

CSDN

博客学院下载GitChat论坛...

写博客

发Chat

登录注册

沧海一声笑的专栏

目录视图摘要视图RSS 订阅

个人资料

vfdvf

原创49

粉丝2

喜欢2

评论3

等级: 博客 4

积分: 1393

访问量: 7万+

排名: 3万+

前十名婚纱照

文章搜索

文章分类

Pppoe (18)

Ethernet (5)

git 用法 (4)

RTSP (6)

Binder分析 (6)

RtspServer实践 (10)

Top基础与实践 (30)

基础点漏洞 (4)

TCP协议栈进阶 (16)

unix环境高级编程 (14)

c++技术基础 (1)

算法实践 (3)

文章存档

2017年12月 (1)

2017年11月 (8)

2017年10月 (2)

2017年9月 (7)

2017年8月 (59)

展开

阅读排行

TCP重传机制

2017年08月09日 17:22:14740人阅读评论(0)收藏举报

分类: Top基础与实践 (29)

目录(?)

这里的文章重传已经写的比较好的一篇，因此，直接拿来参考！

其实，重传目前在网络的应用从编程的角度来看，用的比较少！就当作一种参考的工具吧！

TCP要保证所有的数据包都可以到达，所以，必须要有**重传机制**。

超时重传机制：就是发送端死等接收端的ack，直到发送端超时之后，在发送一个包，直到收到接收端的ack为止。

例如：接收端给发送端的Ack确认只会确认最后一个连续的包，比如，发送端发了1,2,3,4,5一共五份数据，接收端收到了1，2，于是回ack 3，然后收到了4（注意此时3没收到），此时的TCP会怎么办？等待发送端的ACK 3，直到超时后，就会再发送3。**面临一个艰难的选择，就是，是重传之前的一个还是重传所有的问题。对于上面的示例来说，是重传#3呢还是重传#3，#4，#5呢？**

快速重传机制：这个机制不以时间为驱动，而是以数据来重传！如果接收端包收包没有连续到达，就ACK最后那个可能被丢了的包，如果发送方连续收到接收端3次相同的ack，就重传。

例如：如果发送方发出了1，2，3，4，5份数据，第一份先到送了，于是就ack回2，结果2因为某些原因没收到，3到达了，于是还是ack回2，后面的4和5都到了，但是还是ack回2，因为2还是没有收到，于是发送端收到了三个ack=2的确认，知道了2还没有到，于是就马上重传2。然后，接收端收到了2，此时因为3，4，5都收到了，于是ack回6。

它依然面临一个艰难的选择，就是，是重传之前的一个还是重传所有的问题。对于上面的示例来说，是重传#2呢还是重传#2，#3，#4，#5呢？因为发送端并不清楚这连续的3个ack (2)是谁传回来的？

于是进一步，引入了下面的机制：从发送端入手，选择确定(SACK)（参看RFC 2018），这种方式需要在TCP头里加一个SACK的东西，ACK还是Fast Retransmit的ACK，SACK则是汇报收到的数据碎版。

第1页 共7页

2018/3/17 下午7:47

| | |
|-------------------------------|---------|
| [教程] apply update from ADB... | (10480) |
| pppoe协议和pppd源码分析 | (2674) |
| TCP 、UDP、IP包的最大长... | (2490) |
| PPPOE源码分析 | (1523) |
| 深入理解 linux swapper 进程 | (1394) |
| 浅谈tcp cubic拥塞算法以及优... | (1334) |
| android使用socket使底层和fr... | (1249) |
| select 设置发送超时时发送注意... | (1132) |
| PPPoE工作原理以及PPPoE... | (1128) |
| PPPOE的用户空间实现 | (1075) |

最新评论

TCP拥塞控制慢启动窗口设置====...

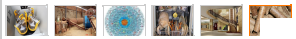
on_way_ : 很详细的文章，文章出处有吗

PPPOE的用户空间实现

vfdvf : [reply]zhaowen_cug[reply] 都是网上的资料，建议多google!



