

第一部分实验总结报告

袁欣怡 2021.4.24

CONTENTS

1

实验过程

2

实验心得&
与理论课联系

3

遇到的问题

4

实验建议

02 - Mininet



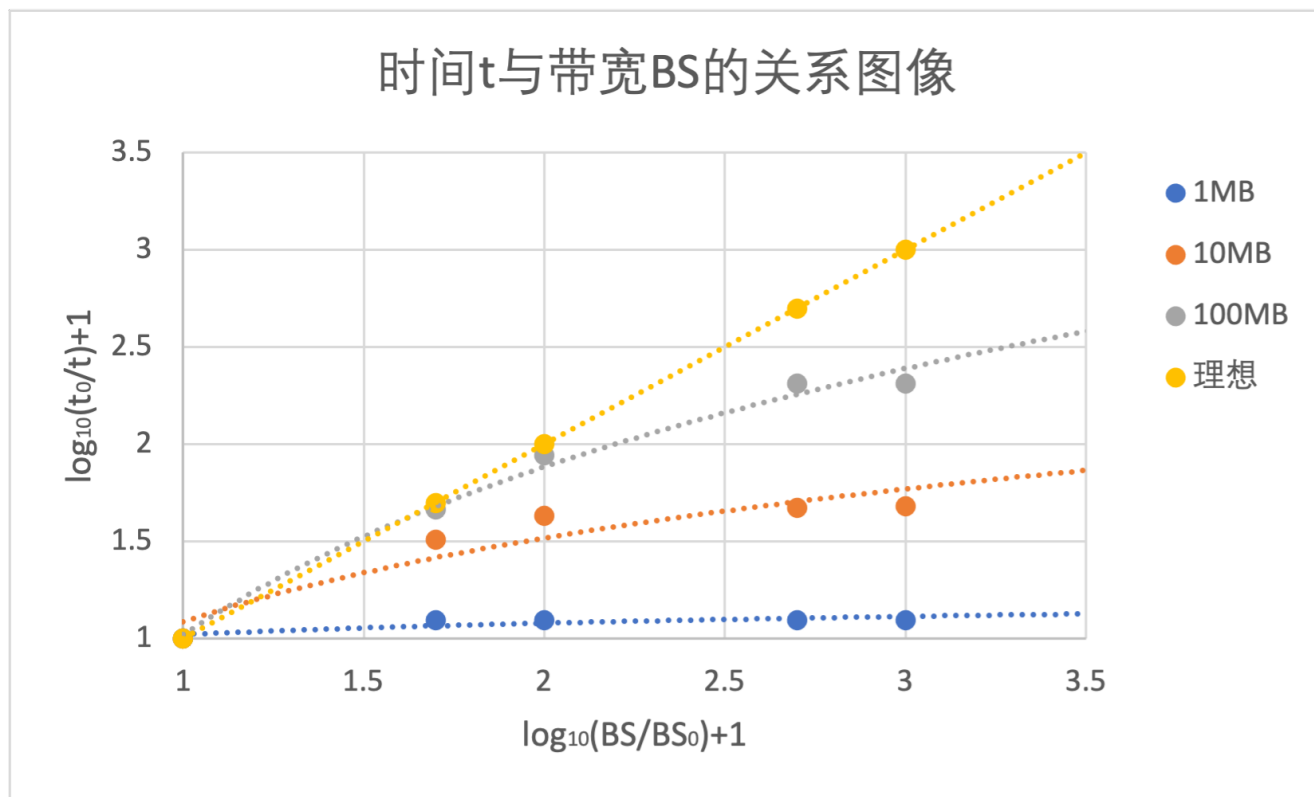
1. 互联网协议实验

- (1) 搭建实验环境：安装wireshark，搭建mininet实验环境，连接互联网，配置dns
- (2) 使用wget获取根页面www.baidu.com
- (3) 观察wireshark的输出

2. 流完成时间实验

- (1) 编写脚本fct_exp.py
- (2) 在结点 h2 分配一个数据包空间，并在结点 h1 进行抓包，统计传输时间
- (3) 修改数据包大小、延迟和带宽等参数，再次统计传输时间
- (4) 处理分析实验数据

