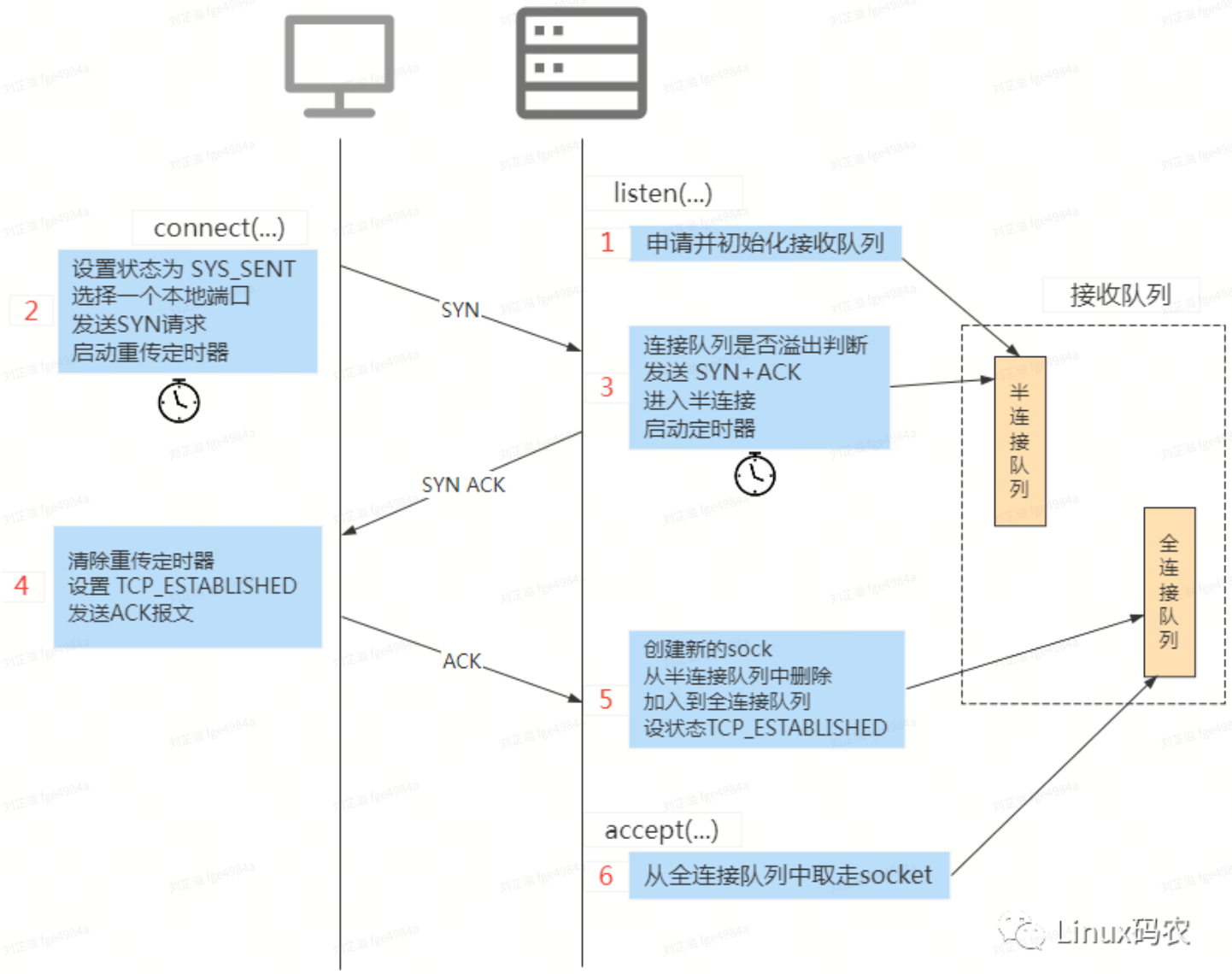
TCP半连接与全连接队列

TCP半连接与全连接

TCP三次握手内核动作

- `listen()` 会初始化半连接队列
- `connect()` 发送SYN请求，server对半连接队列进行溢出判断(**后面会详细讲解**)，没有溢出发送SYN+ACK包；client收到后发送ACK包；Server创建新的sock，从半连接队列中删除，同时将新建的sock加入全连接队列
- `accept()` 就是从全连接队列中取出一个文件描述符



