TCP连接过程解析

> 传输控制协议（TCP，Transmission Control Protocol）是一种面向连接的、可靠的、基于字节流的传输层通信协议。

一、介绍

- 对`TCP`连接过程的理解有助于以后的编程，尤其是需要扫描网络或传输数据的时候。
- `TCP`连接过程总共分为三个阶段。分别为握手、传输和挥手。通过对比`wireshark`抓包和`TCP`连接示意图来掌握`TCP`连接过程。着重介绍序列号的的变化规则。

一、抓包

- 此次演示使用`Python`程序来建立`TCP`连接进行通信。包括握手、数据传输和挥手三个阶段。
- 设置`wireshark`抓包过滤器规则：

```
host khdxs7.server and port 8600
```

- 在服务器进行监听，启动`server.py`。

```
python3 server.py
```

- 在本地计算机启动客户端，发起连接。

```
python3 client.py
```

> 传输内容为`khdxs7`，共6个字符。

二、分析

> `wireshark`查看`TCP`连接过程：
>
> `wireshark` --> `Statistics` --> `Flow Graph`，`Flow type`设置为`TCP Flows`。

1. 标志位（flags)

- **SYN**（Synchronous）：同步标志，发起连接请求，用于建立连接。
- **ACK**（Acknowledgement）：确认标志，用于表示已收到请求和确认身份。
- **PSH**（Push）：推送标志，用于发送数据。
- **FIN**（Finish)：结束标志，发起断开连接请求，用于断开连接。
- **RST**（Reset）：复位标志，用于出现异常时断开连接。

2. 序列号与确认号（Seq and Ack)

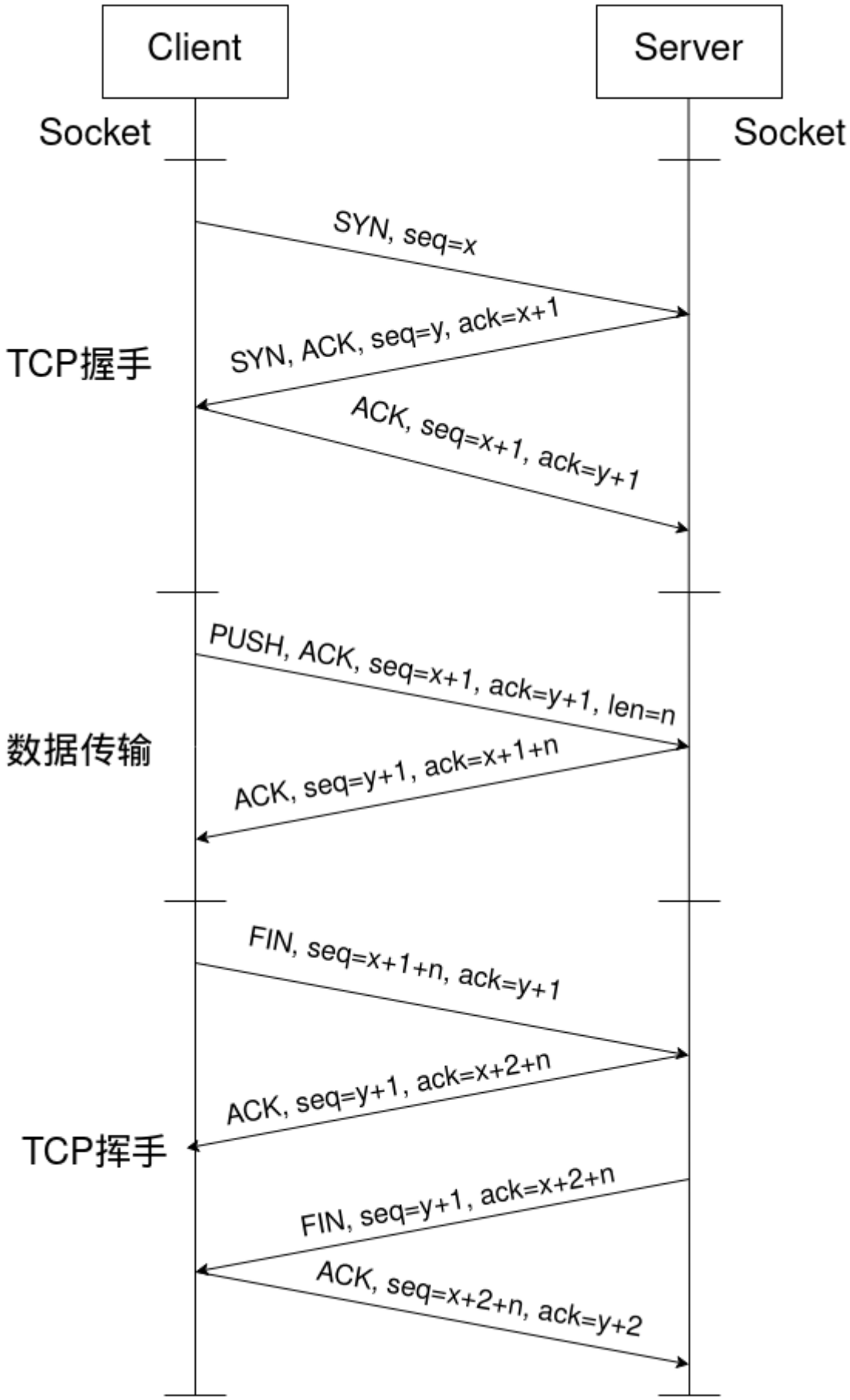
- 序列号增加规则：

```
Seq = 己方序列号（Seq) + 己方数据长度（Len） + （如果SYN=1或FIN=1，则加一）
```

- 确认号增加规则：

```
Ack = 对方序列号（Seq） + 对方数据长度（Len）+ （如果SYN=1或FIN=1，则加一）
```

3. TCP连接示意图


Client
Server
Socket
Socket
SYN, seq=x
SYN, ACK, seq=y, ack=x+1
ACK, seq=x+1, ack=y+1
TCP握手
PUSH, ACK, seq=x+1, ack=y+1, len=n
ACK, seq=y+1, ack=x+1+n
数据传输
FIN, seq=x+1+n, ack=y+1
ACK, seq=y+1, ack=x+2+n
FIN, seq=y+1, ack=x+2+n
ACK, seq=x+2+n, ack=y+2
TCP挥手