**北京邮电大学软件学院**

**2020-2021学年第一学期实验报告**

**课程名称： 通信协议软件设计**

**实验名称： 基于SPIN的通信协议验证**

**项目完成人：**

**指导教师：**\_\_\_\_\_\_雷友珣\_\_\_\_\_\_\_\_

**日 期： 2020年 11月 25 日**

1. **实验目的**

通过本实验使学生理解协议验证的意义、理解通信协议性质、掌握基于PROMELA语言的协议建模和验证。

1. **实验内容**
2. 下载协议验证工具SPIN软件包，并在Linux操作系统环境下安装SPIN软件。
3. 对所选定的通信协议进行分析，理解协议机制和工作原理，对所选择的通信协议的PROMELA语言模型进行分析。
4. 选择需要验证的协议性质：系统不变性（Invariant）、无死锁（Deadlock）、无活锁（Livelock）、断言（assertion）等，使用SPIN工具对协议进行验证。
5. **实验环境**
6. Linux系统主机；
7. SPIN软件6.5.0版本，软件下载网址：

<https://github.com/nimble-code/Spin>

<http://spinroot.com/spin/Src/index.html>

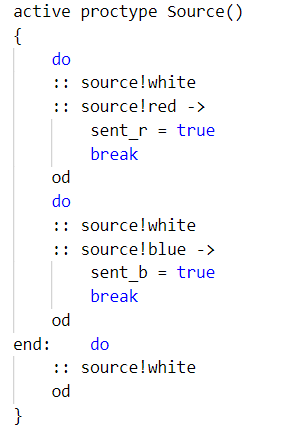
1. **实验结果**

**【协议分析】**

由于本人在小组实验一中负责回退N协议SDL描述，因此选择\Spin\Examples\Exercises\ ex\_6.pml的PROMELA语言模型进行分析。

1. **协议总体分析**

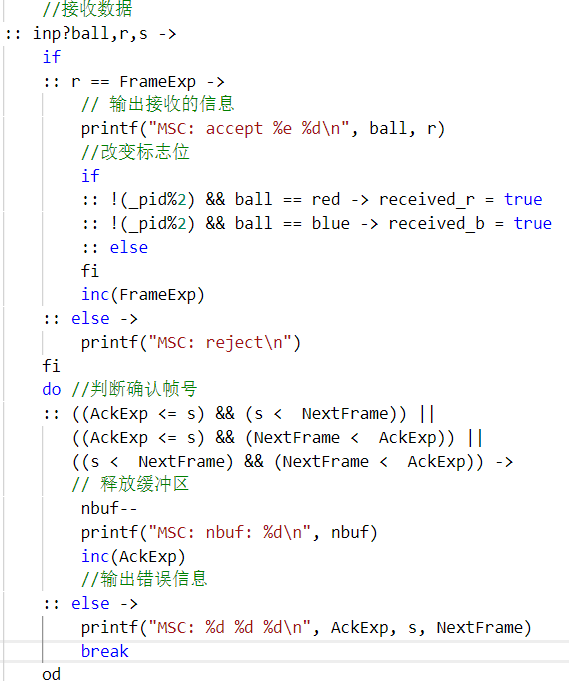
此协议模型以回退N协议为基础，实现数据链路层发送方发送方到接收方的消息传递，协议模型中共包含两个进程，proctype p5()、Source()，其中proctype p5()为数据链路层的接收方和发送方，总共由三部分组成①从网络层接收消息并发送 ②接收来自同级的数据链路层消息 ③时间超时重新发送；Source()为网络层，只要功能是发送消息下面对这两个进程进行简要分析。

