

- 1、与Netty server建立连接时，发送握手消息handshark
- 2、与Netty server握手成功，接收到握手响应，并发送绑定用户消息bind user
- 3、与Netty server绑定用户成功，接收到OK响应，绑定成功；

## 建立连接，开始握手

调用ConnClientBoot#connect，与Netty server建立连接；

连接建立成功，由ConnClientChannelHandler#channelActive()处理，发送握手请求；

```
ConnClientChannelHandler channelActive()

@Override
public void channelActive(ChannelHandlerContext ctx) throws Exception {
    int clientNum = STATISTICS.clientNum.incrementAndGet();
    LOGGER.info("client connect channel={}, clientNum={}", ctx.channel(), clientNum);

    1 for (int i = 0; i < 3; i++) {
        if (clientConfig != null) break;
        clientConfig = ctx.channel().attr(CONFIG_KEY).getAndSet(null);
        if (clientConfig == null) TimeUnit.SECONDS.sleep(1);
    }

    if (clientConfig == null) {
        throw new NullPointerException("client config is null, channel=" + ctx.channel());
    }

    2 connection.init(ctx.channel(), true);
    if (perfTest) {
        handshake(); 3
    } else {
        tryFastConnect();
    }
}
```

CONFIG\_KEY属性对应的clientConfig对象，是在ConnClientBoot#connect创建连接时(还未与netty server建立连接)设置；

- 1、重试3次，获取clientConfig对象，如果3次都获取不到则抛出NPE异常
- 2、连接初始化
- 3、与Netty server握手

## 握手成功，开始绑定用户

```

ConnClientChannelHandler channelRead()

@Override
public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
    connection.updateLastReadTime();
    if (msg instanceof Packet) {
        Packet packet = (Packet) msg;
        Command command = Command.toCMD(packet.cmd);
        if (command == Command.HANDSHAKE) {
            1 int connectedNum = STATISTICS.connectedNum.incrementAndGet();
            2 connection.getSessionContext().changeCipher(new AesCipher(clientConfig.getClientKey(), clientConfig.getIv()));
            3 HandshakeOkMessage message = new HandshakeOkMessage(packet, connection);
            message.decodeBody();
            4 byte[] sessionKey = CipherBox.I.mixKey(clientConfig.getClientKey(), message.serverKey);
            5 connection.getSessionContext().changeCipher(new AesCipher(sessionKey, clientConfig.getIv()));
            6 connection.getSessionContext().setHeartbeat(message.heartbeat);
            7 startHeartBeat(message.heartbeat - 1000);
            LOGGER.info("handshake success, clientConfig={}, connectedNum={}", clientConfig, connectedNum);
            8 bindUser(clientConfig);
            if (!perfTest) {
                9 saveToRedisForFastConnection(clientConfig, message.sessionId, message.expireTime, sessionKey);
            }
        }
    }
}

```

- 1、统计功能，连接数+1
- 2、设置AES加密
- 3、解码接收到的握手响应消息HandshakeOkMessage
- 4、创建sessionKey
- 5、设置AES加密
- 6、设置心跳时间(毫秒)
- 7、开始心跳检测（发送心跳包）
- 8、绑定用户
- 9、保存配置、会话信息到redis中

## 绑定成功

```

ConnClientChannelHandler channelRead()

} else if (command == Command.OK) {
    OkMessage message = new OkMessage(packet, connection);
    message.decodeBody();
    int bindUserNum = STATISTICS.bindUserNum.get();
    if (message.cmd == Command.BIND.cmd) {
        bindUserNum = STATISTICS.bindUserNum.incrementAndGet();
    }

    LOGGER.info("receive {}, bindUserNum={}", message, bindUserNum);
}

```