实验三 传输层的 TCP 协议分析

3.1 实验目的

1. 理解 TCP 报文首部格式和字段的作用，TCP 连接的建立和释放过程，TCP 数据传输中的编号与确认的过程。

2. 理解 TCP 的错误恢复的工作原理和字节流的传输模式，分析错误恢复机制中 TCP 双方的交互情况。

3. 理解 TCP 的流量控制的工作原理，分析流量控制中 TCP 双方的交互情况。

4. 理解 TCP 的拥塞控制的工作原理，分析拥塞控制中 TCP 双方的交互情况。

3.2 实验内容

1. 使用基于 TCP 的应用程序（比如浏览器下载文件）传输文件，在客户端和服务器均要捕获 TCP 报文。

2. 分析 TCP 报文首部信息、TCP 连接的建立和释放过程、TCP 数据的编号与确认机制。观察几个典型的 TCP 选项：MSS、SACK、Window Scale、Timestamp 等，查资料说明其用途。

3. 观察和估计客户机到服务器的 RTT，双方各自的 MSS，计算丢包率及重传的流量。

4. 观察 TCP 的流量控制过程，和拥塞控制中的慢启动、快速重传、拥塞避免，快速恢复等过程【观察拥塞控制的难度较大，观察到两个过程即可】。

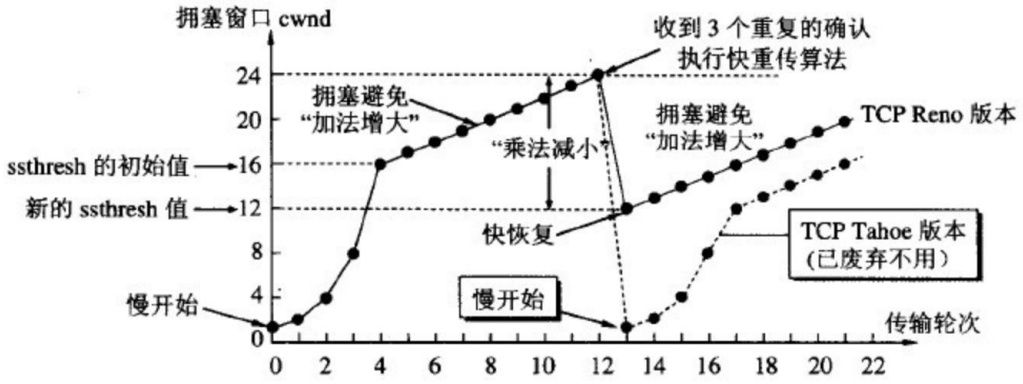

图 3-0 典型的 TCP 拥塞控制过程图例

5. *（可选）注意观察初始的 cwnd 是多少，看看不同的操作系统初始 cwnd

的差别。观察有没有 Delay ACK 的应答模式，注意不同操作系统的差异。

3.3 实验原理

3.3.1 TCP 协议报文格式

TCP 协议工作在网络层之上，是一个面向连接的、端到端的、可靠的传输层协议。TCP 的报文格式如图 3-1，详细地规范参阅 RFC 793。

0 16

32

源端口 Source port								目的端口 Destination port
顺序号 Sequence number								
确认号 Acknowledgement number								
Data Offset	Resrvd	U R	A C	P S	R S	S Y	F I	窗口大小 Window
校验和 Checksum								紧急指针 Urgent pointer
选项和填充 Option + Padding								
数据 Data								

图 3-1 The TCP header structure

1）源端口号，标识主机上发起传送的应用程序；目的端口标识主机上传送要到达的应用程序。源端和目的端的端口号，用于寻找发端和收端应用进程。这两个值加上 I P 包首部中的源端 I P 地址和目的端 I P 地址唯一确定一个 T C P 连接。

2）顺序号字段：占 32 比特。用来标识从 TCP 源端向 TCP 目标端发送的数据字节流，它表示在这个报文段中的第一个数据字节序号。

3）确认号字段：占 32 比特。只有 ACK 标志为 1 时，确认号字段才有效。它包含目标端所期望收到源端发送的下一个数据字节号。

4）Data Offset 字段：占 4 比特。给出头部占 32 比特的数目，同时也指出数据的开始位置。没有任何选项字段的 TCP 头部长度为 20 字节；最多可有 60 字节的 TCP 头部。

5）Resrvd 预留：由跟在数据偏移字段后的 6 位构成，预留位通常为 0.

