

# TCP通信

## 消息组成

- 源Mac地址, 目标Mac地址
- 源IP地址, 目标IP地址
- 源端口号, 目标端口号 (某一进程, 应用)
- 数据包

## UDP通信

## 顺序问题

采用流水线式发送消息, 如何保证消息顺序的正确性呢

A-> B发送消息, 每一个消息会有一个seq编号, B 在接收到消息后, 会将seq + 1 得到 ACK 消息, 在发送给 A

## 效率问题

如何避免发送一个消息, 需要等到收到确认收到信息后才发送第二个消息

采用流水线式发送消息, 不再等待确认收到消息。

## 丢包问题

如何知道对方是否收到?

- 1, A -> B 发送消息
- 2, B-> A 发送ACK消息, 表示已收到消息

## 流量问题

有时后 A 的消息发送能力强, B的消息接受能力弱, A发送的太多导致B 不能完全接收, 这时候怎么办

A在发送消息的时候, 带上自己的窗口能力大小, B在发送 ACK信息的时候 也带上自己的窗口能力大小, 根据窗口能力 (win) 的最小值来传输数据。

根据 A, B和网络来确定最大的 数据传输能力

## 拥塞问题

有时不是 A, B 窗口能力的大小, 而是网络的问题, 网络传输的能力 弱。

A在发送消息时, 先进行尝试性发送, 传输的数据量从小到大, 来获取该网络能传输的最大数据能力。

## 连接问题

有时B还没准备完毕, 导致 A一开始发送的数据发生了 数据 浪费

- A:我准备好了 (SYN)
- B:我知道了 (ACK), 我准备好了 (SYN)
- A: 我知道了 (ACK)

## 三次握手

## 断开问题

断开连接, 需确保消息接收完全部的数据

- A : 我准备关闭了 (FIN)
- B: 我知道了 (ACK), 接着处理未完成的任务
- B: 我也准备关闭了 (FIN)
- A: 我知道了 (ACK)

## 四次挥手