1. **Source()进程：**

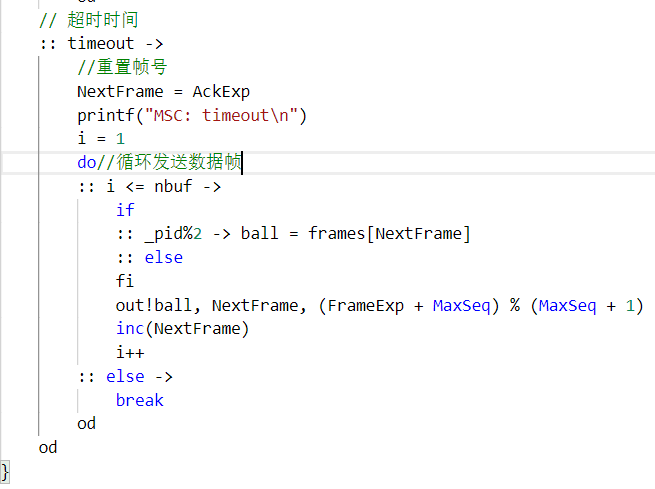
Source()进程（网络层）的主要功能是向source通道发送数据，主要由三个循环构成，第一个循环当发送的数据为red时跳出，red标志位设为true，第二个循环当发送数据为blue时跳出，blue标志位设为true，第三个循环包含end标记，表示这为结束状态，source()进程无线循环向source发送white。

1. **p5()进程**

p()进程(数据链路层)总体是一个大循环，然后再三个部分进行循环，第一部分解析：当nbuf<MaxSe，即帧缓冲区未满时，nbuf++，如果当前进程的pid%2不为0，则从source通道接收数据。之后将对应的帧号存入缓冲区，帧数据发送到out通道，NextFrame改变，第一部分结束。



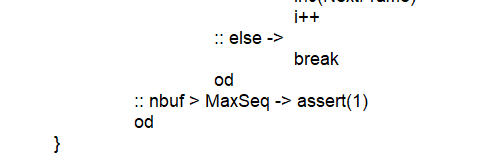
p()进程第二部分解析：当循环时能够从inp通道接收数据，则进入此状态，从inp通道接受数据后，如果接收的数据帧号等于期待的确认帧号，则输出接收的信息，如果进程pid为0，且接收的数据为red或blue，则将对应的标志位设为true，FrameExp改变，如果不等于确认帧号，则输出拒绝信息，之后开始循环判断确认帧号，如果确帧号满足条件，则nbuf--，即释放一个缓冲区，输出当前缓冲区内帧的数量，AckExp改变，如果不满足条件则输出对应信息，跳出循环第二部分结束。



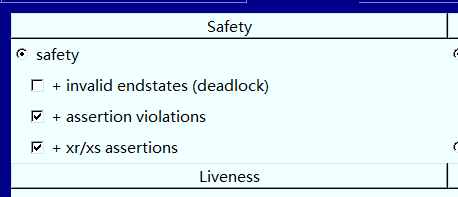
p()进程第三部分解析：当进程循环时检测到前两个部分不可进入时，帧超时事件触发，进程重置下一发送帧号，输出帧超时消息，然后开始循环发送之之后所有的帧信息到out通道，当缓冲区中最后一帧被重新发送后，跳出循环。

