

北京邮电大学软件学院
2015-2016 学年第二学期实验报告

课程名称： 通信软件设计

项目名称： 基于 Telelogic Tau 集成开发环境的
计算机网络协议的分析、设计和验证

项目完成人：

姓名： 陈一曲 学号： 2013212098

姓名： 吴征航 学号： 2013212091

姓名： 傅 滢 学号： 2013212108

姓名： 孔丹晨 学号： 2013212113

姓名： 佟 鑫 学号： 2013212044

指导教师： 雷友珣、孙艺

日 期： 2015 年 12 月 29 日

一、实验目的

通过本实验初步掌握采用形式化描述语言分析和设计通信软件的基本方法,加深对计算机网络协议和协议软件的理解和掌握。

二、实验内容

1. 选定某一数据链路层协议,应用 Telelogic Tau 集成开发环境工具对该数据链路层协议实体软件进行需求分析、设计和验证。也可选择某一网络层协议或 TCP 协议对该协议的协议实体软件进行分析与设计。
2. 用 MSC 图描述该协议实体软件与外界环境之间的所有可能的交互场景。
3. 用 SDL 图描述该协议实体软件的设计。
4. 使用 Telelogic Tau 集成开发环境中的仿真和验证工具对所设计的协议实体软件进行仿真和验证,修改设计中的问题。

三、实验环境

项目	版本
操作系统	Windows 10 Pro.
虚拟机	VMware Workstation 12
虚拟机中操作系统	Windows XP
编译环境	VC++ 6.0
开发工具	Telelogic Tau 4.3

四、实验结果

1. 本小组首先对数据链路层支持双向通信、具有差错控制功能和流量控制功能的协议进行调研,并将滑动窗口(A Protocol Using Selective Repeat)中的一位的滑动窗口协议(A One-Bit Sliding Window Protocol),回退 N 协议(A Protocol Using Go Back N),以及选择性重传协议(A Protocol Using Selective Repeat)进行对比,并最终确定了对回退 N 协议进行数据链路层的设计、分析与验证。

2. 使用 MSC 图对协议进行描述

2.1 正常情况

发送方数据链路层在没有分组到来的时候一直保持等待状态,一旦接收到上层(网络层)发送来的数据,就离开等待状态并把网络层的数据包装成帧并将帧依次发送给接收方数据链路层,发送方发送窗口中缓存着已经发送出去但还未收到确认的帧。

接收方数据链路层收到发来的帧后，将收到的帧去掉帧头传给接受方的网络层，并启动一定时器：

- 若在定时器超时前，接收方网络层也有数据要传送给发送方，那么接收方的 ACK 确认消息会被封装在接收方发送给发送方的数据帧的帧头中进行捎带确认。
- 若是接收方的定时器超时，那么接收方会向发送方传送一个单纯的 ACK 确认帧。发送方一旦受到 ACK 确认消息后，就会将发送窗口移动一格。

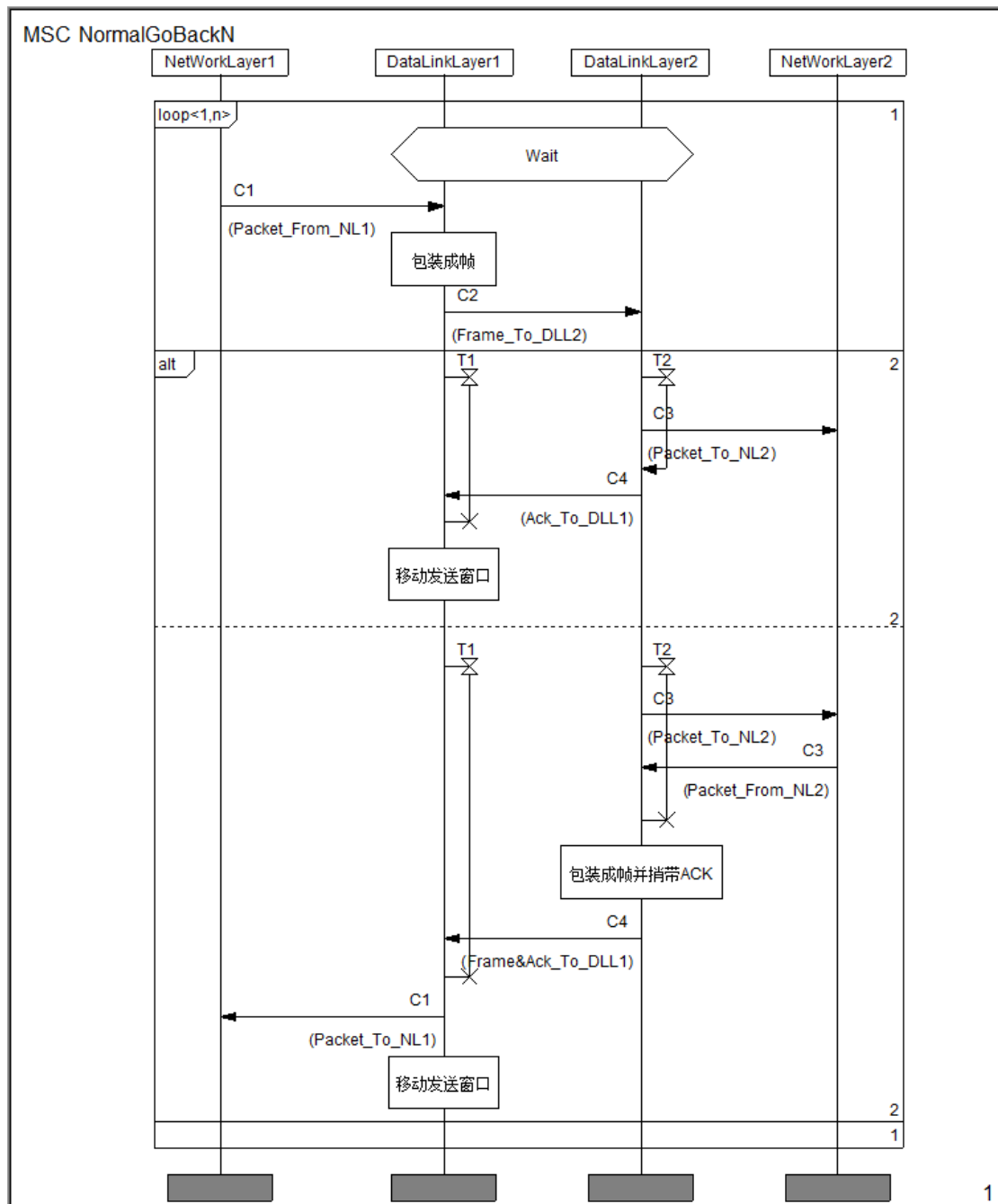


图 1.正常情况下的回退 N 协议

2.2 发送方滑动窗口满

当发送方的网络层发送数据包的速度过快, 或者由于接收方迟迟没有向发送方发出确认帧或数据帧导致发送方滑动窗口填满。则数据链路层的发送方会传送给网络层一个信息, 通知网络层该滑动窗口已满, 并拒绝接受网络层传给它的新数据包。直到接收方向发送方发出确认帧或者传送数据, 则发送方窗口向前移动并准备接受网络层传送给它的新数据包。

