

# 传输层的可靠数据传输



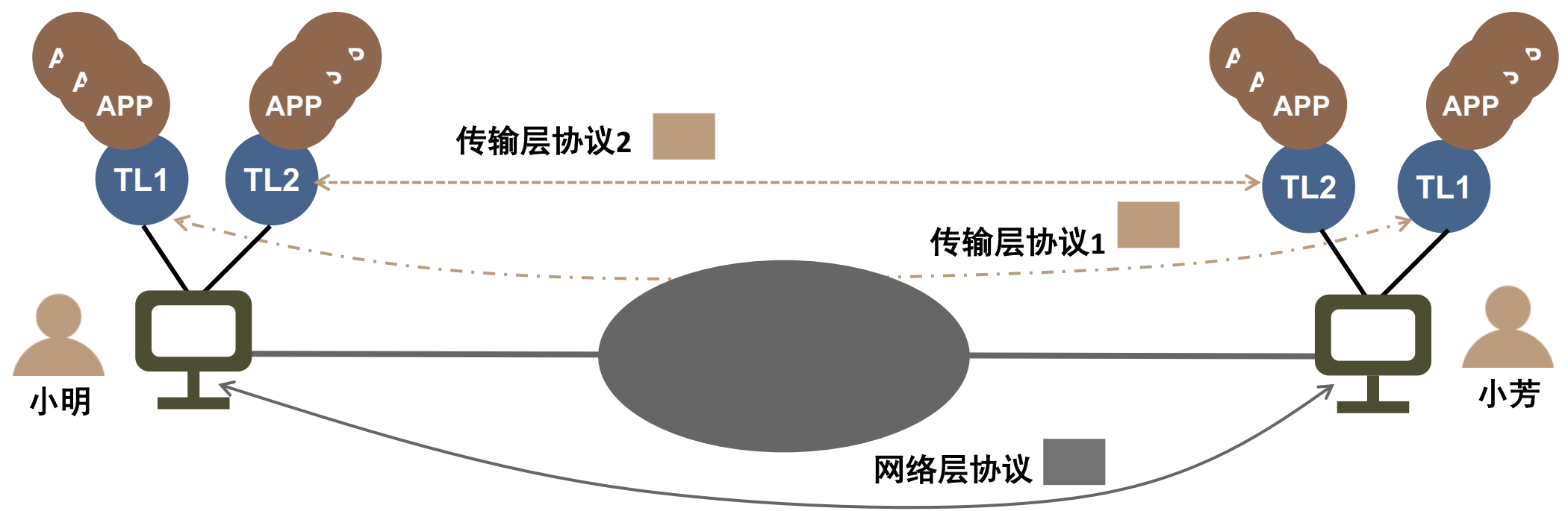
# 传输层使用网络层服务

## 可靠网络服务

- 报文不会丢包
- 报文不会损坏
- 报文保证顺序

## 不可靠网络服务

- 包可能会丢失
- 可能重复损坏
- 可能乱序



# 基于可靠的网络传递服务

## 假设：

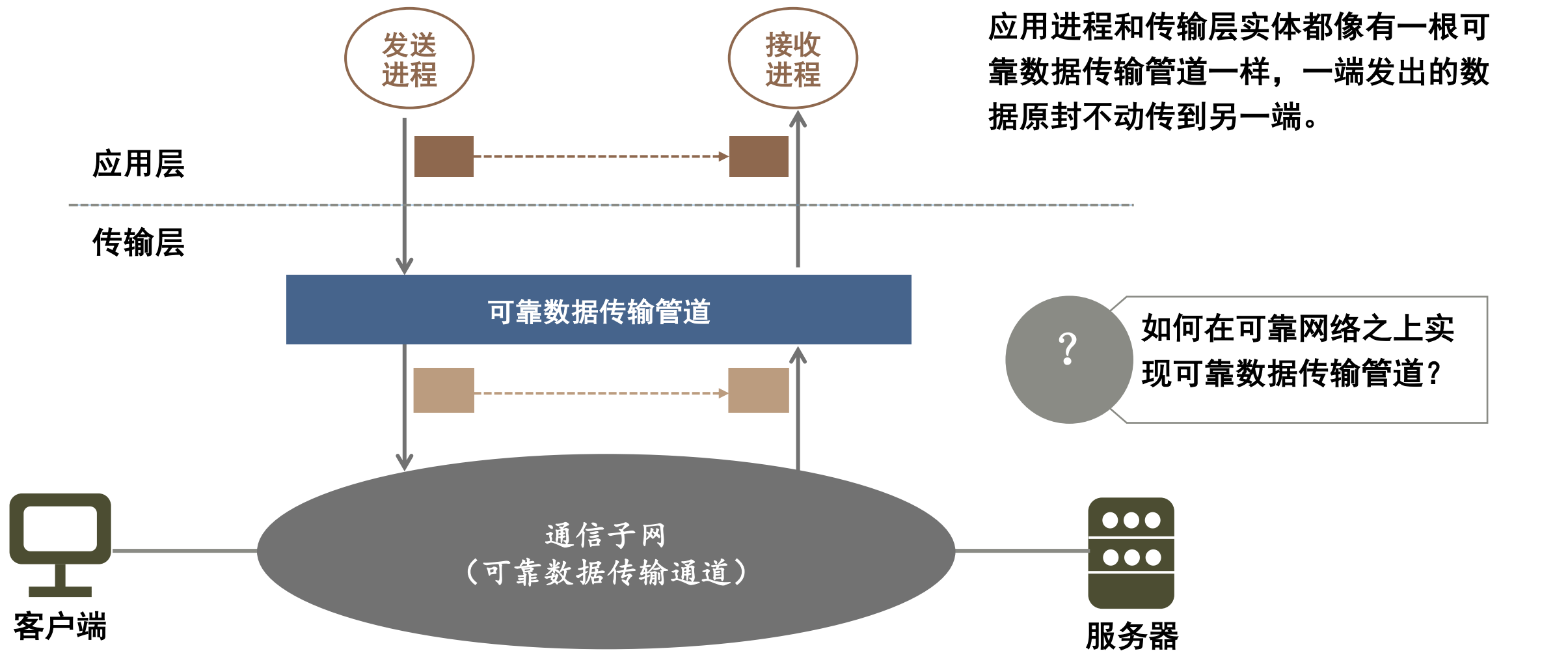
- 底层网络提供可靠数据传输服务
- 接收端有能力及时接收数据

## 特性

- 发送端发出的报文按序到达接收端
- 传递服务无需反馈机制



