

<	2017年10月						>
日	一	二	三	四	五	六	
24	25	26	27	28	29	30	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31	1	2	3	4	

公告

昵称：李大嘴
 园龄：6年7个月
 粉丝：93
 关注：7

搜索

常用链接

我的随笔
 我的评论
 我的参与
 最新评论
 我的标签
 更多链接

我的标签

VS2008 (3)
 SVN (2)
 C++ (1)
 Google (1)
 MySQL (1)
 PostgreSQL (1)

随笔分类

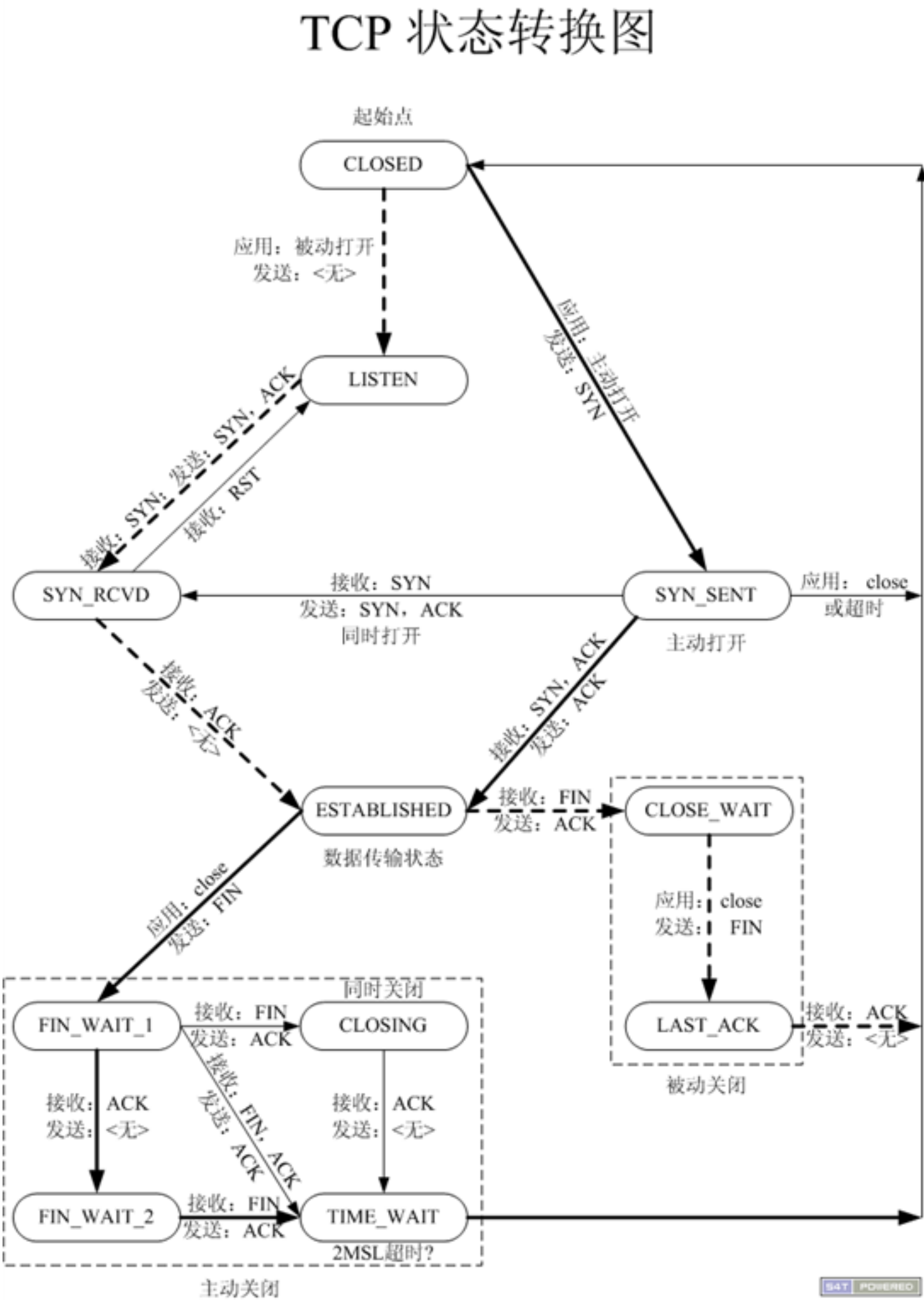
C++(23)
 Data Warehouse(1)
 Database(5)
 Debugging(8)
 Hadoop(4)
 Math&DS&AA(2)
 Network Programming(4)
 OOAD(17)
 Operating System(2)
 Platform Building(3)
 ServerSide(1)
 Software Engineering(17)
 The Others(4)
 TopCoder(1)
 Unix&Linux(2)
 求职应试(9)
 事件系统(3)