新型燃料



联系我们

请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 💬 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

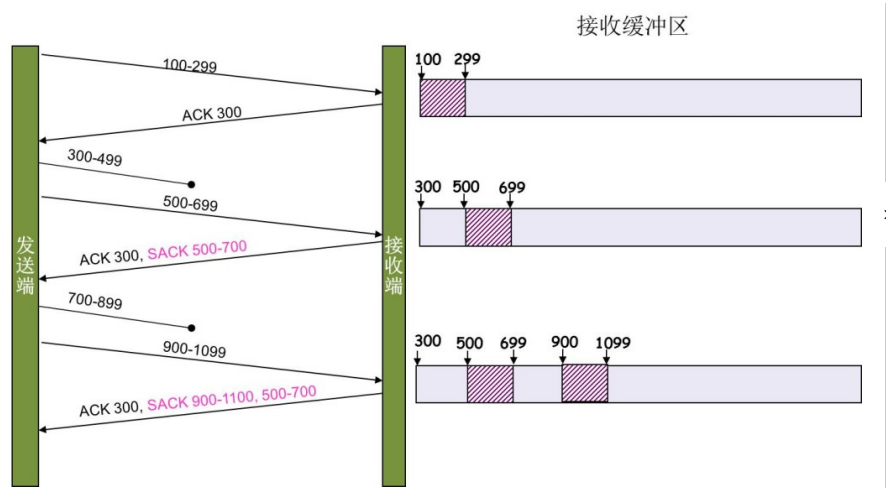
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



这样，在发送端就可以根据回传的SACK来知道哪些数据到了，哪些没有到。于是就优化了Fast Retransmit的算法。当然，这个协议需要两边都支持。在Linux下，可以通过tcp_sack参数打开这个功能（Linux 2.4后默认打开）。

这里还需要注意一个问题——接收方Reneging，所谓Reneging的意思就是接收方有权把已经报给发送端SACK里的数据给丢了。这样干是不被鼓励的，因为这个事会把问题复杂化了，但是，接收方这么做可能会有些极端情况，比如要把内存给别的更重要的东西。所以，发送方也不能完全依赖SACK，还是要依赖ACK，并维护Time-Out，如果后续的ACK没有增长，那么还是要把SACK的东西重传，另外，接收端这边永远不能把SACK的包标记为Ack。

注意：SACK会消费发送方的资源，试想，如果一个攻击者给数据发送方发一堆SACK的选项，这会导致发送方开始要重传甚至遍历已经发出的数据，这会消耗很多发送端的资源。详细的東西請參看《TCP SACK的性能权衡》

Duplicate SACK – 重复收到数据的问题

Duplicate SACK又称D-SACK，其主要使用了SACK来告诉发送方有哪些数据被重复接收了。RFC-2883 里有详细描述和示例。下面举几个例子（来源于RFC-2883）

D-SACK使用了SACK的第一个段来做标志，

- 如果SACK的第一个段的范围被ACK所覆盖，那么就是D-SACK
- 如果SACK的第一个段的范围被SACK的第二个段覆盖，那么就是D-SACK

示例一：ACK丢包

下面的示例中，丢了两个ACK，所以，发送端重传了第一个数据包（3000-3499），于是接收端发现重复收到，于是回了一个SACK=3000-3500，因为ACK都到了4000意味着收到了4000之前的所有数据，所以这个SACK就是D-SACK——旨在告诉发送端我收到了重复的数据，而且我们的发送端还知道，数据包没有丢，丢的是ACK包。

| 1 | Transmitted | Received | ACK Sent |
|---|-------------|-----------|-------------------------|
| 2 | Segment | Segment | (Including SACK Blocks) |
| 3 | | | |
| 4 | 3000-3499 | 3000-3499 | 3500 (ACK dropped) |
| 5 | 3500-3999 | 3500-3999 | 4000 (ACK dropped) |
| 6 | 3000-3499 | 3000-3499 | 4000, SACK=3000-3500 |

