**计算机网络实验报告#**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **课程名称** | 计算机网络 | **班级** | 计算机科学与技术21-1班 | **实验日期** | 12.15 |
| **姓名** | 杨程锦 | **学号** | 2021214710 |  |  |
| **实验名称** | TCP实验 | | | | |
| **实验目的及要求** | | | | | |
| 掌握TCP协议的原理，深入理解TCP协议中的连接管理、可靠传输机制、流量控制机制和拥塞协议。 | | | | | |
| **实验环境** | | | | | |
| 本实验在PC 机上利用Wireshark 进行操作。 | | | | | |
| **实验内容** | | | | | |
| 1. TCP协议基础   I．传输层源地址结构  II．传输层目的地址结构  b）分析TCP连接管理的机制  I．建立连接机制  II．释放连接机制   1. 连接过程中的异常处理   c）分析TCP可靠传输原理  I．确认机制  II．重传机制  III．分析其传输模型  d）分析TCP的流量控制原理  I．流量控制机制  II．零窗口处理机制  III．小窗口、傻瓜窗口问题  e）分析TCP的拥塞控制原理 | | | | | |
| **实验步骤** | | | | | |
| **a．**  Ethernet II：源MAC地址：90:2e:16:64:68:d8 目标MAC地址：58:69:6c:a6:97:79  IPV4： 源IP地址：172.16.7.246 目标IP地址：114.230.222.138  TCP： 源端口：52109 目标端口：443  **b．**  **I．建立连接机制**      主机A：172.16.7.246 主机B：114.230.222.138  第一次握手：  主机A发送报文，将Flags的SYN置位1（表示申请连接主机B的端口），产生Seq Number = 0。  第二次握手：  主机B收到请求后确认联机，向A发送Ack Number =1，（主机A产生的Seq Number+1），将Flags的SYN置位1（表示申请连接主机A的端口），ACK置位1（表示确认主机A的请求），产生Seq Number = 0。  第三次握手：  主机A收到报文后，将ACK置位1（表示确认主机B的请求），Ack Number = 1，之后主机B收到报文后确认Seq值和ACK，最后成功建立连接。  **II．释放连接机制**      服务端：123.249.99.208 客户端：172.16.7.246  第一次挥手：  服务端发送报文，将Flags的FIN、ACK置位1，表示自己没有数据发送，请求断开连接，并且进入FIN\_WAIT\_1状态，随机产生Seq Number = 64、Ack Number = 2。  第二次挥手：  客户端收到FIN之后，确认不会有数据从服务端发来，将返回报文ACK置位1，Ack Number = 65、Seq Number = 2，客户端进入CLOSE\_WAIT状态。  第三次挥手：  客户端发送报文，将Flags的FIN、ACK置位1，表示自己没有数据发送，请求断开连接，并且进入LAST\_ACK状态，然后直接端口TCP会话连接，释放资源，Seq Number = 2、Ack Number = 65。  第四次挥手：  服务端收到客户端的FIN信号后，进入TIMED\_WAIT状态，将ACK置位1。服务端在TIMED\_WAIT状态下，等待一段时间，没有数据到来，就认为对方收到了自己发送的ACK并且正确关闭进入CLOSE状态，然后服务端断开TCP连接，释放资源；客户端收到服务端的ACK回复之后，进入CLOSE状态并关闭本端的会话接口，释放资源。  **III．连接过程中的异常处理**  1、消息丢失的情况  ACK不会重传，SYN和FIN报文段有最大重传次数。无论是SYN还是FIN，达到最大重传次数后对端若仍无响应则直接进入CLOSED状态。  2、尝试连接IP不存在或端口存在的目的主机  （1）当试图连接一个IP不存在的主机时  局域网内：向局域网请求获取目的主机MAC地址，且本地主机不能发出TCP握手消息。  局域网外：触发TCP超时重传，直到最大重传次数后关闭TCP连接。  （2）当试图连接一个IP存在但端口不存在的主机时    目的主机收到握手消息之后，判断端口上没有应用进程，然后会发送RST复位报文，发送段在收到RST后关闭TCP连接。  **c）分析TCP可靠传输原理**  **I．确认机制**    No.1341的Ack Number = 955597，是客户端对服务器发送的数据中序列号为955597的确认，之后服务端向客户端发送三次数据，每次发送长度为1514B，其中包含头部54B，数据1460B，三次数据总计4380B；No.1345的Ack Number = 959977，是客户端对服务器发送数据中序列号为959977的确认。由于TCP为累计确认，会对三次发送的数据只确认一次，所以会对955597 + 3 \* 1460 = 959977进行确认。  **II．重传机制**  （1）TCP Fast Retransmission  当发送方收到3个或以上[TCP Dup ACK]，就意识到之前发的包可能丢了，于是快速重传它。    