
Redis网络协议简介

小彬

2015-07-08

大纲

- 网络协议存在的必要性

- RESP

REdis Serialization Protocol

Tcp通信

Client

Server

三次握手建立连接

SYN

SYN ACK

ACK

发送数据

PSH

返回数据

ACK

四次握手关闭连接

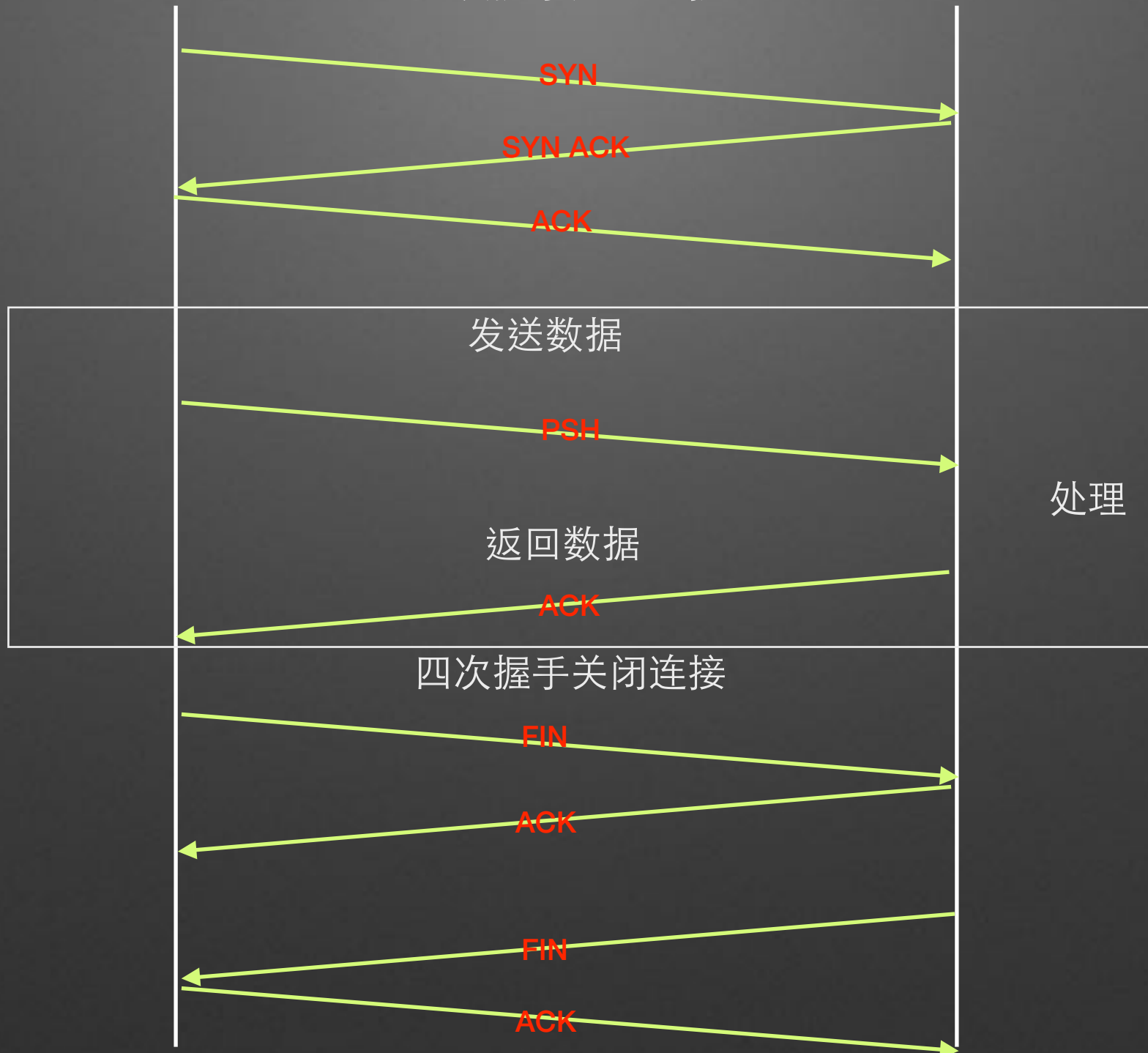
FIN

ACK

FIN

ACK

处理



代码

Client

```
client = socket.socket(  
    socket.AF_INET,  
    socket.SOCK_STREAM  
)  
client.connect(addr)  
client.send( "yourdream.cc" )  
client.recv()  
...  
client.close()
```

Server

```
sock = socket.socket(  
    socket.AF_INET,  
    socket.SOCK_STREAM)  
sock.bind(addr)  
sock.listen(5)  
connection,address = sock.accept()  
recedata = connection.recv(buf_size)  
connection.send('[%s] :%s' %(ctime(),recedata))
```

这里有问题吗？

几个网络问题

半包

粘包

tcp分段

IP分片

MTU

几个网络问题

client.send是否真的发送数据了？

不一定！

- 开启Nagle算法，tcp_nodelay=0
- 超过了MSS的大小，会被分段发送

几个网络问题

client发送A B两个数据到server，要分别处理

```
client.send(A)  
client.send(B)
```

有哪几种情况？

- 发送A后发送B
- 先发送A和B的一部分，再发送B剩余部分
- 先发送A的一部分，再发送A剩余部分和B

服务端如何区分A和B两条数据？

几个网络问题

封包

网络协议

包长+包内容

几个网络问题

短连接

长连接

半包

粘包

RESP (REdis Serialization Protocol)

```
> set yourdream great
```

```
OK
```

```
*3\r\n$3\r\nSET\r\n$9\r\nyourdream\r\n$5\r\ngreat\r\n
```

```
+OK
```

```
> get yourdream
```

```
great
```

```
*2\r\n$3\r\nGET\r\n$9\r\nyourdream\r\n
```

```
$5\r\ngreat\r\n
```

set yourdream great

Request

*3\r\n 数组长度

\$3\r\n 字符串长度

SET\r\n 字符串

\$9\r\n

yourdream\r\n

\$5\r\n

great\r\n

Response

+OK

RESP (REdis Serialization Protocol)

Simple Strings +

Errors -

Integers :

Bulk Strings \$

Arrays *

* Binary Safe String

二进制数据，如JPEG图像，对象序列化数据

RESP (REdis Serialization Protocol)

- 实现方便
- 解析高速
- 阅读友好

RESP (REdis Serialization Protocol)

<http://redis.io/topics/protocol>