软件复用第一次讨论课方案

1252899 阮康乐

1. 长连接心跳机制

长连接：

在一个TCP连接上可以持续发送多个数据包。

主要用于操作频繁，点对点的通讯，而且连接数不能太多。

心跳机制：

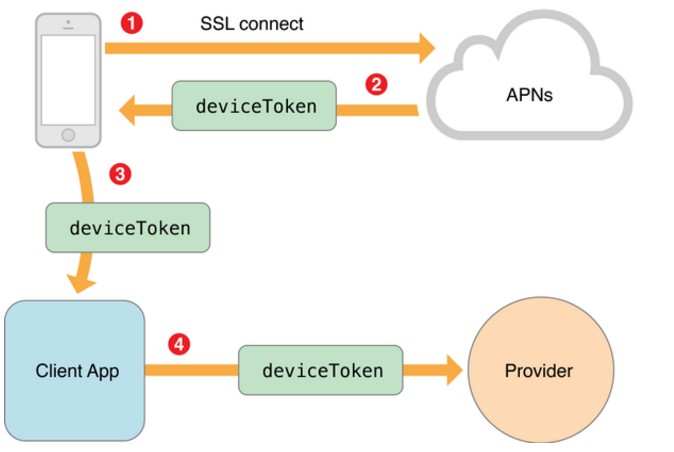
是TCP在一段时间间隔后发送确认连接端是否还在，如果在的话就回传一个包确定网络有效，如果心跳包有问题，则通知上层应用当前网络有问题。

发包方：可以是客户也可以是服务端，看哪边实现方便合理。一般是客户端。服务器也可以定时[轮询](http://www.baike.com/sowiki/%E8%BD%AE%E8%AF%A2?prd=content_doc_search)发心跳下去。心跳包之所以叫心跳包是因为：它像心跳一样每隔固定时间发一次，以此来告诉服务器，这个客户端还活着。事实上这是为了保持[长连接](http://www.baike.com/sowiki/%E9%95%BF%E8%BF%9E%E6%8E%A5?prd=content_doc_search)，至于这个包的内容，是没有什么特别规定的，不过一般都是很小的包，或者只包含包头的一个空包。总的来说，心跳包主要也就是用于长连接的保活和断线处理。一般的应用下，判定时间在30-40秒比较不错。

Ex:IOS系统的长连接心跳机制

iOS 的推送：就是 Apple 官方的 APNs （Apple Push Notification service）。

以微信推送为例，腾讯的服务器（Provider）会给苹果公司对应的服务器（APNs）发出通知，然后再中转传送到你的设备（Devices）之上。当你接收到通知，打开应用，才开始从腾讯服务器接收数据。整个APN保持一个长连接，通过客户端的“心跳”来保持消息推送。





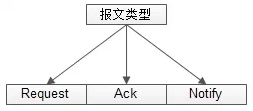
1. 消息不遗漏

im的客户端与服务器通过发送报文（也就是网络包）来完成消息的传递，报文分为三种

请求报文（request，后简称为为R）

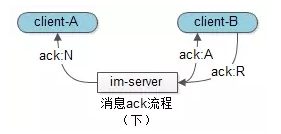
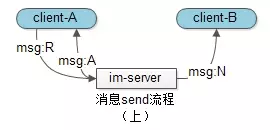
应答报文（acknowledge，后简称为A）

通知报文（notify，后简称为N），这三种报文的解释如下：



**R**：客户端主动发送给服务器的报文  
**A**：服务器被动应答客户端的报文，一个A对应一个R  
**N**：服务器主动发送给客户端的报文

要想实现应用层的消息可靠投递，必须加入应用层的确认机制，即：要想让发送方client-A确保接收方client-B收到了消息，必须让接收方client-B给一个消息的确认，这个应用层的确认的流程，与消息的发送流程类似：



1.client-A向im-server发送一个消息请求包，即msg:R

2.im-server在成功处理后，回复client-A一个消息响应包，即msg:A

3. 如果此时client-B在线，则im-server主动向client-B发送一个消息

通知包，即msg:N（当然，如果client-B不在线，则消息会存储离线）

4．client-B向im-server发送一个ack请求包，即ack:R

5．im-server在成功处理后，回复client-B一个ack响应包，即ack:A

6．则im-server主动向client-A发送一个ack通知包，即ack:N

1. 消息不重复

由发送方client-A生成一个消息去重的msgid，保存在“等待ack队列”里，同一条消息使用相同的msgid来重传，供client-B去重

1. 消息压缩

1.route压缩

2.基于protobuf的传输数据压缩

protobuf协议是由Google制定的，主要用于其内部的rpc调用和文件编码。原生的protobuf包括两部分内容：基于二进制的数据编码协议和基于proto元数据的代码生成器。首先，需要根据每条消息来编写对应的proto文件，然后使用google提供的代码生成器，基于proto文件来生成相应的编码器和解码器，然后使用生成的编/解码器来进行编/解码操作