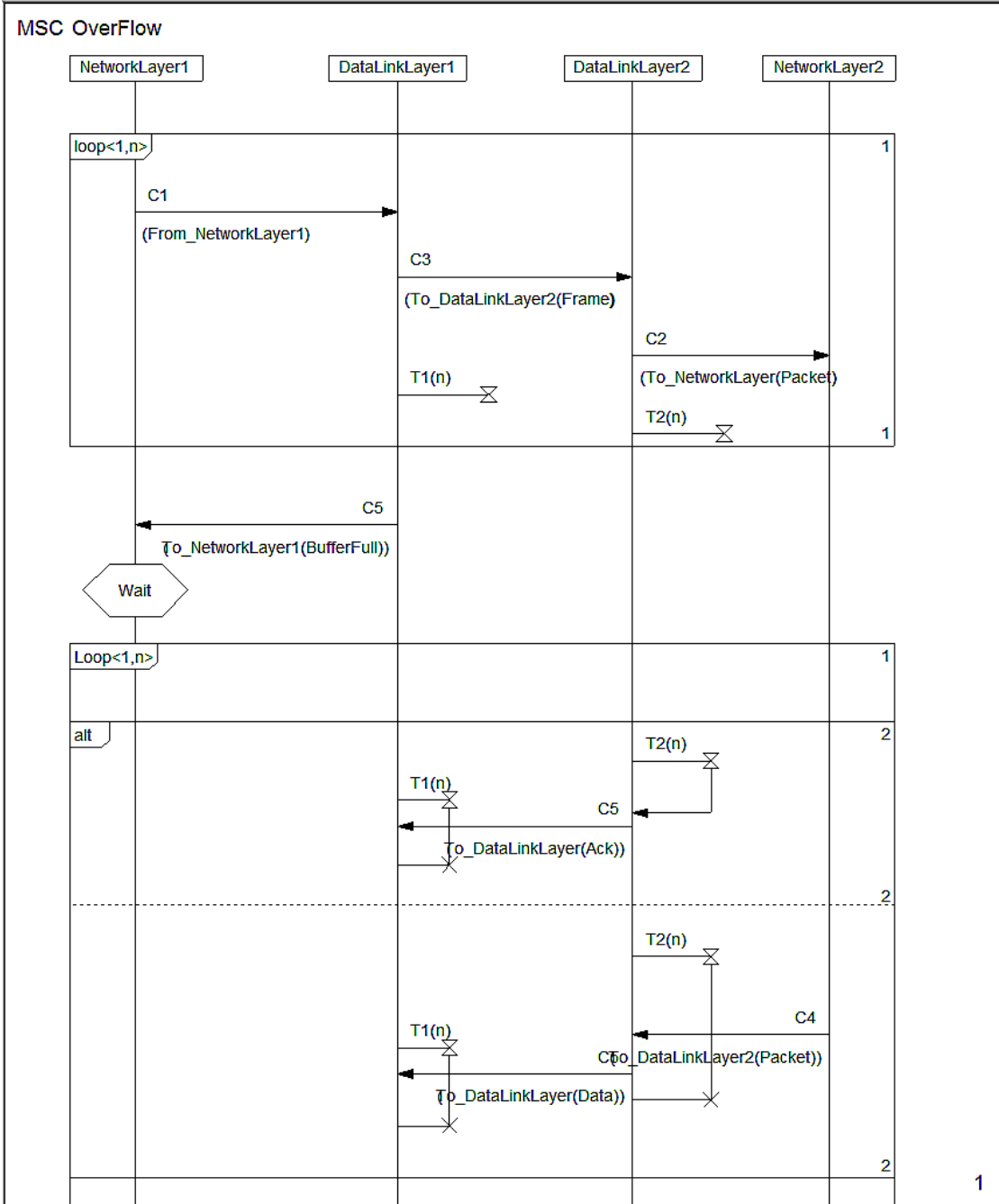


图 2. 发送方滑动窗口满-1

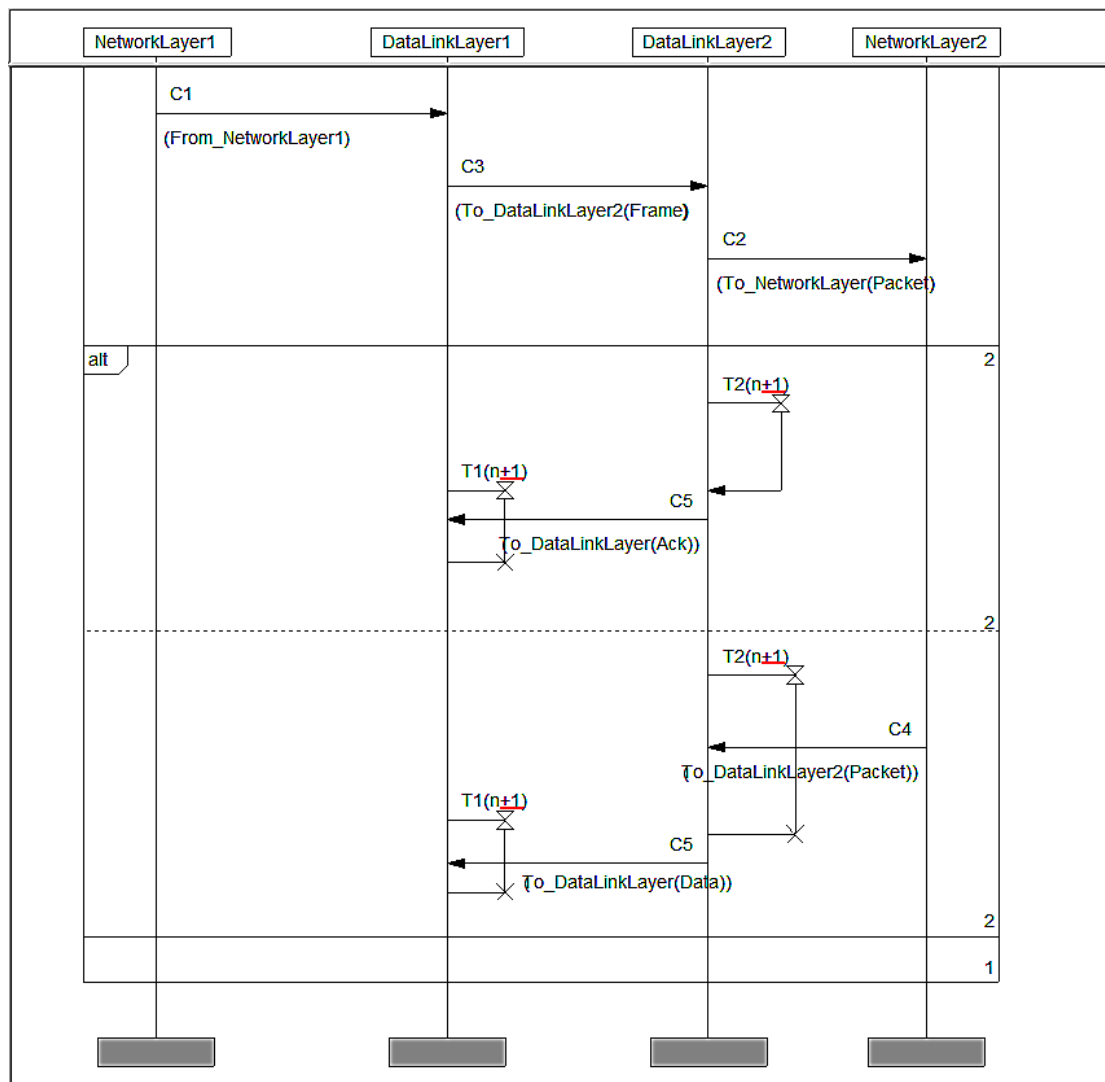
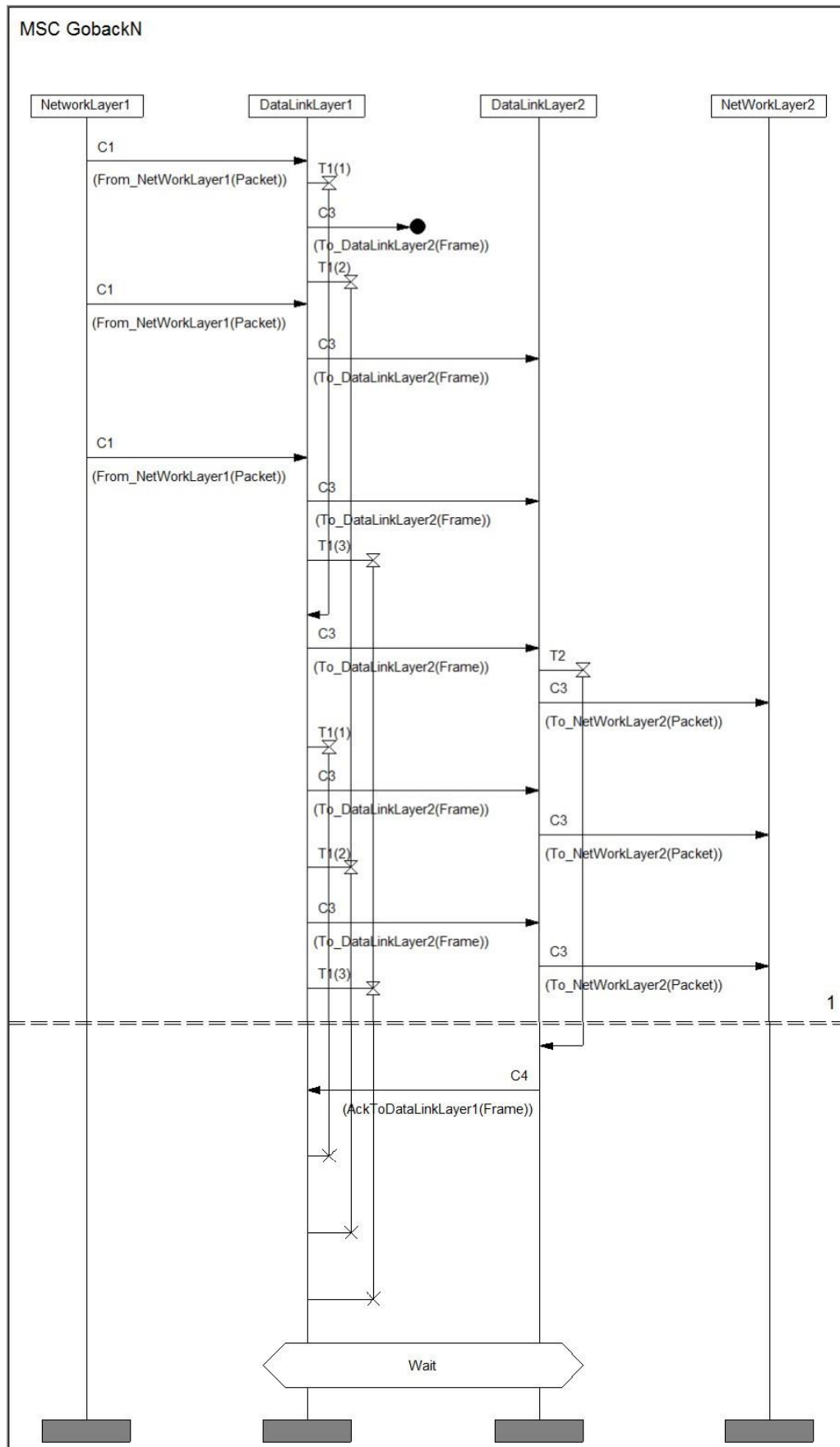


图 3.发送方滑动窗口满-2

2.3 丢失一帧

发送方的数据链路层给接收方的数据链路层发送一帧后帧丢失，接收方的数据链路层没有收到一帧，所以不会向发送方传送 Ack 或数据，导致发送方的数据链路层计时器超时，发送方重发该帧。



