**实验报告**

|  |  |  |
| --- | --- | --- |
| 课程名称：计算机网络实验 | 班级：软件1912 | 实验日期：2021/4/27 |
| 姓名：李越 | 学号：20192071 | 指导老师：钟表 |
| 实验名称：零比特插入 | 实验序号：02 | 实验成绩： |
| **一．预备知识**  点对点协议PPP（Point-to-Point Protocol）是目前使用得最广泛的数据链路层协   议。现在大部分用户是使用电话线拨号接入因特网的。而从用户计算机到ISP的链路所使用的数据链路层协议就是PPP协议。PPP协议必须保证数据传输的透明性。这就是说，不能限制网络层交下来的分组中不许出现某种比特组合。只要不出现传输差错，所有的数据都必须原封不动地传送到相邻结点。如果数据中碰巧出现了和帧定界符一样的比特组合时，就要采取有效的措施来解决这个问题。  PPP协议中，用在ＳＯＮＥＴ／ＳＯＨ链路时，是使用同步传输，而不是异步传输，这种情况下，PPP协议采用0比特填充的方法实现透明传输．  PPP帧格式   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Ｆ  7Ｅ | Ａ  ＦＦ | C  03 | 协议 | 信息部分 | FCS | F  7E |   标志字段Ｆ（Flag）规定为0X7E,0X1E即：01111110  具体方法：  在发送端扫描整个信息字段，只要发现5个连续1，则立即填入1个0，因此在经过这种0比特填充后，可保证信息字段中不会出现6个连续1．在接收端收到1个帧时，先找到标志字段Ｆ，以确定一个帧边界，接着再对后面的比特流扫描，每当发现5个连续1时，就把这5个连续1后的一个0删除，以还原后来的信息比特流．这样就保证了透明传输，在所传输的数据比特流中可以传送任意组合的比特流，而不会引起对帧边界的判断错误。  ①信息字段中出现了和               0101111110001010  标志字段F完全一样             会被误认为是标志字段F  的8比特组合  ②发送端在5个连续的               010011111010001010  1之后填入0比特再              发送端填入0比特  发送出去  ③在接收端把5个连续               010011111010001010  的1之后的0比特删除            接收端删除填入的0比特  二．实验目的  掌握零比特插入原理及方法  三．实验任务  1.在windows环境下编写零比特插入程序。  代码：   1. def byteInsert(arr): 2. count = 0 3. **for** i in range(len(arr)): 4. **if** arr[i] == 1: 5. count += 1 6. **if** count == 6: 7. arr.insert(i, 0) 8. **else**: 9. count = 0 10. **return** arr  13. def restore(arr): 14. count = 0 15. **for** i in range(len(arr) - 1): 16. **if** arr[i] == 1: 17. count += 1 18. **if** count == 5: 19. arr.pop(i + 1) 20. **else**: 21. count = 0 22. **return** arr  25. def main(): 26. list = [0,1,0,1,1,1,1,1,1,0,0,0,1,0,1,0] 27. print("发送端") 28. before = byteInsert(list) 29. print(before) 30. print("接收端") 31. print(restore(before))  34. **if** \_\_name\_\_ == '\_\_main\_\_': 35. main()   2.调试程序并运行。 | | |