软件复用第一次讨论课方案

一、不稳定网络下长连接心跳机制：

1. 设计原则

一般情况下，心跳机制的设计过程，主要从以下几个方面进行考虑：

1) 为了减少协议的开销，心跳消息的交互应该尽量少；

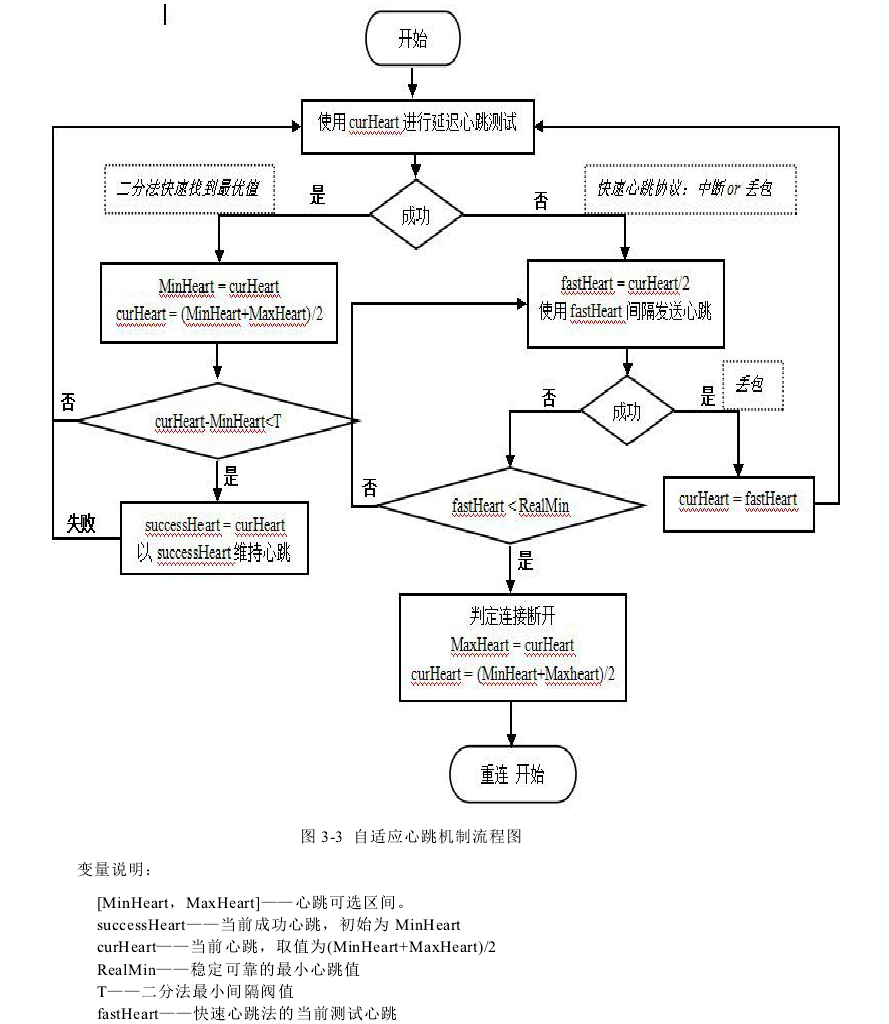
2) 为了保证通信的实时性，心跳值应足够小以避免防火墙、NAT 超时导致连接中断；

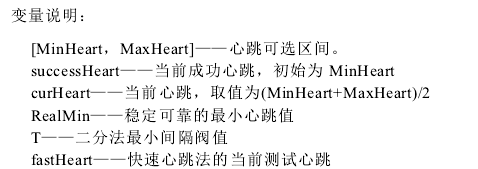
3) 为了增加协议对连接断开的响应速度，超时发现的时间应尽可能短；

4) 为了提高协议的可靠性，由于网络拥塞、网路暂断等原因导致的心跳包偶然丢失从而引发的协议连接终止的可能性应该尽可能小。[1]

2.设计方案

从理论上的心跳值范围内取平均值，再逐步逼近最大的心跳间隔值。如果心跳测试未成功，可能是丢包或者说因为防火墙或NAT超时，这是减小心跳间隔值；当心跳测试成功时，尝试增加心跳值，以减少消耗的资源。通过逐次逼近，最终得出一个略小于防火墙或NAT时限的值。参考文献中的设计方案如下：