2.4 ACK 丢失

接收方返回给发送方的 ACK 消息意外丢失后，发送方的定时器会有超时动作，然后发送方会发送窗口中最早一个未被确认的帧开始，将发送窗口中缓存着的帧依次发送给接收方。

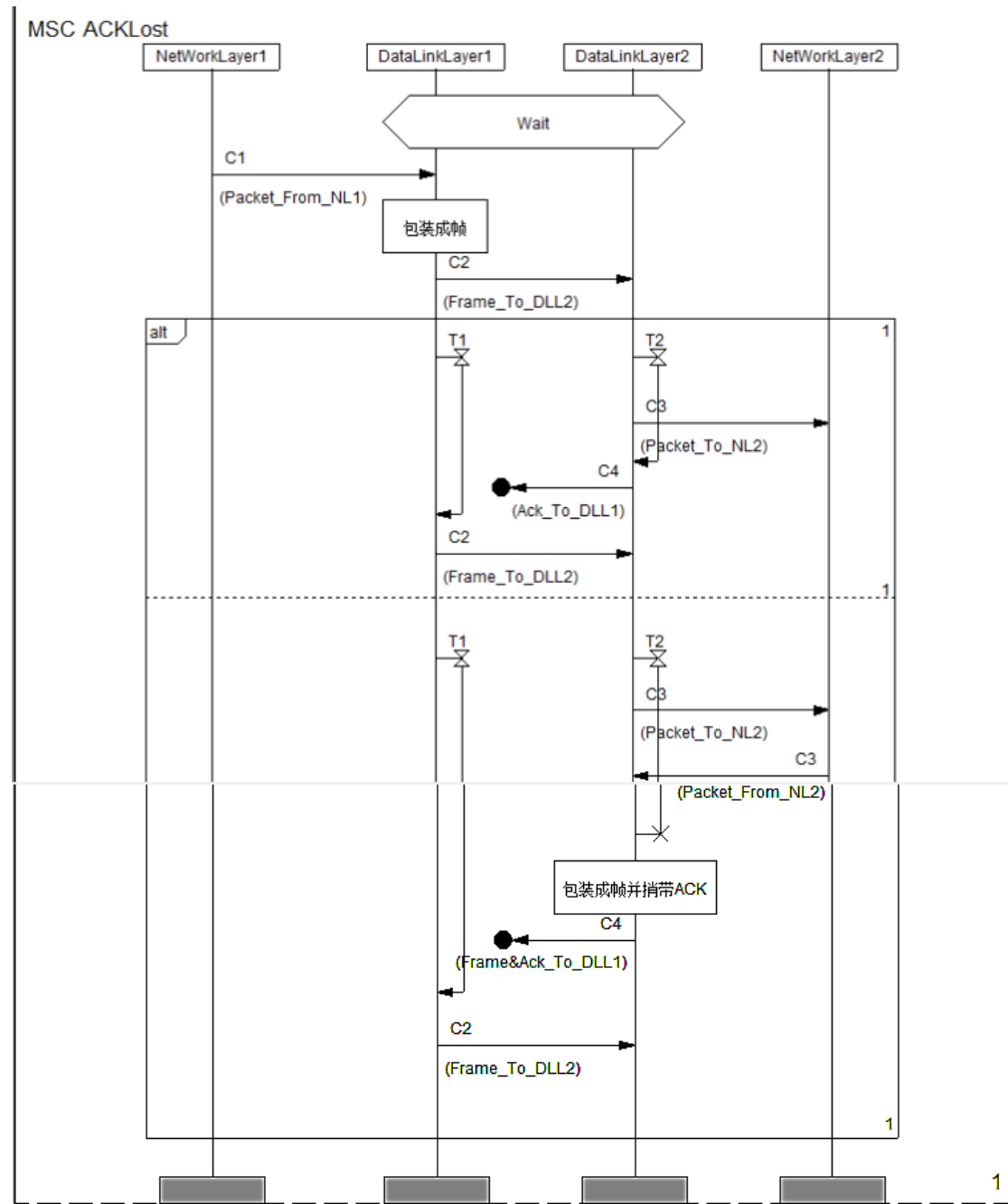


图 5.ACK 丢失

3. SDL

3.1 管理进程变量声明页

本小组设计的数据链路层 PDU(Protocol Data Unit)格式如下：

NEWTYPE Frame STRUCT

```
Info  Packet; /* 来自网络层的包*/
Seq   Integer; /* 该帧的序列号*/
Ack   Integer; /*该帧携带的确认号*/
```

ENDNEWTYPE;

其中网络层传送给数据链路层的包的格式为：

NEWTYPE Packet STRUCT

```
Data charstring; /*包要传送的内容*/
Flag integer; /*该包携带的数据是否有效*/
```

ENDNEWTYPE;

