数据发送接收的方式：

异步：

异步双工：接收和发送时在同一个进程中，有两个不同的子进程分别负责发送和接收

异步单工：接收和发送用两个不同的程序来完成。

同步 ：

报文的发送和接收是同步进行的，也就是发送完之后，就等待返回，考虑超时，这样就保证了不会无限等待。

（1）ajax轮询

ajax轮询的原理非常简单，让浏览器隔个几秒就发送一次请求，询问服务器是否有新信息。

（2）long poll（长轮询）

long poll 其实原理跟 ajax轮询 差不多，都是采用轮询的方式，不过采取的是阻塞模型（一直打电话，没收到就不挂电话），也就是说，客户端发起连接后，如果没消息，就一直不返回Response给客户端（对于PHP有最大执行时间，建议没消息，执行到一定时间也返回）。直到有消息才返回，返回完之后，客户端再次建立连接，周而复始。

从上面可以看出其实这两种方式，都是在不断地建立HTTP连接，关闭HTTP协议，由于HTTP是非状态性的，每次都要重新传输 identity info （鉴别信息），来告诉服务端你是谁。然后等待服务端处理，可以体现HTTP协议的另外一个特点，被动性。

何为被动性呢，其实就是，服务端不能主动联系客户端，只能有客户端发起。从上面很容易看出来，不管怎么样，上面这两种都是非常消耗资源的。

ajax轮询 需要服务器有很快的处理速度和资源。（速度）long poll 需要有很高的并发，也就是说同时接待客户的能力。（场地大小）

（3）WebSocket

Websocket解决了HTTP的这几个难题。首先，被动性，当服务器完成协议升级后（HTTP->Websocket），服务端就可以主动推送信息给客户端啦。解决了上面同步有延迟的问题。

解决服务器上消耗资源的问题：其实我们所用的程序是要经过两层代理的，即HTTP协议在Nginx等服务器的解析下，然后再传送给相应的Handler（php等）来处理。简单地说，我们有一个非常快速的 接线员（Nginx） ，他负责把问题转交给相应的 客服（Handler） 。Websocket就解决了这样一个难题，建立后，可以直接跟接线员建立持久连接，有信息的时候客服想办法通知接线员，然后接线员在统一转交给客户。

由于Websocket只需要一次HTTP握手，所以说整个通讯过程是建立在一次连接/状态中，也就避免了HTTP的非状态性，服务端会一直知道你的信息，直到你关闭请求，这样就解决了接线员要反复解析HTTP协议，还要查看identity info的信息。

目前唯一的问题是：不兼容低版本的IE

**轮询**：概括来说是服务端定时主动的去与要监控状态的客户端（或者叫其他系统）通信，询问当前的某种状态，客户端返回状态信息，客户端没有返回或返回错误、失效信息、则认为客户端已经宕机，然后服务端自己内部把这个客户端的状态保存下来（宕机或者其他），如果客户端正常，那么返回正常状态，如果客户端宕机或者返回的是定义的失效状态那么当前的客户端状态是能够及时的监控到的，如果客户端宕机之后重启了那么当服务端定时来轮询的时候，还是可以正常的获取返回信息，把其状态重新更新。

短轮询：浏览器发起一个“询问”请求，服务器无论有无新数据，都立即响应（有就返回新数据，没有就返回一个表示’空’的自定义数据格式），一个HTTP连接结束。

长轮询：长轮询的经典实现 —— Comet：基于 HTTP 长连接的“服务器推”技术

浏览器发起一个“询问”请求，当没有新数据时，服务器端并不立即响应，而是等待数据，当有新数据产生时，才向浏览器响应，一个HTTP连接结束。

**心跳**：最终得到的结果是与轮询一样的但是实现的方式有差别，心跳不是服务端主动去发信息检测客户端状态，而是在服务端保存下来所有客户端的状态信息，然后等待客户端定时来访问服务端，更新自己的当前状态，如果客户端超过指定的时间没有来更新状态，则认为客户端已经宕机或者其状态异常

长连接：减少了每次建立连接和断开连接的开销，连接不会长时间保持的，如果长时间没有响应，超时会自动断开的。但是如果都保持连接，那么服务器会爆掉，所以TCP连接数量是有限制的，但长连接占用服务器资源

判断长连接传输结束，两部分：1是判断传输数据是否达到了Content-Length指示的大小；2动态生成的文件没有Content-Length，它是分块传输（chunked），这时候就要根据chunked编码来判断，chunked编码的数据在最后有一个空chunked块，表明本次传输数据结束。

短连接：在Htpp 1.0之前是没有长连接的，现在在http 1.1才开始有长连接，并且大部分都是长连接。短连接的流程就是：建立连接->发送数据->断开连接。一次发送之后立即断开连接。

短连接的好处：

既然有长连接了，为什么还需要用短连接的，因为短连接虽然是之前一直使用的，但是长连接保持了连接不被断开，那么如果并发量很好的时候，就会出现问题，所以短连接不用一直占用服务器资源，可以让服务器空出资源来解决其他的连接。

银行一般使用短连接。好像是因为：管理起来比较简单，存在的连接就是有用的。