No.1460中客户端将Ack Number = 49732的报文返回给服务端，客户端理应发送Seq Number = 49732的报文，但是No.1461中，服务端发送了Seq Number = 52652的报文，说明中间缺失了一段数据（TCP Previous segment not captured）。  之后，服务端持续发送数据，客户端返回了7次数据缺失的报文（TCP Dup ACK），直到服务端对其做出处理，将缺失的数据进行快速重传（TCP Fast Retransmission）。  （2）TCP Retransmission  如果一个包真的丢了，又没有后续包可以在接收方触发[Dup Ack]，就不会快速重传。这种情况下发送方等到超时了再重传，此类重传包就会被Wireshark标上[TCP Retransmission]。    客户端发出SYN = 1的报文，请求于服务端建立TCP连接，而服务端未收到该包围，并且二者之间没有TCP连接，所以无法触发Dcp ACK，无法进行快速重传，只能等待超时进行重传（TCP Retransmission）。  （3）TCP Spurious Retransmission      客户端发送No.6391-No.6395报文，均是数据顺序错乱的报文，因为前面有一段数据服务端未收到，而服务端返回的报文仍是对缺失数据的确认，因此，之后会进行重传（TCP Spurious Retransmission）。  **III．分析其传输模型**  （1）三次握手  1）客户端向服务器发送SYN包 2）服务端回应SYN+ACK数据包 3）客户端发送Ack数据包  （2）数据传输流量控制  1）TCP发送方和接收方会协调窗口大小 2）利用序列号和确认号保证数据传输的有序性  （3）超时与重传机制  1）TCP连接设置定时器来检测是否超时 2）超时之后TCP通过重传保证丢失的数据包被成功接收  （4）四次挥手  1）客户端和服务器分别用FIN和ACK进行挥手 2）挥手的最后几个数据包可以保证连接被成功关闭  **d）分析TCP的流量控制原理**  **I．流量控制机制**    TCP使用流量控制来确保发送方不会过度快速地向接收方发送数据，以避免数据的丢失和网络拥塞。这种流量控制通过动态调整TCP窗口大小来实现，其中窗口大小表示接收方能够接收的未确认数据的字节数，窗口变化可以适应不断变化的网络条件，确保数据的可靠传输同时充分利用可用的带宽。  **II．零窗口处理机制**    当TCP窗口大小为0时，它表示接收方暂时不接受数据。TCP使用滑动窗口机制来进行流量控制，其中窗口大小表示接收方缓冲区中可用于接收数据的空间大小。当窗口大小为0时，接收方表示它的缓冲区已满，不能接收更多的数据。当零窗口发生时，发送方应该周期性的检测等待窗口不为0再发送数据；发送方也可也启用延迟重传机制来等待接收方接收到窗口通告。  **III．小窗口、傻瓜窗口问题**    在四次挥手时，即发送方发送带有FIN标志的数据包，通常窗口大小被设置为一个相对较小的值，因为在这个阶段数据传输已经完成，只需用于最后的确认和关闭过程。这可以看作是一种优化，因为在关闭连接的最后阶段，不需要维护大窗口大小。    该窗口大小是513字节，而数据包长度为1412字节，是一个相对较大的数据包；窗口相对较小，可能导致发送方无法充分利用可用带宽，并且后续窗口没有持续增大，因此导致出现傻瓜窗口。  **e）分析TCP的拥塞控制原理**    慢启动：当连接开始时，发送速率呈指数型增长。因此TCP开始发送时候速率很慢，但是慢启动阶段增长很快。因此，在7.5s之前，处于慢启动阶段。  拥塞避免：当在7.5s到196.5s之间增长非常缓慢几乎没有，说明进入了拥塞避免阶段；之后再次增速，是由于窗口变大再次进入快速增长。 | | | | | |
| **关键问题及分析** | | | | | |
| 实验中观察到，在网络拥塞和数据丢包情况下，TCP协议的表现与理论有所差异。这可能是由于网络环境的不确定性和实际应用中的复杂性所致。此外，TCP的流量控制和拥塞控制策略在高负载网络中可能不够高效，需要进一步优化以适应不断变化的网络条件。 | | | | | |
| **总结** | | | | | |
| 本次实验深入分析了TCP协议，特别是其连接管理、可靠传输、流量控制和拥塞控制机制。通过实验，我们直观地了解了TCP连接的建立与释放过程，以及TCP如何在网络环境中处理错误和拥塞。实验结果表明，TCP协议能够有效地管理网络流量，保证数据传输的可靠性和稳定性。 | | | | | |
|  |  | | | | |

**实验成绩评定表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 评价内容 | | 权重 | 得分 |
| **验收** | 实验原理是否理解；程序能否运行；实验结果是否正确；任务是否全部完成。 | | 0.5 |  |
| **实验报告** | 1 | 报告格式是否规范，语言使用是否规范，行文是否流畅，是否图文并茂； | 0.2 |  |
| 2 | 实验原理、实验步骤描述是否正确、详实；  程序流程图是否规范，代码实现是否正确；  实验数据记录是否完整，实验结果是否正确；  实验结果的分析、对比是否充分； | 0.2 |  |
| 3 | 实验体会是否正确，是否提出了自己独到见解。 | 0.1 |  |
| 合计 |  | | | |
|  | | | | |