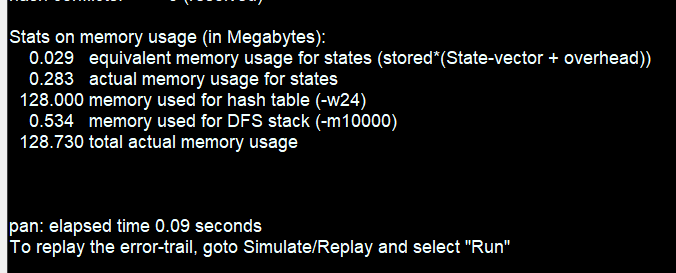
**【系统不变性验证】**

* 在此程序p5()进程最后中添加语句: （用于检测程序中帧缓冲区大小不能超过MaxSeq），如果出现nbuf>MaxSeq 的情况，则程序运行时会阻塞。



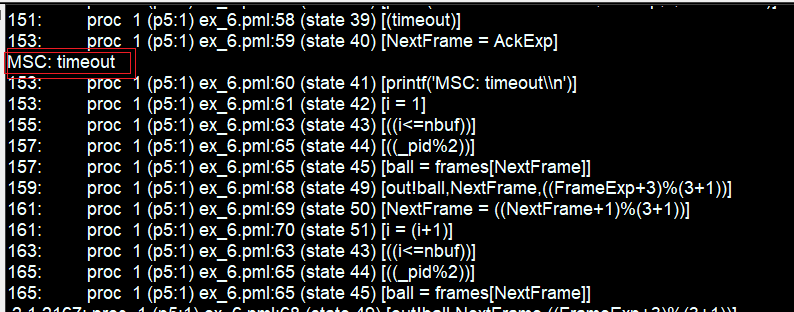
运行结果：





系统检测后，没有错误发生，说明nbuf永远小于MaxSeq，此程序满足回退n协议中发送窗口大小小于最大帧号。

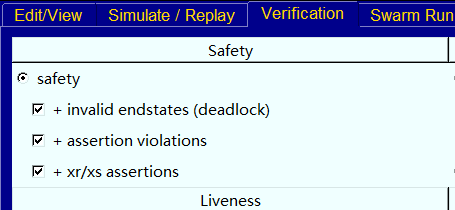
* 运行程序，检测timeout事件是否会发生

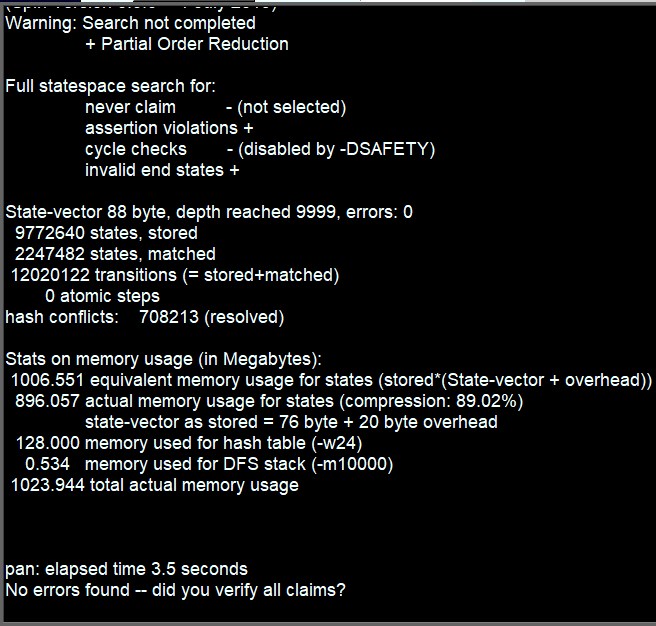


timeout事件发生，当超时事件出现时，程序会进行帧的重发。

**【无死锁验证】**

在ispin中点击Verification选项，然后在safety区域中勾上invalid endstates（deadlock）选项，进行程序死锁检测，击run查看检测结果。

