 因此这是一项困难的任务。例如, TCP 是在不可靠的 (IP) 端到端网络层之上实现的可靠数据传输协议。更一般的情况是, 两个可靠通信端点的下层可能是由一条物理链路 (如在链路级数据传输协议的场合下) 组成或是一个全球互连网络 (如在运输级协议的场合下) 组成。然而, 就我们的目的而言, 我们可将较低层直接视为不可靠的点对点信道。

在本节中, 考虑到底层信道模型越来越复杂, 我们将不断地开发一个可靠数据传输协议的发送方和接收方。例如, 我们将考虑当底层信道能够损坏比特或丢失整个分组时, 需要什么样的协议机制。这里贯穿我们讨论始终的一个假设是分组将以它们发送的次序进行交付, 某些分组可能会丢失; 这就是说, 底层信道将不会对分组重排序。图 3-8b 图示说明了用于数据传输协议的接口。通过调用 `rdt_send()` 函数, 可以调用数据传输协议的发送