

通过UDP的聊天软件

1 连接方式

1.1 中心化版本

在聊天用户之间架设服务器, 用户发送的消息先发送至服务器, 随后由服务器解析消息的目的地, 然后转发消息至另外的用户.

- 对于用户端而言, 用户产生的消息发送至固定的服务器IP和端口, 用户端同时维护一个消息循环, 不断从特定端口接收来自服务器的消息推送.
- 对于服务器, 服务器维护一个消息循环, 在收到用户端发送的消息后, 解析消息希望送达的用户, 并根据服务器中存储的该用户的IP和端口号, 将该消息转发给目的用户. 在这种情况下, 服务器需要知道互相聊天用户的IP和程序的端口号.

虽然UDP本身是不基于连接的, 但是UDP包头中存在源端号以及IP协议中存在源IP, 因此可以在不解析内容的情况下辨别消息的来源. 在解析内容后, 也可以在给定规则下辨别消息的来源和目的地.

1.2 P2P版本

直接在聊天用户之间利用UDP发送消息, 无第三方介入, 仅有两个用户端.

- 对于用户端, 用户产生的消息直接发送至另外用户的IP和程序对应的端口. 同时用户端维护消息循环, 不断从接收端口收听对面发送来的消息.

需要注意的是, P2P版本要求在聊天前得知对方用户的IP和端口号. 除非每个用户有相对固定的IP(类似手机号)以及规定某个端口号为该软件的保留端口. 这样的方案并不现实. 可以参考P2P文件传输中, 存在一个相对固定的中央服务器存储在线用户的IP和端口号. 具体而言, 每个用户上线后, 向固定的服务器IP和端口发送UDP消息, 告知上线以及当前的IP和端口. 接下来可以通过几次UDP消息获取到要聊天对面的IP和端口号. 在获取了对方的IP和端口号后, 就可以通过P2P的方式直接建立与对面用户的连接, 并进行聊天.

2 在UDP上建立相对稳定的聊天

由于UDP并不能保证消息的稳定传输, 可以参考TCP的改进策略, 将TCP的部分机制整合进聊天软件中, 在应用层实现基于UDP的稳定传输.

- 给消息添加sequence number. 当接收消息的一方接收到乱序的消息时, 可知可能有丢包的发生. 可以请求发送方重新发送该消息.
- 添加ACK和超时重传机制. 当接收消息的一方收到特定sequence number的消息后, 向发送方发送ACK确认消息送达. 若发送方迟迟未收到ACK的确认, 则提示可能有丢包的发生, 可以重新发送该消息.
- 添加消息校验机制. 可以添加特定的机制检测有没有bit error等错误的发生, 若接收方检测到错误, 可以要求发送方重传.