本 SDL 图使用到的信号有：

ToDL1(Frame), /*发给数据链路层 1 的帧*/

ToDL2(Frame), /*发给数据链路层 2 的帧*/

FromNW1(Packet), /*来自网络层 1 的包*/

FromNW2(Packet), /*来自网络层 2 的包*/

AckToDL1(Frame), /*发送给数据链路层 1 的确认帧*/

AckToDL2(Frame), /*发送给数据链路层 2 的确认帧*/

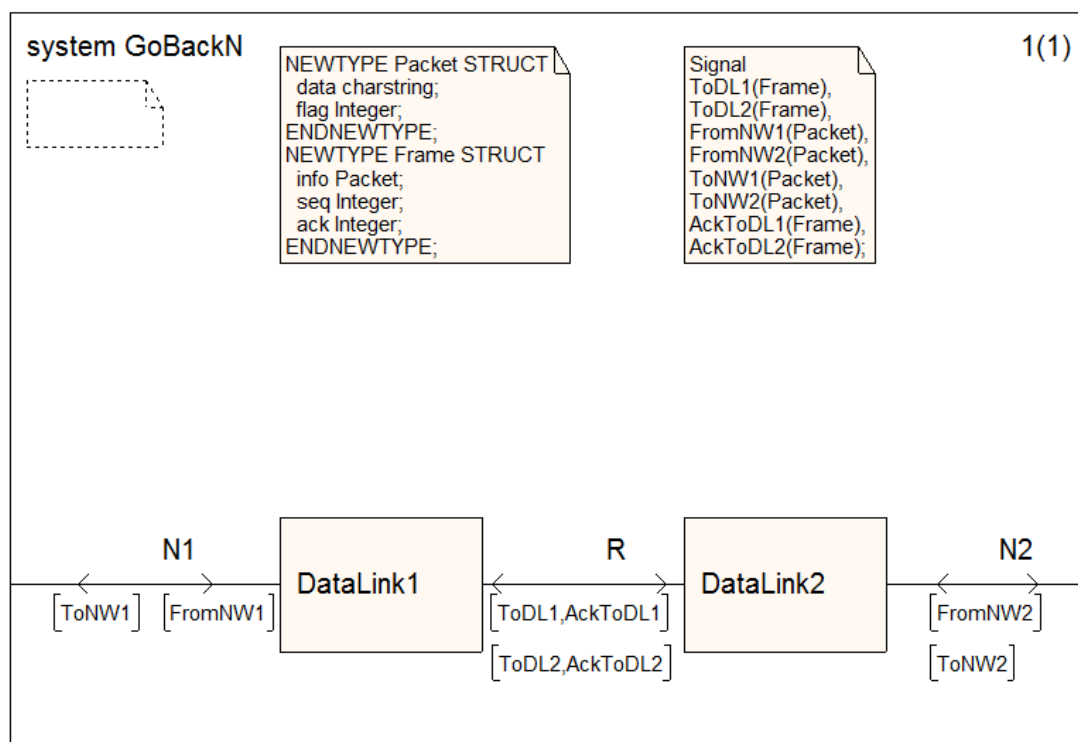


图 6.系统变量声明页

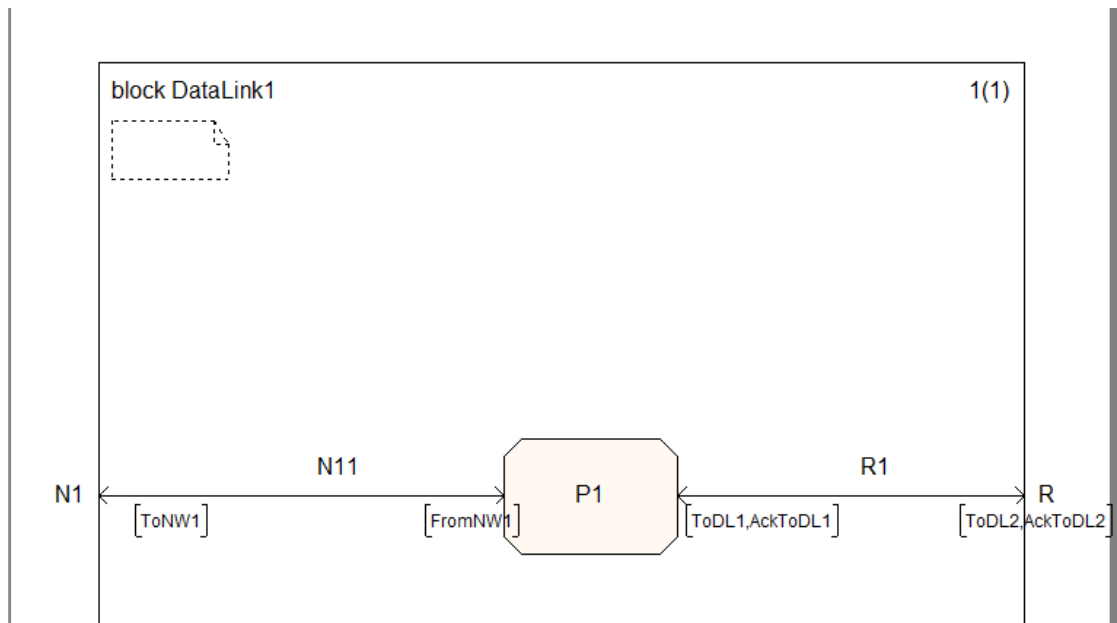