6）控制标志位（U、A、P、R、S、F）：占 6 比特。各比特的含义如下：

URG：紧急指针（urgent pointer）值有效；

ACK：确认号 Acknowledgement number 值有效；

PSH：接收方应该尽快将这个报文段交给应用层；

RST：重建连接；

SYN：发起一个连接；

FIN： 释放一个连接。

7）窗口大小字段：占 16 比特。此字段用来进行流量控制。单位为字节数，这个值是本机期望一次接收的字节数。

8）TCP 校验和字段：占 16 比特。对整个 TCP 报文段，即 TCP 头部和 TCP 数据进行校验和计算，并由目标端进行验证。

9）紧急指针字段：占 16 比特。URG 设置时有效，它是一个正偏移量，和序号字段中的值相加指向数据包中的第一个重要数据字节。

10）选项字段：占 32 比特。可能包括"窗口扩大因子"、"时间戳"等选项。

3.3.2 TCP 连接的建立与撤销

TCP 连接的建立采用了三次握手方式，连接的撤销则是四次握手，TCP 连接的建立和撤销的过程如图 3-2 所示：

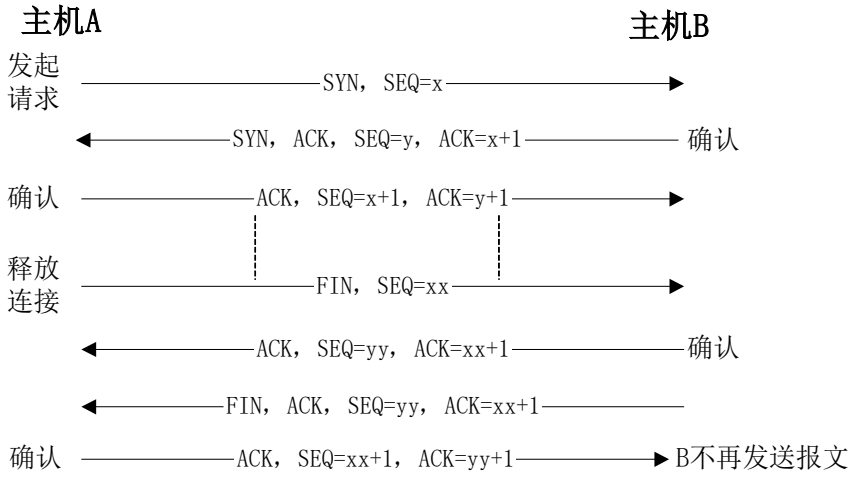

图 3-2 TCP 连接的建立的三次握手

3.4 实验环境与分组

1）云服务器一台，启动 Apache2 服务（或其他服务器程序）。

2）每 2 名同学一组，各自在电脑上运行客户端程序（浏览器或其他客户端程序）。

3）使用客户端程序下载数据，运行 Wireshark 软件捕获报文。【注意：可以关闭 Wireshark 的 HTTP 协议分析，专注在 TCP 协议上，关闭方法是：菜单‘分析’—>‘启用的协议’中，取消‘HTTP’的选择。】

