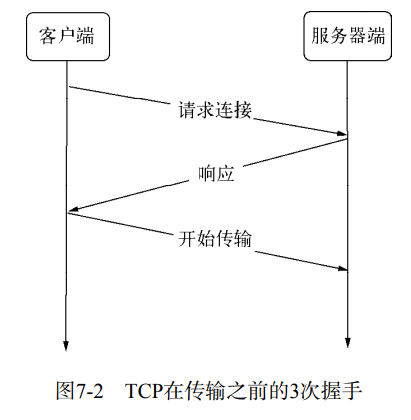
# 网络编程

7.1.1 构建tcp服务

TCP连接时需要三次握手。



开启tcp服务器[openTcp](openTcp.js)，连接tcp服务器[connect](connectedTcp.js)。

7.1.3 tcp服务的事件

服务器事件：

* Listen事件：调用server.listen()绑定服务器端口或者domain socket后触发。绑定方法，server.listen(port,fun)的第二个参数。
* connection事件：每个客户端套接字连接到服务器端时触发。绑定方法，net.createServer（）的最后一个参数（第一或者第二）。
* close事件：服务器关闭时会触发。注意，如果存在连接，这个事件不会被触发直到所有的连接关闭。
* error事件。

连接事件

* + data：一端调用write()后，另一端触发data事件。
  + end：连接中任意一端发送了FIN数据会触发事件。
  + connect：用于客户端，套接字和服务器连接成功触发。
  + drain：调用write()的当前端触发。
  + error：
  + close：套接字完全关闭时触发事件。
  + timeout：一定时间后连接不再活跃，触发该事件。

Tcp套子是可读可写的Stream对象，可以用pipe（）巧妙实现管道操作。

Tcp针对网络中的小数据包采用Nagle算法。只有缓存区的数据到达一定数量时才会发送，但是这有可能造成数据的延迟。可以使用socket.setNoDelay(true)使得write（）可以立即发送数据。调用write（）并不是每次都会触发data事件，关掉Nagle算法后，触发一次data事件。

7.2 构建upd服务器

开启ucp服务器[openUdp](openUdp.js)，连接tcp服务器[connect](connectedUdp.js)。

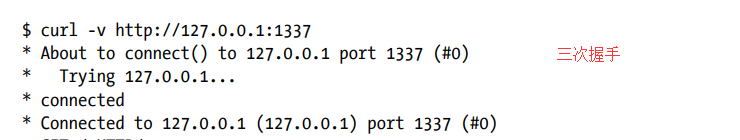
7.2.4 UDP套字事件

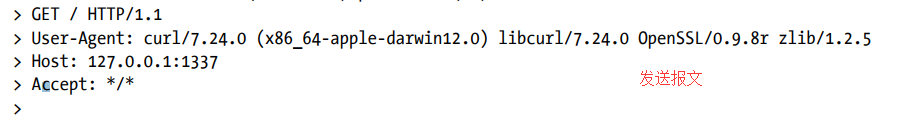
Message：监听端口接收到UDP套字后触发，触发携带带有数据的buffer和一个远程地址。

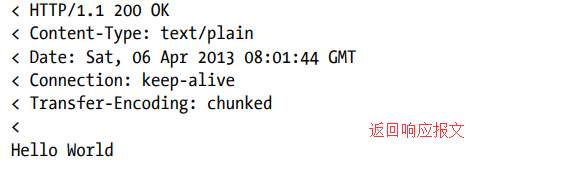
Listening：UDP开始侦听时触发。

7.3 构建HTTP服务

7.1.1 HTTP请求







浏览器其实是一个HTTP请求代理。主要做的事情是发送HTTP请求和响应处理HTTP请求。

7.3.2 http模块