随笔档案

2012年1月 (2)
 2011年10月 (3)
 2011年9月 (14)
 2011年8月 (8)

TCP状态转换图

Posted on 2011-07-12 11:47 李大嘴 阅读(39808) 评论(1) 编辑 收藏



注：主动、被动 与 服务器、客户端没有明确的对应关系。

这个图N多人都知道，它排除和定位网络或系统故障时大有帮助，但是怎样牢牢地将这张图刻在脑中呢？那么你就一定要对这张图的每一个状态，及转换的过程有深刻 的认识，不能只停留在一知半解之中。下面对这张图的11种状态详细解析一下，以便加强记忆！不过在这之前，先回顾一下TCP建立连接的三次握手过程，以及 关闭连接的四次握手过程。

1、建立连接协议（三次握手）

- （1）客户端发送一个带SYN标志的TCP报文到服务器。这是三次握手过程中的报文1。
- （2）服务器端回应客户端的，这是三次握手中的第2个报文，这个报文同时带ACK标志和SYN标志。因此它表示对刚才客户端SYN报文的回应；同时又标志SYN给客户端，询问客户端是否准备好进行数据通讯。
- （3）客户必须再次回应服务段一个ACK报文，这是报文段3。

2、连接终止协议（四次握手）

由于TCP连接是全双工的，因此每个方向都必须单独进行关闭。这原则是当一方完成它的数据发送任务后就能发送一个FIN来终止这个方向的连接。收到一个 FIN只意味着这一方向上没有数据流动，一个TCP连接在收到一个FIN后仍能发送数据。首先进行关闭的一方将执行主动关闭，而另一方执行被动关闭。

- （1）TCP客户端发送一个FIN，用来关闭客户到服务器的数据传送（报文段4）。
- （2）服务器收到这个FIN，它发回一个ACK，确认序号为收到的序号加1（报文段5）。和SYN一样，一个FIN将占用一个序号。
- （3）服务器关闭客户端的连接，发送一个FIN给客户端（报文段6）。
- （4）客户段发回ACK报文确认，并将确认序号设置为收到序号加1（报文段7）。

CLOSED：这个没什么好说的了，表示初始状态。

LISTEN：这个也是非常容易理解的一个状态，表示服务器端的某个SOCKET处于监听状态，可以接受连接了。

SYN_RCVD：这个状态表示接受到了SYN报文，在正常情况下，这个状态是服务器端的SOCKET在建立TCP连接时的三次握手会话过程中的一个中间状态，很短暂，基本 上用netstat你是很难看到这种状态的，除非你特意写了一个客户端测试程序，故意将三次TCP握手过程中最后一个ACK报文不予发送。因此这种状态 时，当收到客户端的ACK报文后，它会进入到ESTABLISHED状态。

SYN_SENT：这个状态与SYN_RCVD遥相呼应，当客户端SOCKET执行CONNECT连接时，它首先发送SYN报文，因此也随即它会进入到了SYN_SENT 状 态，并等待服务端的发送三次握手中的第2个报文。SYN_SENT状态表示客户端已发送SYN报文。

ESTABLISHED：这个容易理解了，表示连接已经建立了。

FIN_WAIT_1：这个状态要好好解释一下，其实FIN_WAIT_1和FIN_WAIT_2状态的真正含义都是表示等待对方的FIN报文。而这两种状态的区别 是：FIN_WAIT_1状态实际上是当SOCKET在ESTABLISHED状态时，它想主动关闭连接，向对方发送了FIN报文，此时该SOCKET即 进入到FIN_WAIT_1状态。而当对方回应ACK报文后，则进入到FIN_WAIT_2状态，当然在实际的正常情况下，无论对方何种情况下，都应该马 上回应ACK报文，所以FIN_WAIT_1状态一般是比较难见到的，而FIN_WAIT_2状态还有时常常可以用netstat看到。