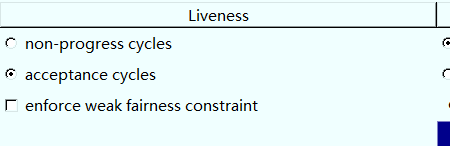


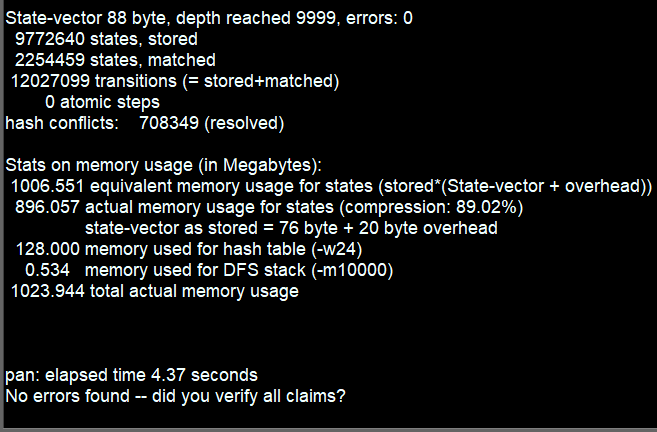


通过安全性检测，发现没有错误发生。因此断定此程序无死锁。

**【无活锁验证】**

在Liveness区域，点击acceptance cycles，使得对于标记为end的循环忽略，之后运行程序查看检测结果。

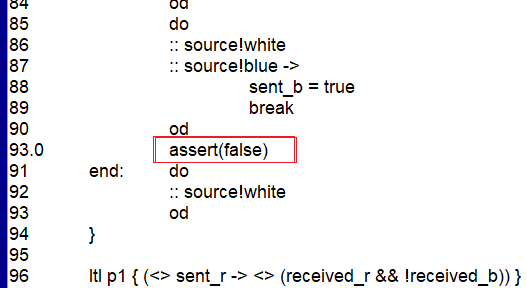


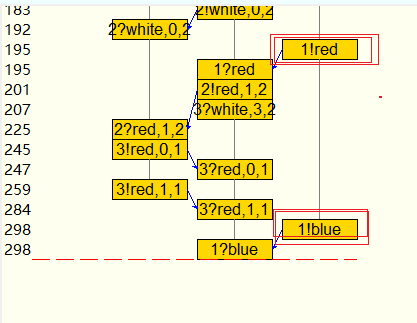


结果显示，没有错误发现，说明除了标记为end的无限循环，没有其他的死循环，因此此程序无活锁。

**【断言测试】**

在Source()进程发送完red和blue数据后添加断言，检测Source()能否将red和blue数据发送出去。





运行程序发现运行阻塞，而且是在Source()进程发送完red和blue数据后发生，说明系统能检测到断言。