时间与带宽关系



理想情况：

趋势线为斜率为1的直线

实际情况：

1. 实际传输时间比预计时间长很多
2. 文件越小、带宽越大，实际曲线离理想曲线越远

慢启动 & 拥塞避免机制：

慢启动算法：主机刚开始发送数据报的时先探测网络的状况。如果网络状况良好，即发送方发送的报文段可以被接受，那么就加倍发送窗口的大小。因此上图中的增长是非线性的。

拥塞窗口的大小不会无限制的增长下去，因此小数据包受慢启动机制影响会比大数据包更显著。

实验心得&与理论课的联系 ●●●●

1. 深入理解DNS协议、TCP协议、HTTP协议、ARP协议的功能和他们之间的协作。
2. 作图分析了不同带宽对传输文件所需要的时间的影响，了解了TCP协议中的慢启动和拥塞避免机制。

03 - S o c k e t



使用C语言实现HTTP服务器和HTTP客户端，具体要求包括：

1. 服务器监听80端口，收到请求并做出回答
2. 支持HTTP Get方法
3. 支持连续多次获取文件
4. 支持多个客户端获取文件
5. 支持请求不存在的文件

实验心得



HTTP报文是用于HTTP协议交互的信息。因为HTTP通信过程包括客户端往服务器端发送请求以及服务器端给客户端返回响应两个过程，所以会产生相应的**请求报文**和**响应报文**。本次实验中需要具体了解这两种报文的格式。

一开始用SimpleHTTPServer测试自己写的client时一直出错，但是用自己写的server运行时却正常，因此判断可能是自己的写法不够规范。通过wireshark抓包可以对比运行SimpleHTTPServer和自己的server时的区别，从而确定问题所在。

与理论课的联系 ●●●●

1. 了解了HTTP协议发挥作用的场合和具体的工作方式
2. 学习了HTTP报文的格式，并且在实践中加深了理解

04-Broadcast



1. 实现结点广播 broadcast_packet 函数。
2. 使用 three_nodes_bw.py 验证三个节点相互能够ping通。
3. 用 iperf 测量广播网络的效率。
4. 构建环形拓扑结构，复现广播风暴现象。

实验心得&与理论课的联系 ●●●●

1. 实现了广播函数，掌握了广播结点的工作原理
2. 加深对直连网络的理解，了解“广播风暴”这一概念，为交换网络和生成树协议进行铺垫

05-Switching



交换机在转发数据包：将数据包沿目的主机的方向转发。

转发表：存储目的地址和转出端口的对应关系

构建转发表三个关键步骤是**查询**，**插入**和**老化**。

1. 查询：根据目的MAC地址查询转发表。如果查询到则从该端口转发，否则广播。
2. 插入：根据收到该数据包的端口和数据包的源MAC地址，更新转发表中条目。
3. 老化：删除30s内未访问的条目，保持转发表简洁，提高查表的效率。

实现后用iperf测量性能，和广播网络进行比较。

实验心得&与理论课的联系



转发表存储目的地址和转发端口的映射关系，老化时间来保证条目的有效性，这两者之间的配合可以大幅度提高发包的效率。

通过本次实验，加深了对交换网络的理解。

06 - Step



1. 补全已有代码，实现生成唯一的、优先级最高的生成树的算法。
2. 该算法对于 `four_node_ring.py` 和自己构建的不少于七个结点的拓扑结构，可以输出最小生成树拓扑。在 `four_node_ring.py` 的基础上添加两个主机结点，运行完生成树程序后这两个主机结点可以 ping 通。
3. 调研标准生成树协议，考虑如何应对复杂情况和如何提高效率。

实验心得&与理论课的联系



生成树的理论比较复杂，在写代码前需要对整个结构有较为清楚的认识，减少debug的时间。

通过本次实验理解简单版本的生成树协议，并且了解实际应用中的生成树协议时如何提高效率和稳定性的方法。

遇到的问题



1. 由于网络结点的调试方式跟之前不同，所以一开始的时候并不是很适应。但是经过几次实验，已经渐渐熟悉wireshark和gcc的调试方式，提高了调试的效率。
2. 实验中最常见的报错是Segmentation Fault，因为代码中很容易忘记进行初始化，导致访问越界。
3. 在已经提供的代码框架中添加内容的时候，需要对其他很多代码进行理解，要结合整体框架进行分析，注意代码之间的衔接。

改进的建议



可以使用piazza或者sep进行问题讨论。微信群中的消息可能比较多，使用专门的提问平台，可以方便其他同学查看。



1PPT.COM

THANKS

极简通用模板