示例二，网络延误

下面的示例中，网络包（1000-1499）被网络给延误了，导致发送方没有收到ACK，而后面到达的三个包触发了“Fast Retransmit算法”，所以重传，但重传时，被延误的包又到了，所以，回了一个SACK=1000-1500，因为ACK已到了3000，所以，这个SACK是D-SACK——标识收到了重复的包。

这个案例下，发送端知道之前因为“Fast Retransmit算法”触发的重传不是因为发出去的包丢了，也不是因为回应的ACK包丢了，而是因为网络延时了。

| 1 | Transmitted | Received | ACK Sent |
|----|-------------|-----------|-------------------------|
| 2 | Segment | Segment | (Including SACK Blocks) |
| 3 | | | |
| 4 | 500-999 | 500-999 | 1000 |
| 5 | 1000-1499 | (delayed) | |
| 6 | 1500-1999 | 1500-1999 | 1000, SACK=1500-2000 |
| 7 | 2000-2499 | 2000-2499 | 1000, SACK=1500-2500 |
| 8 | 2500-2999 | 2500-2999 | 1000, SACK=1500-3000 |
| 9 | 1000-1499 | 1000-1499 | 3000 |
| 10 | | 1000-1499 | 3000, SACK=1000-1500 |
| 11 | | | ----- |

可见，引入了D-SACK，有这么几个好处：

- 1) 可以让发送方知道，是发出去的包丢了，还是回来的ACK包丢了。
- 2) 是不是自己的timeout太小了，导致重传。
- 3) 网络上出现了先发的包后到的情况（又称reordering）
- 4) 网络上是不是把我的数据包给复制了。

知道这些东西可以很好得帮助TCP了解网络情况，从而可以更好的做网络上的流控。

Linux下的tcp_dsack参数用于开启这个功能（Linux 2.4后默认打开）

最后，由于最近学习netstat命令，下面就通过一个服务器的命令结果，看下问题：

```
[html]
1.  TcpExt:
2.      79 invalid SYN cookies received
3.      8 resets received for embryonic SYN_RECV sockets
4.      9 packets pruned from receive queue because of socket buffer overrun
5.      5 ICMP packets dropped because they were out-of-window
6.      132513 TCP sockets finished time wait in fast timer
7.      5520362 delayed acks sent
8.      4854 delayed acks further delayed because of locked socket
9.      Quick ack mode was activated 265388 times
10.     112119042 packets directly queued to recvmsg prequeue.
11.     255626524 bytes directly in process context from backlog
12.     2657617950 bytes directly received in process context from prequeue
13.     141 packets dropped from prequeue
14.     2779493398 packet headers predicted
15.     108242651 packets header predicted and directly queued to user
16.     209035695 acknowledgments not containing data payload received
17.     1531587862 predicted acknowledgments
18.     131551 times recovered from packet loss by selective acknowledgements
```

```
19. 279 bad SACK blocks received
20. Detected reordering 10 times using FACK
21. Detected reordering 19 times using SACK
22. Detected reordering 37 times using time stamp
23. 214 congestion windows fully recovered without slow start
24. 20 congestion windows partially recovered using Hoe heuristic
25. 3246 congestion windows recovered without slow start by DSACK
26. 61602 congestion windows recovered without slow start after partial ack
27. TCPLostRetransmit: 31880
28. 12361 timeouts after SACK recovery
29. 289 timeouts in loss state
30. 498564 fast retransmits
31. 13967 forward retransmits
32. 42221 retransmits in slow start
33. 56690 other TCP timeouts
34. TCPLossProbes: 1113019
35. TCPLossProbeRecovery: 928744
36. 7366 SACK retransmits failed
37. 13 times receiver scheduled too late for direct processing
38. 967 packets collapsed in receive queue due to low socket buffer
39. 267981 DSACKs sent for old packets
40. 520 DSACKs sent for out of order packets
41. 966679 DSACKs received
42. 1611 DSACKs for out of order packets received
43. 265 connections reset due to unexpected data
44. 105 connections reset due to early user close
45. 1132 connections aborted due to timeout
46. TCPDSACKIgnoredOld: 4605
47. TCPDSACKIgnoredNoUndo: 723291
48. TCPSpuriousRTOs: 13632
49. TCPSackShifted: 701131
50. TCPSackMerged: 540339
51. TCPSackShiftFallback: 1468281
52. TCPBacklogDrop: 7
53. TCPRetransFail: 126
54. TCPRecvCoalesce: 74902345
55. TCPOFOQueue: 61476920
56. TCPOFOMerge: 521
57. TCPChallengeACK: 1636
58. TCPSYNChallenge: 1600
59. TCPSpuriousRtxHostQueues: 159
```