半连接队列和全连接队列

- 半连接队列，SYN队列

- 全连接队列，accept队列

不管是半连接队列还是全连接队列，都有最大长度限制，超过限制时，内核会直接丢弃，或返回 RST 包。

TCP全连接队列情况

- 使用 `ss -lnt` 查看TCP全连接队列情况
- LISTEN状态下
  - Recv-Q：当前全连接队列的大小，也就是当前已完成三次握手并等待服务端 `accept()` 的 TCP 连接；
  - Send-Q：当前全连接最大队列长度
- 非LISTEN状态下
  - Recv-Q：已收到但未被应用进程读取的字节数；
  - Send-Q：已发送但未收到确认的字节数；

```
lzy@lzy-ubuntu2204:~/WorkSpace/NewRelayServer$ ss -lnt
State      Recv-Q     Send-Q         Local Address:Port          Peer Address:Port     Process
LISTEN     0          1024               127.0.0.1:44319              0.0.0.0:*
LISTEN     0          128                  0.0.0.0:22                 0.0.0.0:*
LISTEN     0          4096           127.0.0.53%lo:53                 0.0.0.0:*
LISTEN     0          1024               127.0.0.1:34557              0.0.0.0:*
LISTEN     0          128                127.0.0.1:631                0.0.0.0:*
LISTEN     0          128                    [::]:22                    [::]:*
LISTEN     0          128                   [::1]:631                   [::]:*
```

溢出

- 溢出后的策略由 `tcp_abort_on_overflow` 设置
  - 0： 如果全连接队列满了，那么 server 扔掉 client 发过来的 ACK ；
  - 1： 如果全连接队列满了，server 发送一个 `RST` 包给 client，表示废掉这个握手过程和这个连接；
- **当超过了 TCP 最大全连接队列，服务端默认则会丢掉后续进来的 TCP 连接**，可以使用 `netstate -s` 查看 `overflowed` 的条目

设置全连接队列最大值

- **TCP 全连接队列的最大值取决于 somaxconn 和 backlog 之间的最小值，也就是 min(somaxconn, backlog)**

TCP半连接队列

查看半连接队列

- `netstat -natp | grep SYN_RECV`

溢出

- 服务端有大量的处于 `SYN_RECV` 状态的 TCP 连接。
- 查看溢出 `netstat -s | grep "SYNs to LISTEN"`
- 溢出条件
  - **如果半连接队列满了，并且没有开启 tcp_syncookies，则会丢弃；**
  - **若全连接队列满了，且没有重传 SYN+ACK 包的连接请求多于 1 个，则会丢弃；**
  - **如果没有开启 tcp_syncookies，并且 max_syn_backlog 减去 当前半连接队列长度小于 (max_syn_backlog >> 2)，则会丢弃；**

设置半连接队列最大值

- 半连接队列的最大值 `1 << max_qlen_log` ，在申请队列时确定
- Linux5.0.0:**「理论」**半连接最大值就是全连接队列最大值
  - 由于没有开启 tcp_syncookies， max_syn_backlog 减去当前半连接队列长度小于 (max_syn_backlog >> 2)，客户端发送的ACK包会丢弃；实际的半连接最大值要小于理论值

syncookies

- 不使用SYN半连接队列，与服务器建立连接
  - 服务器根据当前状态计算出一个值，放在己方发出的 SYN+ACK 报文中发出
  - 当客户端返回 ACK 报文时，取出该值验证，如果合法，就认为连接建立成功

防御SYN攻击方法

- 增大半连接队列；
  - **增大 tcp_max_syn_backlog 的值，还需一同增大 somaxconn 和 backlog，也就是增大全连接队列。**
- 开启 tcp_syncookies 功能
- 减少 SYN+ACK 重传次数