# 可靠数据传输基本概念



# 基于可靠网络数据通道的rdt1.0

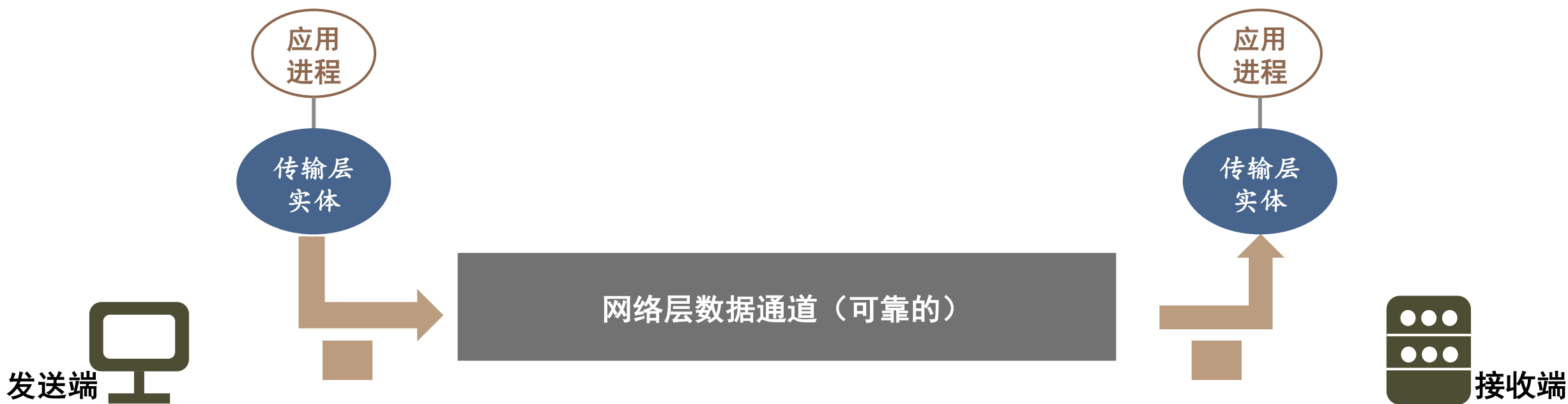
假设：

- 底层网络提供可靠数据传输服务
- 接收端有能力及时接收数据

现在：设计一个可靠的数据传输协议

## 可靠数据传输rdt1.0

- `rdt_send(data)`：发送data
- `rdt_rcv(data)`：接收data



# 可靠数据传输协议rdt1.0——发送端



发送端

初始化

rdt\_send(data)

packet=make\_pkt(data)  
