a)服务端程序

在原有功能的基础上，增加对 “心跳请求”消息的处理。对于接收到的心跳请求消息，如果消息中的dst\_id等于自己的id，就向对端发送“心跳应答”消息。否则，就丢弃此消息。

b)负载均衡程序

之前的负载均衡程序，仅仅实现了轮转式的负载均衡算法。现在需要在原来的基础上，支持更多的负载均衡算法，同时还需要支持会话保持功能、服务端健康检测功能。

具体需要支持的功能如下：

服务端健康检测功能。

负载均衡器每隔0.5秒,向每个服务端发送一个心跳报文。如果连续4次收不到某个服务端的心跳响应，就认为服务端出现了故障。后续对时间请求消息做负载均衡时，就不再分发给此服务端处理。

支持按比例分发的负载均衡算法

由于不同的服务器，配置不同，处理能力不同。因此，有时候将请求消息，根据各服务端的处理能力，按照一定的比例分发到各个服务端，会有更好的整体表现。

按比例分发的负载均衡算法，为每个服务端配置一个权重值（1到10之间的一个数字），在配置文件进行配置。

如果启用了此负载均衡算法，负载均衡程序，将按权重值比例，将时间请求消息分发到相应的服务端。

支持基于最快响应的负载均衡算法

某些情况下，对服务请求的响应速度非常重要。在这种情况下，将服务请求分发给响应速度最快的服务端处理比较合适。

为了支持该功能，负载均衡进程，需要对各个服务端响应心跳请求的速度进行跟踪，从而能够找出响应最快的服务端。

如果启用了此负载均衡算法，负载均衡程序，总是将时间请求消息分发到响应最快的服务端。

基于src\_id的会话保持

启用此功能后，收到来自某个客户端的某条时间请求消息时：

如果发现没有与此客户端对应的会话条目，就认为此消息开启了一条全新的会话，并且此消息是该会话的第一条消息，此时需要将本条时间请求消息按照当前使用的负载均衡算法分发到某个服务端x上，同时创建一个新的会话条目，在其中记录此消息的src\_id及服务端x的id；

如果发现有与此客户端对应的会话条目，则认为此消息是该会话上的后续消息，此时将此消息分发到会话条目记录的服务端上即可。

如果一条会话连续30秒没有任何时间请求消息，则删除此会话条目。

支持基于usr\_id的会话保持功能

启用此功能后，收到来自某个usr\_id的某条时间请求消息时：

如果发现没有与此usr\_id对应的会话条目，就认为此消息开启了一条全新的会话，并且此消息是该会话的第一条消息，此时需要将本条时间请求消息按照当前使用的负载均衡算法分发到某个服务端x上，同时创建一个新的会话条目，在其中记录此消息的src\_id及服务端x的id；

如果发现有与此usr\_id对应的会话条目，则认为此消息是该会话上的后续消息，此时将此消息分发到会话条目记录的服务端上即可。

如果一条会话连续30秒没有任何时间请求消息，则删除此会话条目。

注意，负载均衡程序支持多种负载均衡算法、支持多种会话保持方式以后，究竟使用哪一种负载均衡算法，哪一种会话保持方式，是通过配置文件进行配置的。负载均衡进程在启动时读入这些信息，运行过程中，不需要再动态改变。

**四、复赛阶段审核标准及评价细则**

复赛阶段

* 程序基本功能（小计60分）

心跳检测功能（20分）

会话保持功能（20分）

基于比例的负载均衡算法功能（10分）

基于最快响应的负载均衡算法功能（10分）

* 代码编写情况（小计20分）

代码注释情况（5分）

评委按注释情况打分（0-5分）

代码复杂情况（5分）

评委按复杂程度酌情打分（0-5分）

具体编码情况（5分）

评委按优劣酌情打分（0-5分）

代码的模块化程度和可复用可扩展程度的情况（0-5分）

评委按优劣酌情打分（0-5分）

* 文档编写情况（小计20分）

**设计文档（10分）**

有系统总体框架设计（4分）

有界面设计（3分）

有数据结构描述（3分）

**用户文档（10分）**

有操作步骤说明（5分）

有截图示例（5分）