Machine，FSM）的定义。图3-9a中的FSM定义了发送方的操作，图3-9b中的FSM定义了接收方的操作。注意到下列问题是重要的，发送方和接收方有各自的FSM。图3-9中发送方和接收方的FSM每个都只有一个状态。FSM描述图中的箭头指示了协议从一个状态变迁到另一个状态。（因为图3-9中的每个FSM都只有一个状态，因此变迁必定是从一个状态返回到自身；我们很快将看到更复杂的状态图。）引起变迁的事件显示在表示变迁的横线上方，事件发生时所采取的动作显示在横线下方。如果对一个事件没有动作，或没有就事件发生而采取了一个动作，我们将在横线上方或下方使用符号Λ，以分别明确地表示缺少动作或事件。FSM的初始状态用虚线表示。尽管图3-9中的FSM只有一个状态，但马上我们就将看到多状态的FSM，因此标识每个FSM的初始状态是非常重要的。

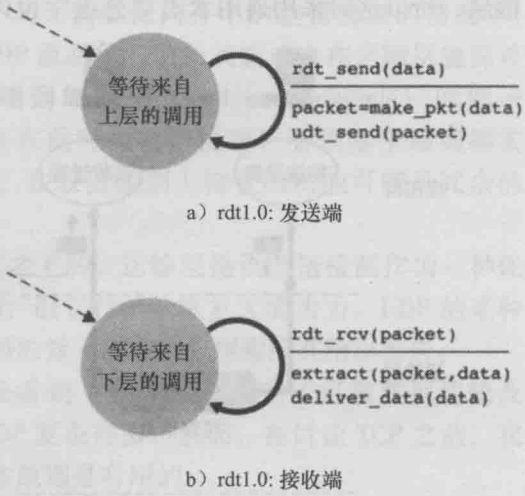

图3-9 rdt1.0：用于完全可靠信道的协议

rdt的发送端只通过rdt_send(data)事件接受来自较高层的数据，产生一个包含该数据的分组（经由make_pkt(data)动作），并将分组发送到信道中。实际上，rdt_send(data)事件是由较高层应用的过程调用产生的（例如，rdt_send()）。

在接收端，rdt通过rdt_rcv(packet)事件从底层信道接收一个分组，从分组中取出数据（经由extract(packet，data)动作），并将数据上传给较高层（通过deliver_data(data)动作）。实际上，rdt_rcv(packet)事件是由较低层协议的过程调用产生的（例如，rdt_rcv()）。

在这个简单的协议中，一个单元数据与一个分组没差别。而且，所有分组是从发送方流向接收方；有了完全可靠的信道，接收端就不需要提供任何反馈信息给发送方，因为不必担心出现差错！注意到我们也已经假定了接收方接收数据的速率能够与发送方发送数据的速率一样快。因此，接收方没有必要请求发送方慢一点！

2. 经具有比特差错信道的可靠数据传输：rdt 2.0

底层信道更为实际的模型是分组中的比特可能受损。在分组的传输、传播或缓存的过程中，这种比特差错通常会出现在网络的物理部件中。我们眼下还将继续假定所有发送的分组（虽然有些比特可能受损）将按其发送的顺序被接收。

在研发一种经这种信道进行可靠通信的协议之前，首先考虑一下人们会怎样处理这类情形。考虑一下你自己是怎样通过电话口述一条长消息的。在通常情况下，报文接收者在听到、理解并记下每句话后可能会说“OK”。如果报文接收者听到一句含糊不清的话时，他可能要求你重复刚才那句话。这种口述报文协议使用了**肯定确认**（positive acknowledgment）（“OK”）与**否定确认**（negative acknowledgment）（“请重复一遍”）。这些控制报文使得接收方可以让发送方知道哪些内容被正确接收，哪些内容接收有误并因此需要重复。在计算机网络环境中，基于这样重传机制的可靠数据传输协议称为**自动重传请求**（Automatic Repeat reQuest，ARQ）**协议**。

基本上，ARQ协议中还需要另外三种协议功能来处理存在比特差错的情况：