rdt\_send(packet)

等待来自上层的调用

事件

动作

定义了当“事件”发生后，  
传输层实体采取的“动作”

上层应用进程调用rdt\_send()发送数据data,触发下列动作:

- ① 生成一个包含了上层数据data的包packet
- ② 调用rdt\_send()发送包

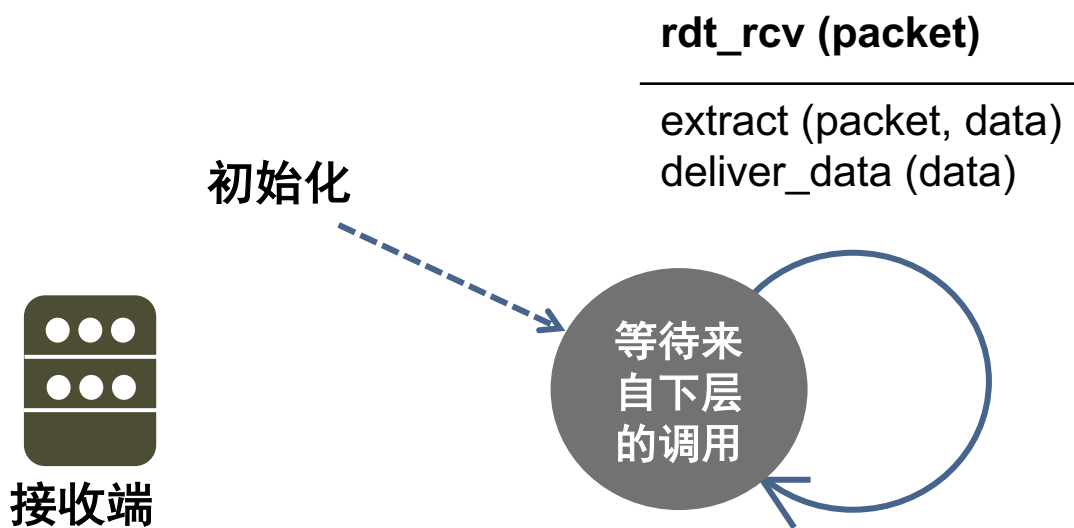
- 发送端传输层实体每次被动地从应用层获得DATA，调用底层可靠的网络传输服务把封装了DATA的包发送给对等的传输层实体
- 可靠的网络传输通道确保接收端的传输层实体一定收到该包