图 7. 数据链路层功能块图

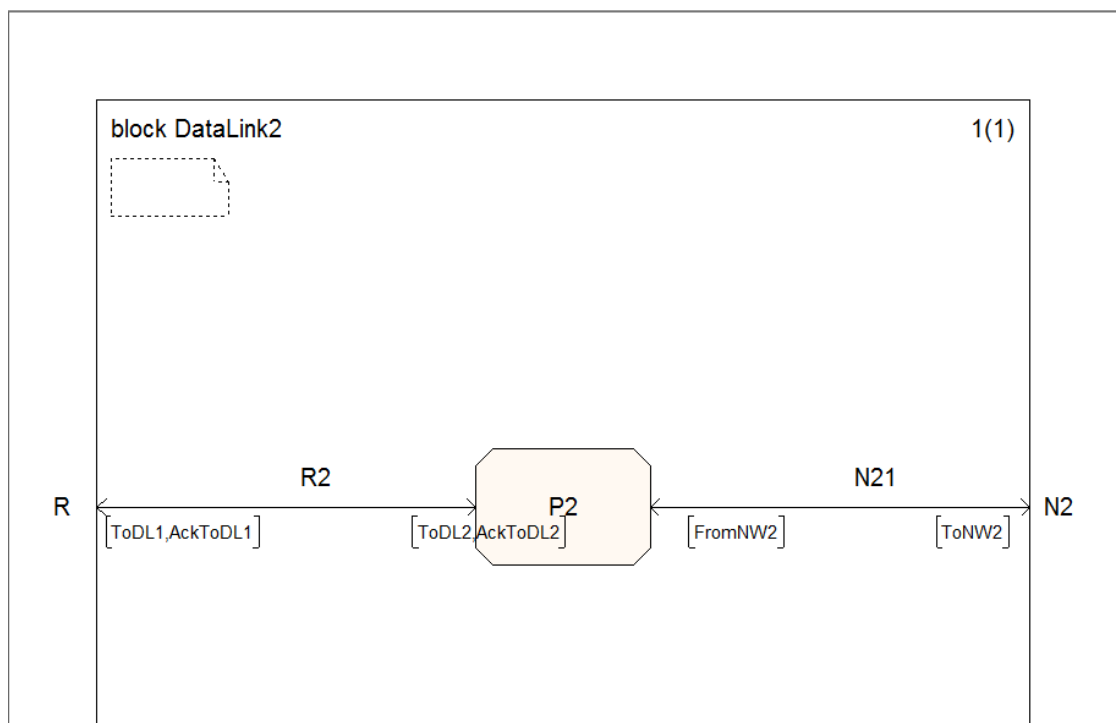


图 8.数据链路层 2 功能块图

P1 进程主要负责

- 接收来自网络层的包
- 接受数据链路层 2 给数据链路层 1 的 Ack 消息
- 接受数据链路层 2 给数据链路层 1 的数据消息

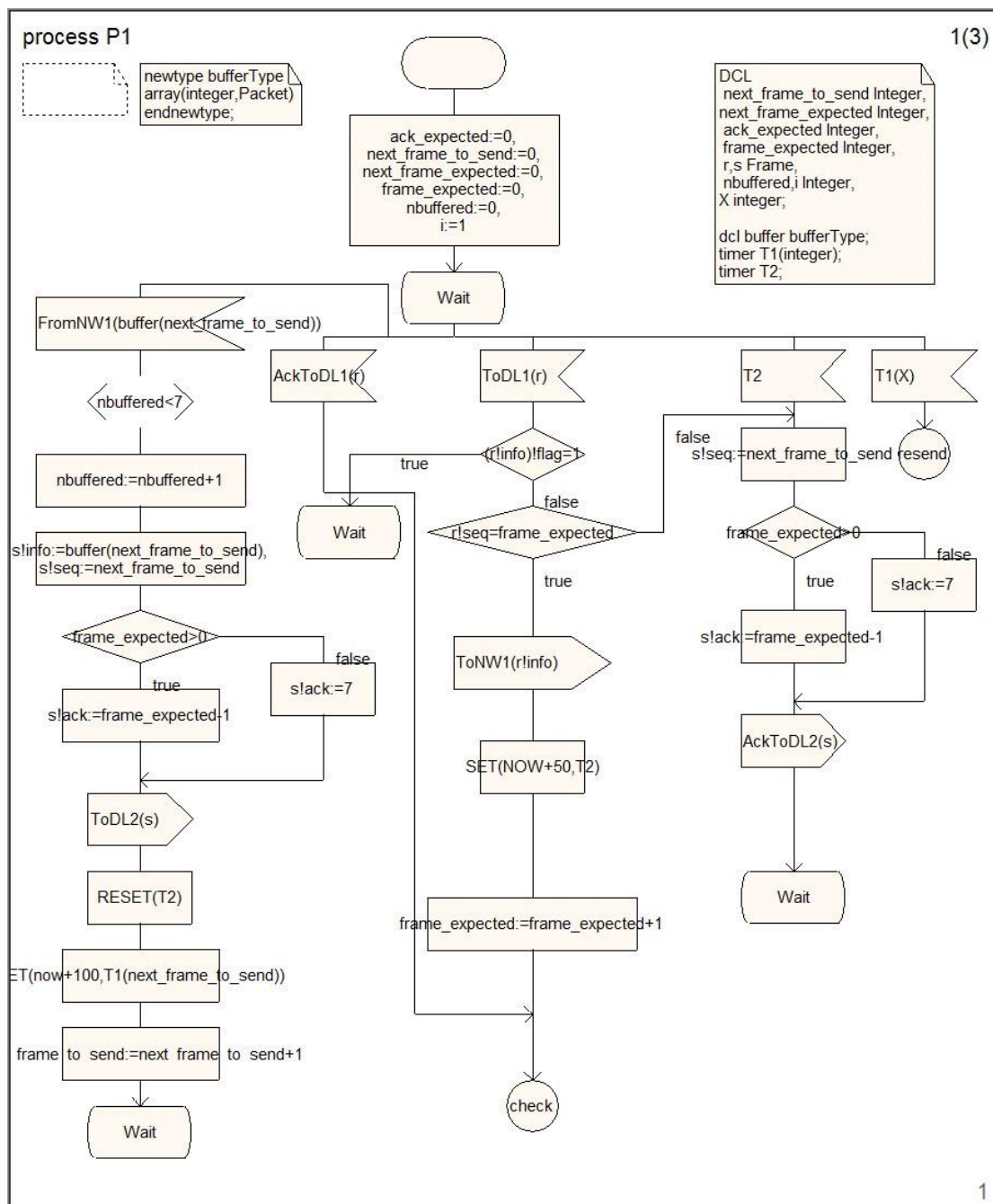


图 9. 进程图 P1

P2 进程主要负责

- 接收来自网络层 2 的包
- 接受数据链路层 1 给数据链路层 2 的 Ack 消息
- 接受数据链路层 1 给数据链路层 2 的数据消息

