

- P12. rdt3.0 协议的发送方直接忽略（即不采取任何动作）接收到的所有出现差错和确认分组的确认号（acknum）字段中的值有差错的分组。假设在这种情况下，rdt3.0 只是重传当前的数据分组，该协议是否还能正常运行？（提示：考虑在下列情况下会发生什么情况：仅有一个比特差错时；报文没有丢失但能出现定时过早超时。考虑到当 n 趋于无穷时，第 n 个分组将被发送多少次。）
- P13. 考虑 rdt 3.0 协议。如果发送方和接收方的网络连接能够对报文重排序（即在发送方和接收方之间的媒体上传播的两个报文段能重新排序），那么比特交替协议将不能正确工作（确信你清楚地理解这时它不能正确工作的原因），试画图说明之。画图时把发送方放在左边，接收方放在右边，使时间轴朝下，标出交换的数据报文（D）和确认报文（A）。要标明与任何数据和确认报文段相关的序号。
- P14. 考虑一种仅使用否定确认的可靠数据传输协议。假定发送方只是偶尔发送数据。只用 NAK 的协议是否会比使用 ACK 的协议更好？为什么？现在我们假设发送方要发送大量的数据，并且该端到端连接很少丢包。在第二种情况下，只用 NAK 的协议是否会比使用 ACK 的协议更好？为什么？
- P15. 考虑显示在图 3-17 中的网络跨越国家的例子。窗口长度设置成多少时，才能使该信道的利用率超过 90%？假设分组的长度为 1500 字节（包括首部字段和数据）。
- P16. 假设某应用使用 rdt3.0 作为其运输层协议。因为停等协议具有非常低的信道利用率（显示在网络跨越国家的例子中），该应用程序的设计者让接收方持续回送许多（大于 2）交替的 ACK 0 和 ACK 1，即使对应的数据未到达接收方。这个应用程序设计将能增加信道利用率吗？为什么？该方法存在某种潜在的问题吗？试解释之。
- P17. 考虑两个网络实体 A 和 B，它们由一条完善的双向信道所连接（即任何发送的报文将正确地收到；信道将不会损坏、丢失或重排序分组）。A 和 B 将以交互的方式彼此交付报文：首先，A 必须向 B 交付一个报文，B 然后必须向 A 交付一个报文，接下来 A 必须向 B 交付一个报文，等等。如果一个实体处于它不试图向另一侧交付报文的状态，将存在一个来自上层的类似于 `rdt_send(data)` 调用的事件，它试图向下传送数据以向另一侧传输，来自上层的该调用能够直接忽略对于 `rdt_unable_to_send(data)` 调用，这通知较高层当前不能够发送数据。[注意：做出这种简化的假设，使你不必担心缓存数据。]

对该协议画出 FSM 说明（一个 FSM 用于 A，一个 FSM 用于 B）。注意你不必担心这里的可靠性机制，该问题的要点在于创建反映这两个实体的同步行为的 FSM 说明。应当使用与图 3-9 中协议 rdt1.0 有相同含义的下列事件和动作：`rdt_send(data)`，`packet = make_pkt(data)`，`udt_send(data)`，`rdt_rcv(packet)`，`extract(packet, data)`，`deliver_data(data)`。保证你的协议反映了 A 和 B 之间发送的严格交替。还要保证在你的 FSM 描述中指出 A 和 B 的初始状态。

- P18. 在 3.4.4 节我们学习的一般性 SR 协议中，只要报文可用（如果报文在窗口中），发送方就会不等待确认而传输报文。假设现在我们要求一个 SR 协议，一次发出一对报文，而且只有在知道第一对报文中的两个报文都正确到达后才发送第二对报文。

假设该信道中可能会丢失报文，但报文不会发生损坏和失序。试为报文的单向可靠传输而设计一个差错控制协议。画出发送方和接收方的 FSM 描述。描述在发送方和接收方之间两个方向发送的报文格式。如果你使用了不同于 3.4 节（例如 `udt_send()`、`start_timer()`、`rdt_rcv()` 等）中的任何其他过程调用，详细地阐述这些动作。举例说明（用发送方和接收方的时序踪迹图）你的协议是如何恢复报文丢失的。

- P19. 考虑一种情况，主机 A 想同时向主机 B 和主机 C 发送分组。A 与 B 和 C 是经过广播信道连接的，即由 A 发送的分组通过该信道传送到 B 和 C。假设连接 A、B 和 C 的这个广播信道具有独立的报文丢失和损坏特性（例如，从 A 发出的报文可能被 B 正确接收，但没有被 C 正确接收）。设计一个类似于停等协议的差错控制协议，用于从 A 可靠地传输分组到 B 和 C。该协议使得 A 直到得知 B 和 C 已经正确接收到当前报文，才获取上层交付的新数据。给出 A 和 C 的 FSM 描述。（提示：B 的 FSM 大体上应当与 C 的相同。）同时，给出所使用的报文格式的描述。