1.长连接心跳机制

自动重连：

实现：客户端定时向服务端发送一个维持连接的包，若长时间未发送维持连接的包，则服务端断开连接，开始重新连接。

客户端：sendObject（）发送 Object 给服务端，在 keepAliveDelay 毫秒内未发送任何数据，则自动发送一个 keepAlive 对象给服务端，用于维持连接。

服务端：检测机制检测由客户端定时发送过来的数据，若未收到，则自动断开与客户端的连接，再建立新连接。

保持持久连接：

1.设置 inactivityTimeout 属性，设置 keepAlive 值，以保持数据层的连接；

<reliableSession enabled=”true” inactivityTimeout=”00:00:30”/>

2.设置 receiveTimeout 属性，服务端运行客户端登录后不做任何应用层服务的调用也可以一直挂着而不回收连接。

receiveTimeout=TimeSpan.MaxValue

变换心跳频率：

根据网络的状态，改变心跳的频率，当网络不稳定时，降低心跳的频率，虽然可能造成网络消息延迟，但可降低 cpu 的使用率，当网络恢复高速时，提高心跳频率，保证消息传送的高速交互。

2.消息不遗漏

客户端发送数据包给服务端后，启动计时器，当服务端接收到数据包并确认后，发送 ACK 给客户端，当客户端收到 ACK 后，确认服务端已收到消息，则停止计时，若在设定的一定时间内未收到 ACK，则客户端重新发送数据包，以实现消息不遗漏。

3.消息不重复

对客户端发送的数据包设置序列号，到服务端收到由客户端发送的数据包时，根据序列号进行比对以判断数据是否重复，若重复，则丢弃重复数据包。

总结消息的不遗漏与不重复：

客户端：对每一次发送的数据包设置序列号，当发送数据包后，启动计时器，等待服务端传回来的 ACK，再停止计时，若由于其他原因服务端未收到数据包或 ACK 在途中丢失，使在预定时间内客户端未收到 ACK，则客户端再次发送数据包，知道确认该数据包已被服务端接收。

服务端：对每一次接收到客户端发送的数据包，进行序列号比对，若当前序列号未接收过，则接收该数据包；若已接收过该数据包，则丢弃该数据包；并向客户端发送 ACK。

若客户端发送的数据包在途中遗漏，则服务端没有向客户端传送 ACK，那么客户端在预定的时间内无法收到 ACK，客户端将会把数据包重发，以保证数据不遗漏。若服务端收到数据包后向客户端发送了 ACK，而 ACK 在途中丢失，则客户端仍会重新发送数据包，而此时服务端收到客户端重发的数据包后，进行序列号比对，发现该数据包已接收过，则将其丢弃，保证消息不重复。

4.消息压缩

网络不稳定时，数据包的发送有延迟，若数据包内有大量重复数据，将数据包进行压缩，可减少数据包的长度，减小消息传送时间。对于不同的报文应采取不同的压缩方法，以达到最佳压缩状态。