FIN_WAIT_2：上面已经详细解释了这种状态，实际上FIN_WAIT_2状态下的SOCKET，表示半连接，也即有一方要求close连接，但另外还告诉对方，

- 2011年7月 (21)
- 2011年6月 (24)
- 2011年5月 (11)
- 2011年4月 (23)
- 2011年3月 (5)

Cloud Computing

51云计算频道

aws.amazon

cloudera

Hadoop资料查询

CSDN Hadoop技术专题

Hadoop wiki

hadoop.apache.org

Hadoop官方网址

hadoop技术论坛

salesforce

Data Warehouse

Alibaba DW

Database

MySQL

PostgreSQL

Linux

Ubuntu中文论坛

Programming

Boost C++ Libraries

C++ Home

codeproject.com/kb/stl/

cplusplus

Cprogramming.com: Your Resource for C++

MSDN

theserverside

一个著名的面向java server端应用的网站

uml.org.cn

VC知识库

www.codeguru.com/Cpp/Cpp/cpp_mfc/stl/

关键函数查询

开源中国社区

曲径通幽

中国开源项目

使用 IBM 中间件实现 SaaS 解决方案

第 1 部分：挑战和体系结构模式

第 2 部分：启用多租户的方法

第 3 部分：单一实例多租户应用程序中的资源共享、隔离和定制

第 4 部分：单一实例多租户应用程序中资源共享的设计模式

第 5 部分：多租户中介方法和三个实现选项

第 6 部分：使用 WebSphere Business Services Fabric 路由多租户请求的 Web 服务中介代理模式

第 7 部分：使用 WebSphere Enterprise Service Bus 路由多个租户的请求的 Web 服务中介代理模式

第 8 部分：WebSphere DataPower SOA Appliances 路由多租户请求的 Web 服务中介代理模式

最新评论

我暂时还有点数据需要传送给你，稍后再关闭连接。

TIME_WAIT: 表示收到了对方的FIN报文，并发送出了ACK报文，就等2MSL后即可回到CLOSED可用状态了。如果FIN_WAIT_1状态下，收到了对方同时带 FIN标志和ACK标志的报文时，可以直接进入到TIME_WAIT状态，而无须经过FIN_WAIT_2状态。

CLOSING: 这种状态比较特殊，实际情况中应该是很少见，属于一种比较罕见的例外状态。正常情况下，当你发送FIN报文后，按理来说是应该先收到（或同时收到）对方的 ACK报文，再收到对方的FIN报文。但是CLOSING状态表示你发送FIN报文后，并没有收到对方的ACK报文，反而却也收到了对方的FIN报文。什 么情况下会出现此种情况呢？其实细想一下，也不难得出结论：那就是如果双方几乎在同时close一个SOCKET的话，那么就出现了双方同时发送FIN报 文的情况，也即会出现CLOSING状态，表示双方都正在关闭SOCKET连接。

CLOSE_WAIT: 这种状态的含义其实是表示在等待关闭。怎么理解呢？当对方close一个SOCKET后发送FIN报文给自己，你系统毫无疑问地会回应一个ACK报文给对 方，此时则进入到CLOSE_WAIT状态。接下来呢，实际上你真正需要考虑的事情是察看你是否还有数据发送给对方，如果没有的话，那么你也就可以 close这个SOCKET，发送FIN报文给对方，也即关闭连接。所以你在CLOSE_WAIT状态下，需要完成的事情是等待你去关闭连接。

LAST_ACK: 这个状态还是比较容易好理解的，它是被动关闭一方在发送FIN报文后，最后等待对方的ACK报文。当收到ACK报文后，也即可以进入到CLOSED可用状态了。