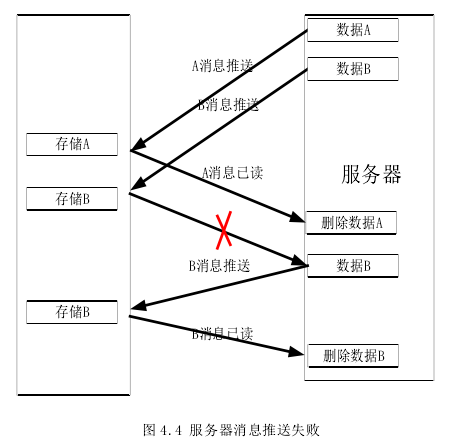




参考文献：

二、消息不遗漏

如建立连接时握手一样，增加返回消息，若发送方没有接收到接收方发出的接收确认消息，则尝试再次发送，直至收到接受确认消息。

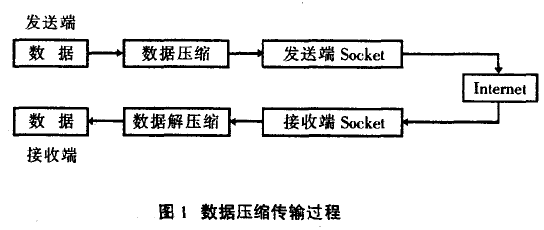
[2]

三、消息不重复

客户端到服务器的握手存在一个问题，如果客户端发送的消息服务器已经接收到，但是当服务器发送反馈信息时，客户端与服务器的通讯链接已经断开，那么客户端本地存储的消息依旧被标记未发送。为了处理这种情况，我们对每一条消息采用了特定的 id 标识，这个 id 可以由时间和客户端特定标识产生。当服务器发送确认信息没有被客户端收到，下次再建立链接时，服务器将不会再发送确认信息，而是由客户端在发一次本地数据库中标识成未发送的数据，服务器接收到了相同 id 的消息后，会把数据库中相同 id 的

消息覆盖，这样就确保了服务器中消息不会重复。[2]

四、消息压缩



无损压缩的算法一般分为两类，一类是基于统计模型的压缩，有Huffman算法和算术编码算法（Arithmetic Coding）等。这类算法压缩率高， 但需要精确的数据统计特性，因为计算时间长；另一类是基于字典（Dictionary）的压缩算法，有LZW算法和Deflate/Inflate算法等。这类算法可以在对数据统计特性一无所知的情况下，是压缩率接近一直统计特性是所能达到的压缩率，而且计算速度快，易于实现，是目前通常的算法。[3] 以上引自2003年文章，目前有消息称Deflate/Inflate算法已过时。

1952年, David A. Huffman提出了一个不同的算法，这个算法可以为任何的可能性提供出一个理想的树。香农-范诺编码（Shanno-Fano）是从树的根节点到叶子节点所进行的的编码，哈夫曼编码算法却是从相反的方向，暨从叶子节点到根节点的方向编码的。

1. 为每个符号建立一个叶子节点，并加上其相应的发生频率
2. 当有一个以上的节点存在时，进行下列循环:
   1. 把这些节点作为带权值的二叉树的根节点，左右子树为空
   2. 选择两棵根结点权值最小的树作为左右子树构造一棵新的二叉树，且至新的二叉树的根结点的权值为其左右子树上根结点的权值之和。
   3. 把权值最小的两个根节点移除
   4. 将新的二叉树加入队列中.
3. 最后剩下的节点暨为根节点，此时二叉树已经完成。[4]

五、参考文献：

1. 温彬民. 一种基于自适应心跳机制的MQTT通信协议的研究与应用[D]. 华南理工大学, 2015.
2. 罗久林. 移动网络下高可靠即时通讯系统的研究与应用[D]. 北京:北京理工大学, 2015:34
3. 李大志. 互联网络数据压缩传输的实现[J]. 气象水文海洋仪器, 2003(1):44-50.
4. http://blog.csdn.net/abcjennifer/article/details/8020695