3.5 实验组网

图 3-3 是本实验的组网图，图中参数仅供参考。

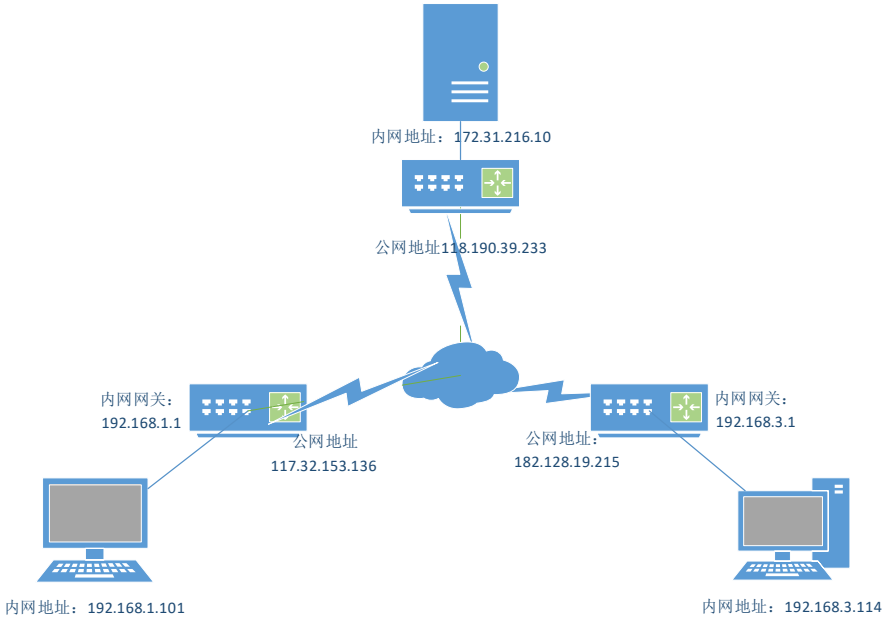

图 3-3 TCP 协议分析组网图

3.6 实验过程及结果分析

步骤 1：PC2 通过 ssh（或远程桌面）登录到服务器 Z 上，在云服务器 Z 上启动合适的服务器程序。

步骤 2：在 PC1 和 Z 上启动报文捕获软件，开始截获报文【注意加过滤器，比如 host w.x.y.z；不熟悉 tcpdump 的可以用 tcpdump -n -s 500 tcp and host A.B.C.D and port P -w server.pcap 选项，把报文记录到文件中，传输到客户端用 Wireshark 分析。其中 A.B.C.D 是客户端的**公网地址，P 是服务端口，如 80】**。

步骤 3：在 PC1 上运行客户端软件，发送和接收一个约 500KB 的文件。文件传输完成后，停止报文截获。

步骤 4：对比观察客户端和服务器截获的报文，分析 TCP 协议的建立过程的三个报文并填写表 3-1。分析 TCP 连接的释放过程，选择 TCP 连接撤销的四个报文并填写表 3-2。

表 3-1 TCP 连接建立过程的三个报文信息【如果有多条，全部列出】

字段名称	第 1 条报文	第 2 条报文	第 3 条报文
报文序号 NO.			
Seq #			

Ack #			
ACK Flag			
SYN Flag			

表 3-2 TCP 连接撤销的四个报文信息

字段名称	首条报文	二条报文	三条报文	四条报文
报文捕获序号 NO.				
Seq #				
Ack #				
ACK				
FIN				

步骤 5：分析 TCP 数据传送阶段的报文，分析其错误恢复和流量控制机制，并填表。【注：出现明显的流量控制的地方，Wireshark 会有[TCP Window Full]标记。如果没有观察到明显的流量控制过程，可以再单独设计实验测试。比如编程设计接收端缓慢接收数据。】

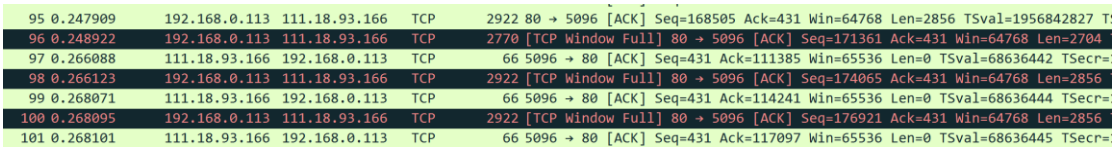

表 3-3 记录 TCP 数据传送阶段的报文

报文序号	报文种类(数据/确认)	序号字段 Seq Number	确认号 Ack Number	数据长度	确认到哪条报文（填序号）	窗口大小

步骤 6、分析客户机和服务器两边各自捕获到的分组，分析整个 TCP 流，估

计双方的 RTT，丢包率和重传流量，平均传输速度等参数。

步骤 7、分析整个 TCP 流的拥塞控制，找到拥塞控制的几个典型过程（即慢启动、快速重传、拥塞避免，快速恢复），计算各个时期发送数据平均传输速度。

步骤 8、如果拥塞控制的相关过程不明显，请设计合适的方法再次测试。

步骤 9、完成其他可选的实验步骤。

3.7 互动讨论主题

1）TCP 的流量控制和拥塞控制有什么不同？

2）TCP 的流量控制是哪一方（接收、发送）来主导的？什么情况下会发生流量控制？

3）讨论传输层与其上下相邻层的关系；

4）讨论 TCP 协议在传输实时语音流方面的优缺点。