讨论课 01：

> 用户登陆后始终在线，考虑低宽带/不稳定网络

- 长连接心跳机制
- 消息不遗漏
- 消息不重复
- 消息压缩

一、基础知识了解与分析

网络中的接收和发送数据都是使用操作系统中的 SOCKET 进行实现。但是如果此套接字已经断开，那发送数据和接收数据的时候就一定会有问题。可是如何判断这个套接字是否还可以使用呢？这个就需要在系统中创建心跳机制。其实 TCP 中已经为我们实现了一个叫做心跳的机制。如果你设置了心跳，那 TCP 就会在一定的时间（比如你设置的是 3 秒钟）内发送你设置的次数的心跳（比如说 2 次），并且此信息不会影响你自己定义的协议。所谓“心跳”就是定时发送一个自定义的结构体（心跳包或心跳帧），让对方知道自己“在线”。 以确保链接的有效性。

在 TCP 的机制里面，本身是存在有心跳包的机制的，也就是 TCP 的选项。系统默认是设置的是 2 小时的心跳频率。但是它检查不到机器断电、网线拔出、防火墙这些断线。而且逻辑层处理断线可能也不是那么好处理。一般，如果只是用于保活还是可以的。心跳包一般来说都是在逻辑层发送空的包来实现的。下一个定时器，在一定时间间隔下发送一个空包给客户端，然后客户端反馈一个同样的空包回来，服务器如果在一定时间内收不到客户端发送过来的反馈包，那就只有认定说掉线了。只需要 send 或者 recv 一下，如果结果为零，则为掉线。

但是，在长连接下，有可能很长一段时间都没有数据往来。理论上说，这个连接是一直保持连接的，但是实际情况中，如果中间节点出现什么故障是难以知道的。更要命的是，有的节点（防火墙）会自动把一定时间之内没有数据交互的连接给断掉。在这个时候，就需要我们的心跳包了，用于维持长连接，保活。在获知了断线之后，服务器逻辑可能需要做一些事情，比如断线后的数据清理呀，重新连接呀当然，这个自然是要由逻辑层根据需求去做了。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。如果实在要求高，那就在 6-9 秒。

但普通的 socket 连接对服务器的消耗太大了，所以才会出现像 MQTT 这种轻量级低消耗的协议来维护长连接。

* MQTT：
消息体为：

bit	7	6	5	4	3	2	1	0
byte 1	Message Type				DUP flag	QoS level		RETAIN
byte 2	Remaining Length							

推送的实现方式：

1.客户端不断的查询服务器，检索新内容，也就是所谓的 pull 或者轮询方式

2.客户端和服务器之间维持一个 TCP/IP 长连接，服务器向客户端 push

3.服务器又新内容时，发送一条类似短信的信令给客户端，客户端收到后从服务器中下载新内容，也就是 SMS 的推送方式

二、过程消息的不遗漏、不重复保证

1.Message Type：

16 个枚举值用来表示消息体的状态

Mnemonic	Enumeration	Description
Reserved	0	Reserved
CONNECT	1	Client request to connect to Server
CONNACK	2	Connect Acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish Acknowledgment
PUBREC	5	Publish Received (assured delivery part 1)
PUBREL	6	Publish Release (assured delivery part 2)
PUBCOMP	7	Publish Complete (assured delivery part 3)
SUBSCRIBE	8	Client Subscribe request
SUBACK	9	Subscribe Acknowledgment
UNSUBSCRIBE	10	Client Unsubscribe request
UNSUBACK	11	Unsubscribe Acknowledgment
PINGREQ	12	PING Request
PINGRESP	13	PING Response
DISCONNECT	14	Client is Disconnecting
Reserved	15	Reserved

主要功能：

CONNECT

TCP 连接建立完毕后，Client 向 Server 发出一个 Request。

CONNACK

Server 发出 Response 响应。

PUBLISH 发布消息

Client/Servier 均可以进行 PUBLISH。

publish message 应该包含一个 TopicName(Subject/Channel)，即订阅关键词。

PUBACK 发布消息后的确认

QoS=1 时，Server 向 Client 发布该确认（Client 收到确认后删除），订阅者向 Server 发布确认。

PUBREC / PUBREL / PUBCOMP

QoS=2 时

1. Server->Client 发布 PUBREC（已收到）；
2. Client->Server 发布 PUBREL（已释放）；
3. Server->Client 发布 PUBCOMP（已完成），Client 删除 msg；

订阅者也会向 Server 发布类似过程确认。

PINGREQ / PINGRES 心跳

Client 有责任发送 KeepAliveTime 时长告诉给 Server。在一个时长内，发送 PINGREQ，Server 发送 PINGRES 确认。

2.QOS：Quality Of Service:

QoS value	bit 2	bit 1	Description		
0	0	0	At most once	Fire and Forget	<=1
1	0	1	At least once	Acknowledged delivery	>=1
2	1	0	Exactly once	Assured delivery	=1
3	1	1	Reserved		

3.Clean Session:

如果为 false(flag=0)，Client 断开连接后，Server 应该保存 Client 的订阅信息。

如果为 true(flag=1)，表示 Server 应该立刻丢弃任何会话状态信息。

三、消息压缩：

1、GZIP 的压缩率最高，但是其实 CPU 密集型的，对 CPU 的消耗比其他算法要多，压缩和解压速度也慢；

2、LZO 的压缩率居中，比 GZIP 要低一些，但是压缩和解压速度明显要比 GZIP 快很多，其中解压速度快的更多；

3、Zippy/Snappy 的压缩率最低，而压缩和解压速度要稍微比 LZO 要快一些。