传输层协议

1. 数据包结构

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 首部 | | | | | | | | | | 数据 |
| 字节 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10…… |
| 含义 | Source ID | Destination ID | Seq number | | Ack number | | reserved | flags | 数据段长度 | 校验 |  |

* 1. Source ID

一个字节，0~255，标识来源

* 1. Destination ID

一个字节，0~255，标识目标

* 1. Seq number

Sequence number，按包计的序列号。如果含有同步化旗标（SYN），则此为最初的序列号；第一个数据包的序列码为本序列号加一。如果没有同步化旗标（SYN），则此为第一个数据包的序列码。

* 1. Ack number

Acknowledgement number，确认号。期望收到的数据包的序列号，也即已经收到的数据包序号加1。

* 1. flags

|  |  |
| --- | --- |
| 位 | 含义 |
| 0 | ACK。建立 TCP 连接后，所有报文段都必须把 ACK 字段置为 1 |
| 1 | SYN |
| 其他 | 保留 |

* 1. 数据段长度

按字节计的数据段长度，包括命令+其他数据

* 1. 校验

校验方法待定

1. 建立连接

无需任何数据交换

1. 数据传输

数据的传输是一个一个数据包进行的，即当前数据包发送完毕并确认收到后，再发送下一数据包。

每个数据包的传输按照以下步骤进行：

1. 发送方首先发送第一个包含序列号为1（可变化）和一些数据的TCP报文段给接收方。
2. 接收方以一个没有数据的TCP报文段来回复（只含报头），用确认号2来表示已完全收到并请求下一个报文段。
3. 发送方在发送了第一条以后，没能收到回应，当时钟（timer）过时（expire）时，他重发第一条。（每次发送者发送一条TCP报文段后，都会再次启动一次时钟：RTT）。

数据包的传输次序由序列号来表明，接收方收到序列号小于当前传输的包的序列号的包时将再次发送包含当前确认号的ACK包。

1. 结束连接

无需任何数据交换

1. 协议分析
   1. seq和ack号的变化举例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 方向 | seq | ack | 数据 |
| n | A->B | x | y | 一些数据 |
| n+1 | B->A | y | x+1 | 空 |
| n+2 | B->A | y | x+1 | 一些数据 |
| n+3 | A->B | x+1 | y+1 | 空 |
| n+4 |  |  |  |  |

* 1. 容错分析
  2. 实现方式举例

建立一个发送函数，每当有发送任务，调用该函数进行发送，发送完毕后进入重发计时状态。

新建一个接收线程，一直阻塞直到读到10个字节（数据包首部长度）。对接收到的首部进行判断：

若源ID和目的ID不符合，则丢弃该包；

若ID符合而seq不符合，发送当前ACK包（指seq = 当前，ack = 当前的包）；

若ID、seq符合而数据段不为空，则认为收到了新包，存储起来并发送ACK。再判断是否处于重发计时状态。若是，则清除计时；

若ID、seq全符合、数据段为空而ack不符合，则丢弃该包；

若ID、seq全符合、数据段为空且ack符合，则判断是否处于重发计时状态。若是，则清除计时；若否，则丢弃该包。

应用层协议

1. 数据包结构

|  |  |  |
| --- | --- | --- |
|  | 首部 | 数据 |
| 字节 | 0 | 1…… |
| 含义 | 命令 |  |

* 1. 命令

|  |  |
| --- | --- |
| 0 | 无特殊含义 |
| 1 | 请求路况信息 |
| 2 | 发送位置信息 |
| 其他 | 保留 |

1. 路况信息请求
2. 小车->上位机：命令位为1，其他数据为空。
3. 上位机->小车：直接回复，命令位为空，其他数据为路况信息，分两次回复
4. 发送位置信息
5. 小车->上位机：命令位为2，其他数据为位置信息