注意：示例中略去了传输层的端口号分配等事宜



北京大学

# 可靠数据传输协议rdt1.0——接收端



底层网络调用rdt\_rcv()把报文传到传输层，触发下列动作：

- ① 从报文packet中取出上层数据data
- ② 调用deliver\_data将数据data传给上层应用进程

事件  
动作

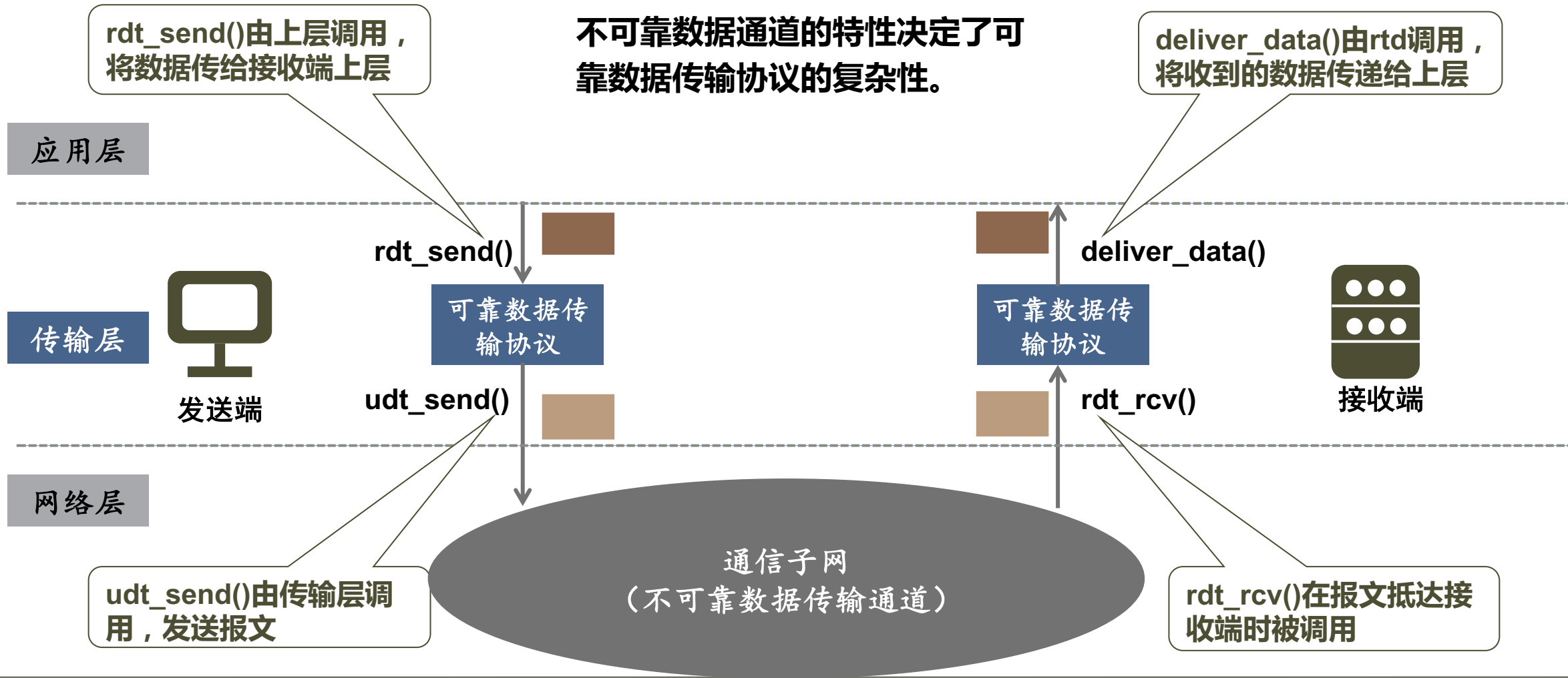
定义了当“事件”发生后，传输层实体采取的“动作”

- 接收端传输层实体等待底层可靠的网络层实体的调用，接收网络层传上来的报文packet
- 传输层实体把报文中的DATA传递给上层应用进程后继续等待下一个报文的到来



# 基于不可靠的网络传递服务

不可靠数据通道的特性决定了可靠数据传输协议的复杂性。



rdt: 可靠数据传输

udt: 不可靠数据传输



北京大学