最后有2个问题的回答，我自己分析后的结论（不一定保证100%正确）

1、 为什么建立连接协议是三次握手，而关闭连接却是四次握手呢？

这 是因为服务端的LISTEN状态下的SOCKET当收到SYN报文的建连请求后，它可以把ACK和SYN（ACK起应答作用，而SYN起同步作用）放在一个报文里来发送。但关闭连接时，当收到对方的FIN报文通知时，它仅仅表示对方没有数据发送给你了；但未必你所有的数据都全部发送给对方了，所以你可以未 必会马上会关闭SOCKET,也即你可能还需要发送一些数据给对方之后，再发送FIN报文给对方来表示你同意现在可以关闭连接了，所以它这里的ACK报文 和FIN报文多数情况下都是分开发送的。

2、 为什么TIME_WAIT状态还需要等2MSL后才能返回到CLOSED状态？

这是因为： 虽然双方都同意关闭连接了，而且握手的4个报文也都协调和发送完毕，按理可以直接回到CLOSED状态（就好比从SYN_SEND状态到ESTABLISH状态那样）；但是因为我们必须要假想网络是不可靠的，你无法保证你最后发送的ACK报文会一定被对方收到，因此对方处于 LAST_ACK状态下的SOCKET可能会因为超时未收到ACK报文，而重发FIN报文，所以这个TIME_WAIT状态的作用就是用来重发可能丢失的 ACK报文。

同时打开

两个应用程序同时执行主动打开的情况是可能的，虽然发生的可能性较低。每一端都发送一个SYN,并传递给对方，且每一端都使用对端所知的端口作为本地端口。例如：

主机a中一应用程序使用7777作为本地端口，并连接到主机b 8888端口做主动打开。

主机b中一应用程序使用8888作为本地端口，并连接到主机a 7777端口做主动打开。

tcp协议在遇到这种情况时，只会打开一条连接。

这个连接的建立过程需要4次数据交换，而一个典型的连接建立只需要3次交换（即3次握手）

但多数伯克利版的tcp/ip实现并不支持同时打开。

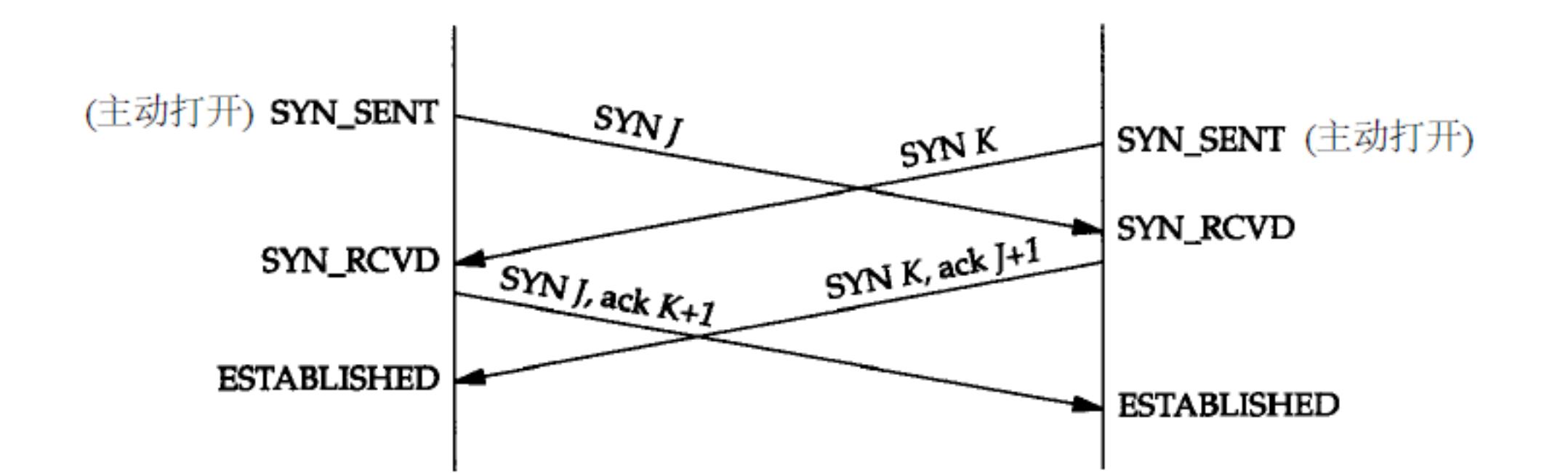


图18-17 同时打开期间报文段的交换

SYN_RCVD与SYN_SEND都是转换为ESTABLISHED的中间状态，目标是两端均转换到ESTABLISHED状态。

同时关闭

如果应用程序同时发送FIN，则在发送后会首先进入FIN_WAIT_1状态。在收到对端的FIN后，回复一个ACK，会进入CLOSING状态。在收到对端的ACK后，进入TIME_WAIT状态。这种情况称为同时关闭。