[html]

```
1. Detected reordering 10 times using FACK
2. Detected reordering 19 times using SACK
3. Detected reordering 37 times using time stamp
```

[html]

```
1. 267981 DSACKs sent for old packets
2. 520 DSACKs sent for out of order packets
3. 966679 DSACKs received
4. 1611 DSACKs for out of order packets received
```

从这里的统计数据可以看出，tcp对于这种数据的问题，采取上述的所有介绍的算法，time stamp，FACK，DSACK，SACK这些算法，了解网络失序有点帮忙，最终还是要抓包来看问题！

至于其他的内容，需要等网络学习完毕之后，再来一一分析！

- [上一篇](#) tcp 状态机中的reset标志
- [下一篇](#) netstat命令学习

PhpStorm工具 - 即刻下载注册



登录官网探索更多优质开发工具,尽享智能高效的编程体验。

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

网络基本功（九）：细说TCP重传



xiongyingzhuantu 2014年10月09日 10:30 14797

转载请在文首保留原文出处：EMC中文支持论坛<https://community.emc.com/go/chinese> 介绍 TCP的主要任务是很简单：打包和发送数据。TCP与...

TCP的超时重传机制与拥塞避免



ahafg 2016年04月04日 21:45 7591

TCP超时与重传机制 TCP协议是一种面向连接的可靠的传输层协议，它保证了数据的可靠传输，对于一些出错，超时丢包等问题TCP设计的超时与重传机制。其基本原理：在发送一个数据之后，就开启一...

技术外文文献看不懂？教你一个公式秒懂英语

不背单词和语法，一个公式学好英语



TCP重传机制



lishanmin11 2017年08月09日 17:22 740

tcp 重传

tcp/ip 上，丢包重传机制



DBC12345666 2015年01月07日 20:23 6394

上篇中，主要向你介绍TCP协议的定义和丢包时的重传机制。下篇中，重点介绍TCP的流控、拥塞处理。废话少说，首先，我们需要知道TCP在网络OSI的七层模型中的第四层——Transport层，IP...

TCP的阻塞和重传机制



farmwang 2017年03月21日 19:52 274

TCP的阻塞和重传机制网络拥堵现在网络上大部分的网络请求都是以TCP的方式进行传输的了。网络链路是固定的，各种链路情况也是不一样的。网络拥堵一直是TCP协议设计和使用的時候尽力要避免的...

程序员不会英语怎么行？

北大猛男教你：不背单词和语法，一个公式学好英语



TCP-IP详解：超时重传机制



wdscq1234 2016年09月11日 00:52 4619

超时重传是TCP保证数据传输可靠性的又一大措施

TCP的重传机制



dreamxiang68 2011年12月30日 10:25 1727

重传机制是TCP 中最重要和最复杂的问题之一。TCP 每发送一个报文段，就对这个报文段设置一次计时器。只要计时器设置的重传时间到但还没有收到确认，就要重传这一报文段。由于TCP 的下层是一...


【原创】TCP超时重传机制探索



heiyeshuwu 2015年06月07日 17:51 4209

TCP对比UDP协议是一个稳定的协议，依赖于三次握手和重传重试机制来保证数据的稳定传输，本文主要是深入探索TCP协议在超时重传方面的内部机制。...


TCP-IP详解：快速重传与快速恢复

 wdscq1234 2016年09月23日 23:20 5331

快速重传算法快速重传算法在之前的文章中有介绍，如果收到一个out-of-order的报文段时，TCP需要立刻产生一个ACK，这个ACK不应该被延时，目的在于让对方知道收到一个失序的报文，并告诉对方...