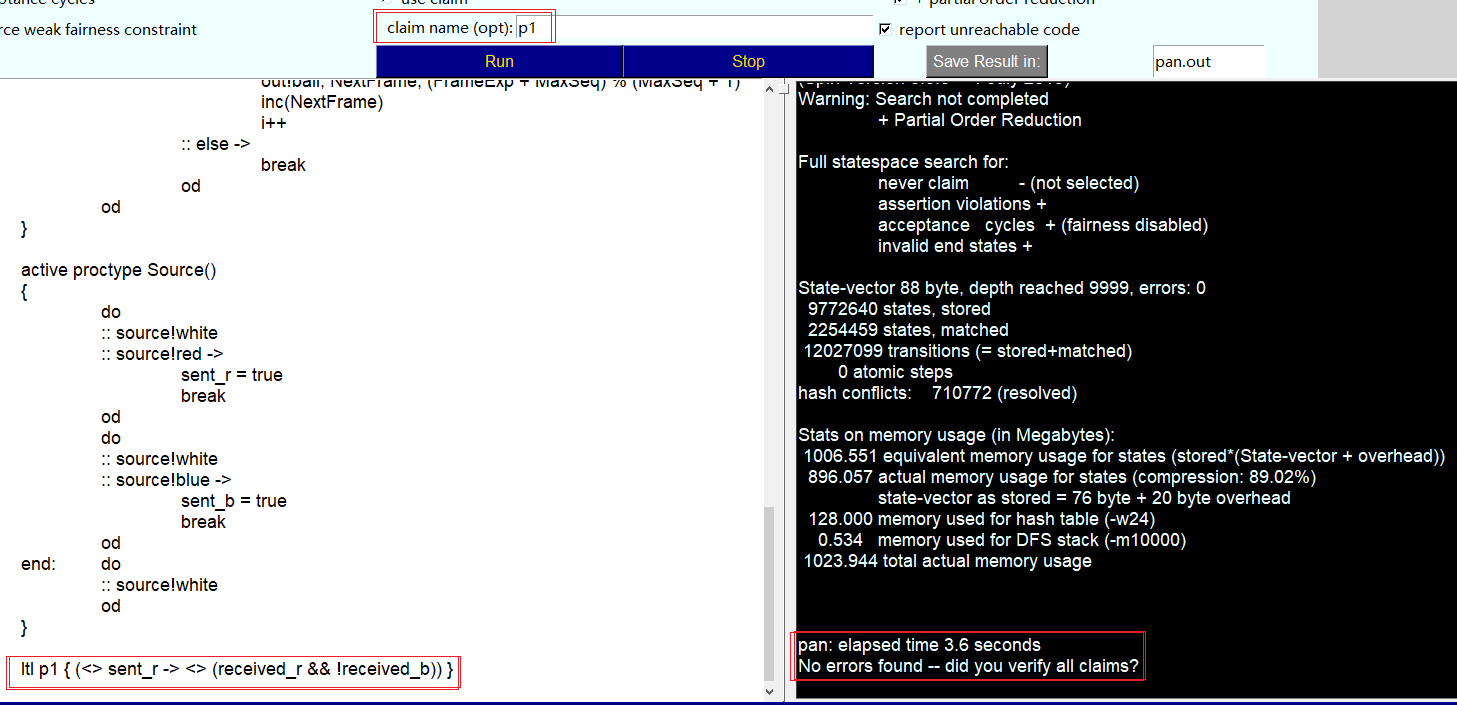
**【LTL测试】**

使用的2条LTL语句对所实现协议的性质进行验证，得到如下结果

1. 检测red和blue数据是否有序发送

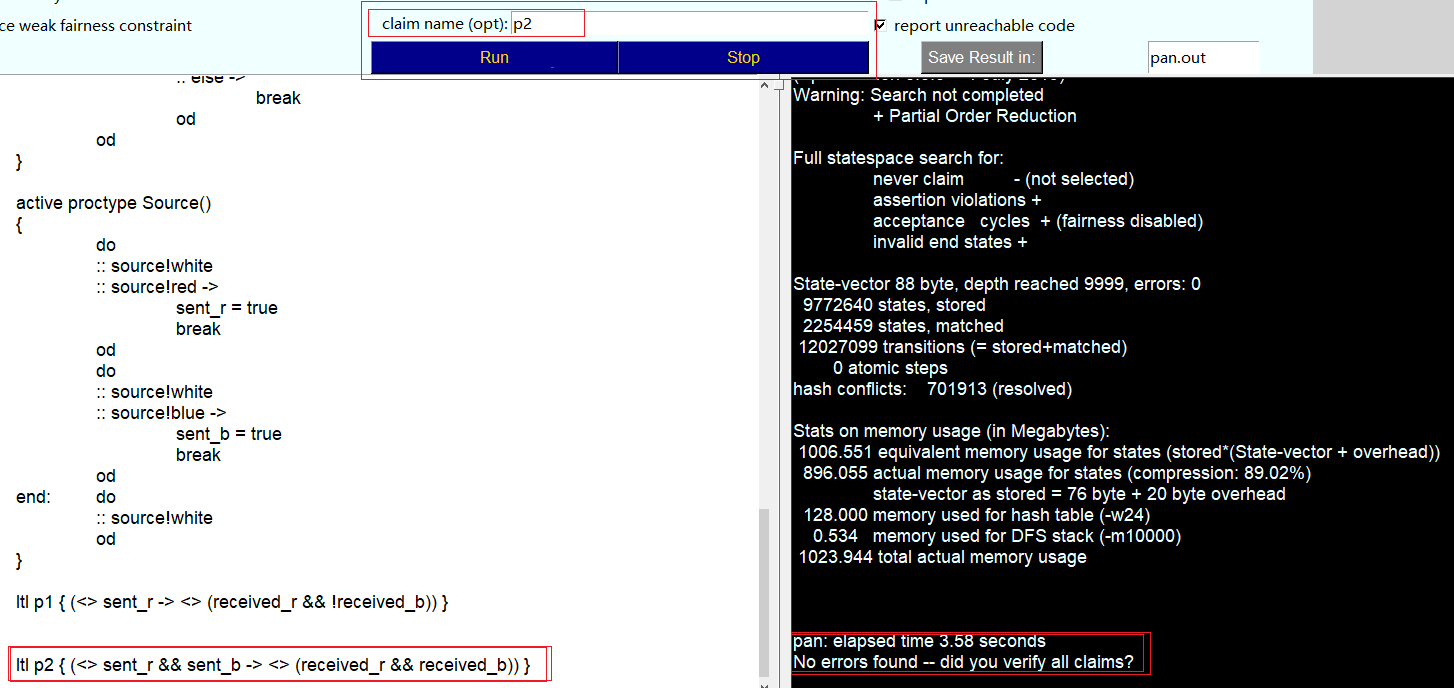
red先接收，blue后接收

ltl p1 { (<> sent\_r -> <> (received\_r && !received\_b)) }



没有发现错误，说明数据是有序发送的。

1. 检测red和blue数据是否被最终被正确接收



没有发现错误，说明red数据和blue数据最终都被发送和接收。

1. **实验心得**

在本次实验中，我学习到了如何对Promela语言描述的通信协议使用ispin软件对进行调试和模型验证，通过对回退N帧协议的模拟和验证，我们对通信协议的实现有了更深层次的了解，对Promela语言更加熟悉。