同时关闭也需要有4次报文交换，与典型的关闭相同。

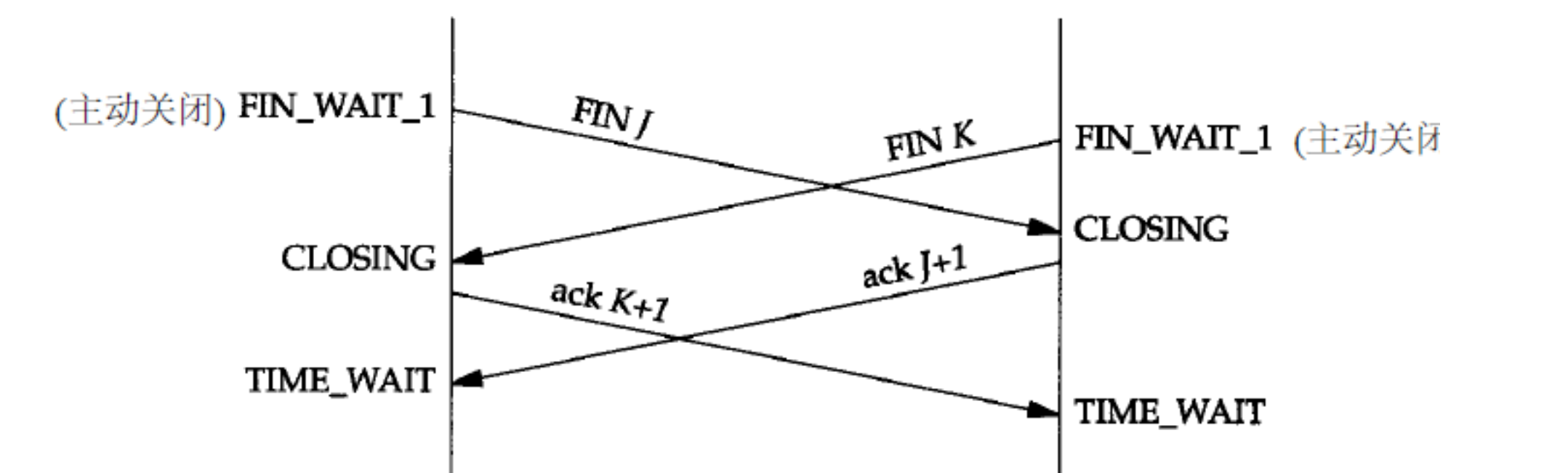


图18-19 同时关闭期间的报文段交换

分类: Network Programming

好文要顶

关注我

收藏该文

李大嘴

关注 - 7

粉丝 - 93

+加关注

7

推荐

0

反对

阅读排行榜

评论排行榜

推荐排行榜

« 上一篇: effective C++ 55 41-55笔记
» 下一篇: 浅析——SCTP协议

Feedback

#1楼
2017-09-24 21:39 by 搏风雨

mark

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

努力加载评论框中...

- 【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【活动】腾讯云 【云+校园】套餐全新升级
- 【推荐】报表开发有捷径: 快速设计轻松集成, 数据可视化和交互



- 最新IT新闻:
- 你以为月入五万的硅谷程序猿想过得像月入五千? 那还不是被逼的
 - 思科即将收购电信软件公司BroadSoft 价格约20亿美元
 - 阅文集团今日在香港IPO, 募集资金最多达11亿美元
 - 趣店抱大腿成700亿市值土豪 校园贷擦边球还打吗?
 - 安卓之父的全面屏Essential Phone降价促销: 4000元
- » 更多新闻...



- 最新知识库文章:
- 实用VPC虚拟私有云设计原则
 - 如何阅读计算机科学类的书
 - Google 及其云智慧
 - 做到这一点, 你也可以成为优秀的程序员
 - 写给立志做码农的大学生
- » 更多知识库文章...