网络基本功（二十四）：Wireshark抓包实例分析TCP重传

网络基本功（二十四）：Wireshark抓包实例分析TCP重传 转载请在文首保留原文出处：EMC中文支持论坛<https://community.emc.com/go/chinese> ...

 mxway 2015年03月14日 18:08 24343

关于TCP快速重传的细节-重传优先级与重传触发条件

这篇文章写的有点过于细节，因此考虑到可读性和日后的可查阅性，我以两个问题作为引子。作为TCP相关项目的招聘，也可以作为面试题，不过，我敢肯定，大多数人都不能回答第一个问题，第二个问题可能会模棱两可。问...


 dog250 2016年09月15日 05:36 5060

虚拟办公室

虚拟办公室出租 提供注册地址 代办注册及

百度广告

TCP重传机制

 apn172 2012年09月30日 17:36 4702


一、概述 TCP提供可靠地传输层。它使用的方法之一就是确认从另一端收到的数据。单数据和确认都可能会丢失。TCP通过在发送时设置一个定时器来解决这种问题。如果当定时器溢出时还没有收到...

《TCP/IP详解》读书笔记（21章）—TCP的超时与重传

TCP提供可靠的传输层。它使用的方法之一就是确认从另一端收到的数据。但数据和确认都有可能会丢失。TCP通过在发送时设置一个定时器来解决这种问题。如果当定时器溢出时还没有收到确认，它就重传该数据。对于实...


 xifeijian 2015年03月19日 12:59 3849

TCP重传分析

 pangyemeng 2017年09月15日 10:52 344

0x01 缘由 最近在结合linux tcp/ip协议栈，以及上层socket编程来进行相关学习，学习过程中发现一些有趣的东西，但是也想做记录。于是有了这篇文章。 tcp超时...

关于TCP乱序和重传的问题

 cws1214 2016年09月04日 09:50 8756


TCP是一个巨复杂的协议，因为他要解决很多问题，而这些问题又带出了很多子问题和阴暗面。所以学习TCP本身是个比较痛苦的过程，但对于学习的过程却能让人有很多收获。关于TCP这个协议的细节，我还是推荐你去...

程序员不会英语怎么行？

老司机教你一个数学公式秒懂天下英语




TCP重传与确认机制

 u014738387 2016年07月27日 16:07 2986

TCP片段重传计时器以及重传队列:TCP按照以下特定顺序工作：1.放置于重传队列中，计时器开始 包含数据的片段一经发送，片段的一份复制就放在名为重传队列的数据结构中，此时启动重传计时器。因此，在某些时...

Tcp重传

 www4 2015年08月19日 15:29 1257

<http://www.vants.org/?post=36> Ø 为什么TCP存在重传 TCP是一种可靠的协议，在网络交互的过程中，由于TCP报文是封装在IP协议中的，IP协议的无连接特性...

为什么TCP存在重传



health747474 2013年10月01日 10:02 1102

TCP是一种可靠的协议，在网络交互的过程中，由于TCP报文是封装在IP协议中的，IP协议的无连接特性导致其可能在交互的过程中丢失，在这种情况下，TCP协议如何保障其传输的可靠性呢？T C P通...

tcp重传



mingxinjueyu 2012年07月03日 15:32 1440

TCP重传 1，TCP是传输层协议，是TCP/IP协议簇中非常重要的一个协议。2，TCP是可靠传输、面向连接的。3，为了保证TCP的可靠传输，TCP重传机制。对于TCP还是上网络课的时候...

TCP超时重传、滑动窗口、拥塞控制、快重传和快恢复

TCP超时重传 原理是在发送某一个数据以后就开启一个计时器，在一定时间内如果没有得到发送的数据报的ACK报文，那么就重新发送数据，直到发送成功为止。 影响超时重传机制协议效率的一个关键参数是重...



qq_26499321 2017年05月08日 19:33 776