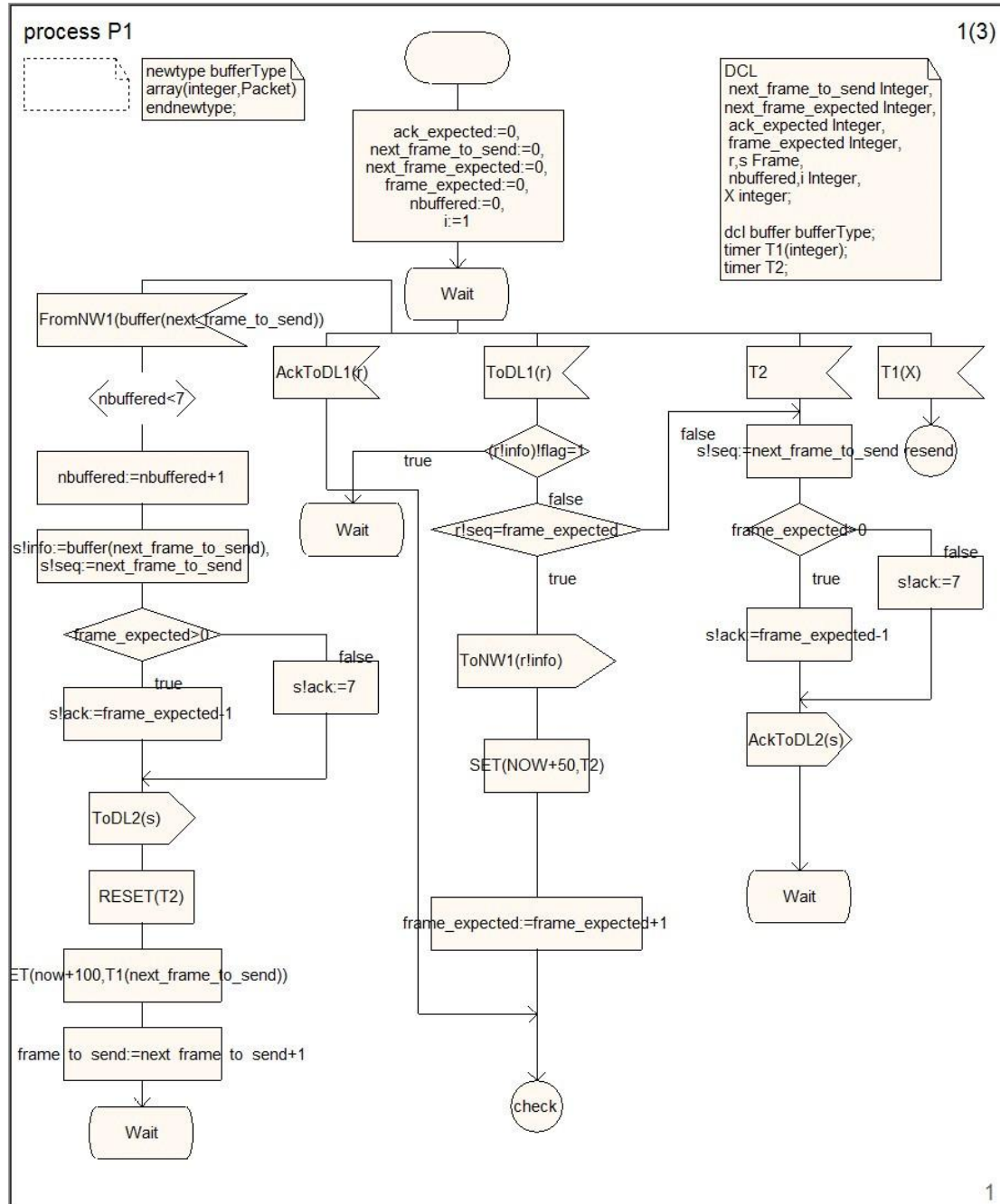


图 10.进程图 P2

4. 协议验证

4.1 正常情况

发送方的网络层使用信道 FromNW1 发给数据链路层 7 个包，包中携带信息的格式数据为 ['Data',0] 其中 Data 表示要传送的数据，0 表示该包是一个有效包；并将这些包发送给 P1 进程。发送方的数据链路层将收下的包依次发送给接收方的数据链路层，期间接收方的网络层并没有消息回复，接受方计时器超时，所以只发 Ack 消息回复给发送方的数据链路层。因为向前确认的存在，所以只确认最后一个帧携带的序列号，即 Ack(6)。

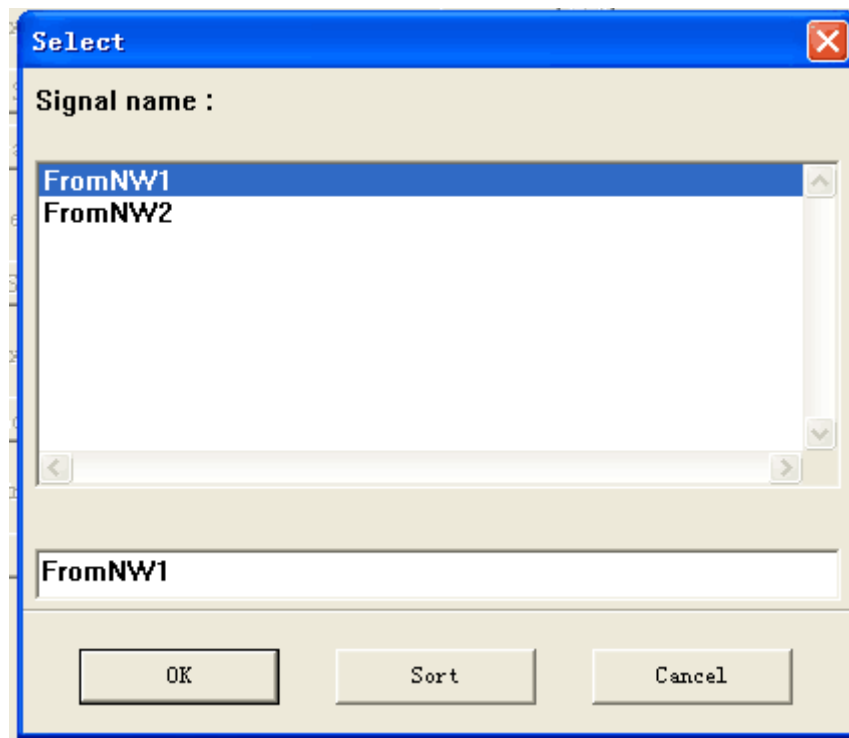


图 11. 网络层 1 发给数据链路层 1 的信道

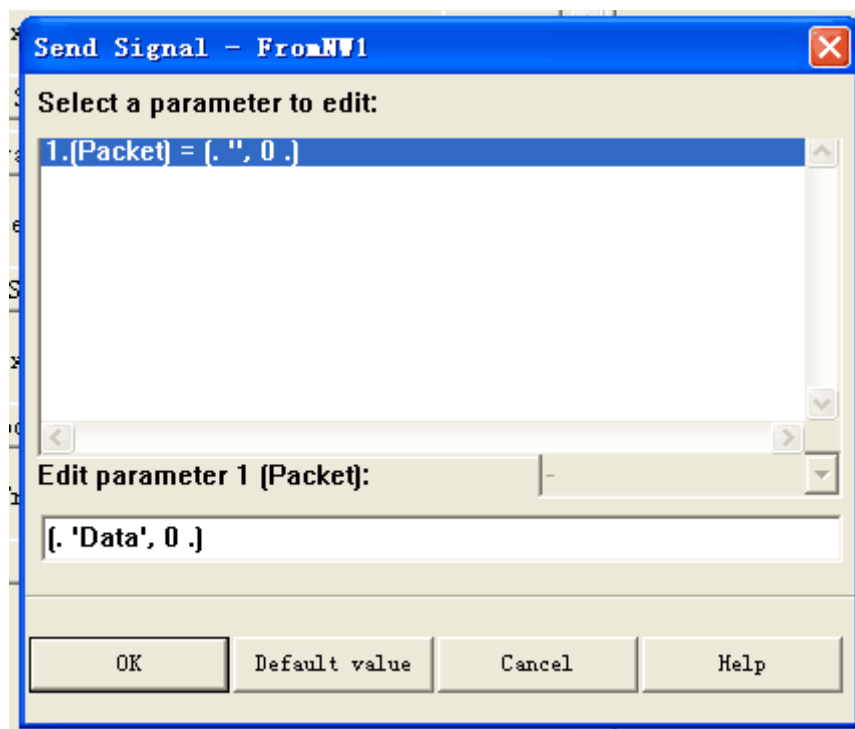


图 12. 网络层要发的包格式

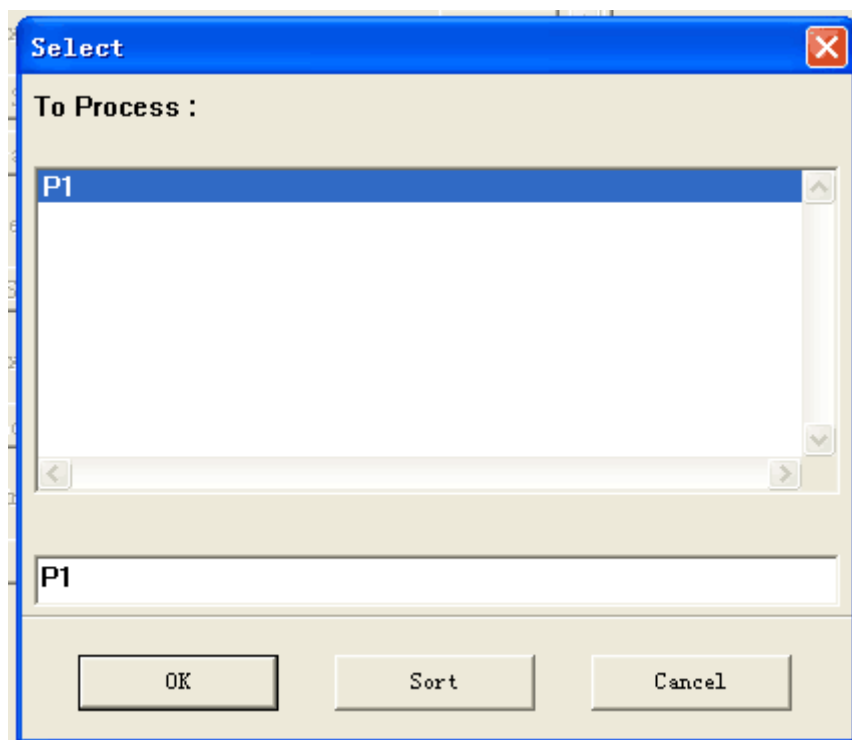


图 13. 将包发送给 P1 进程

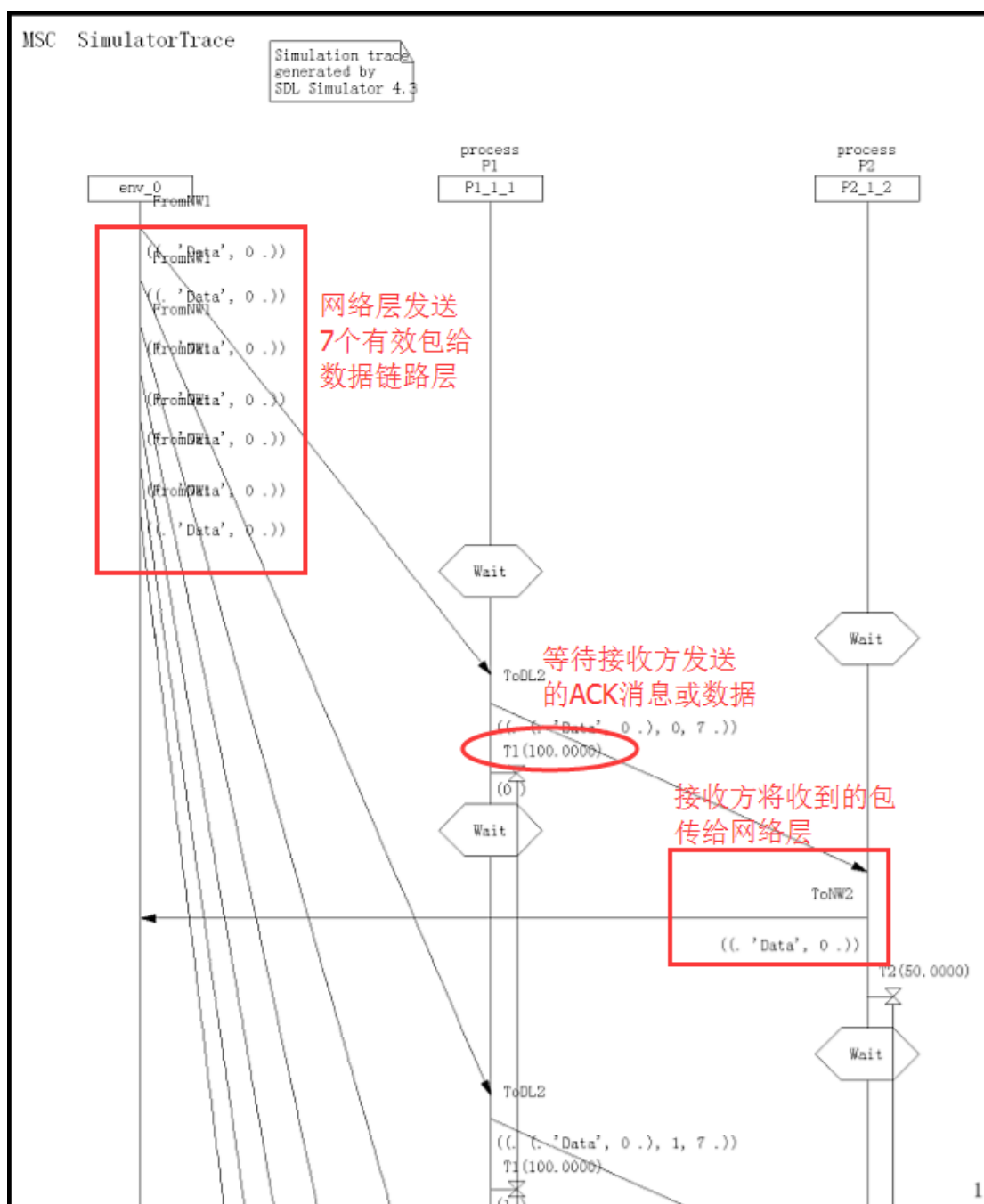


图 14.正常情况下发送 7 个包-1

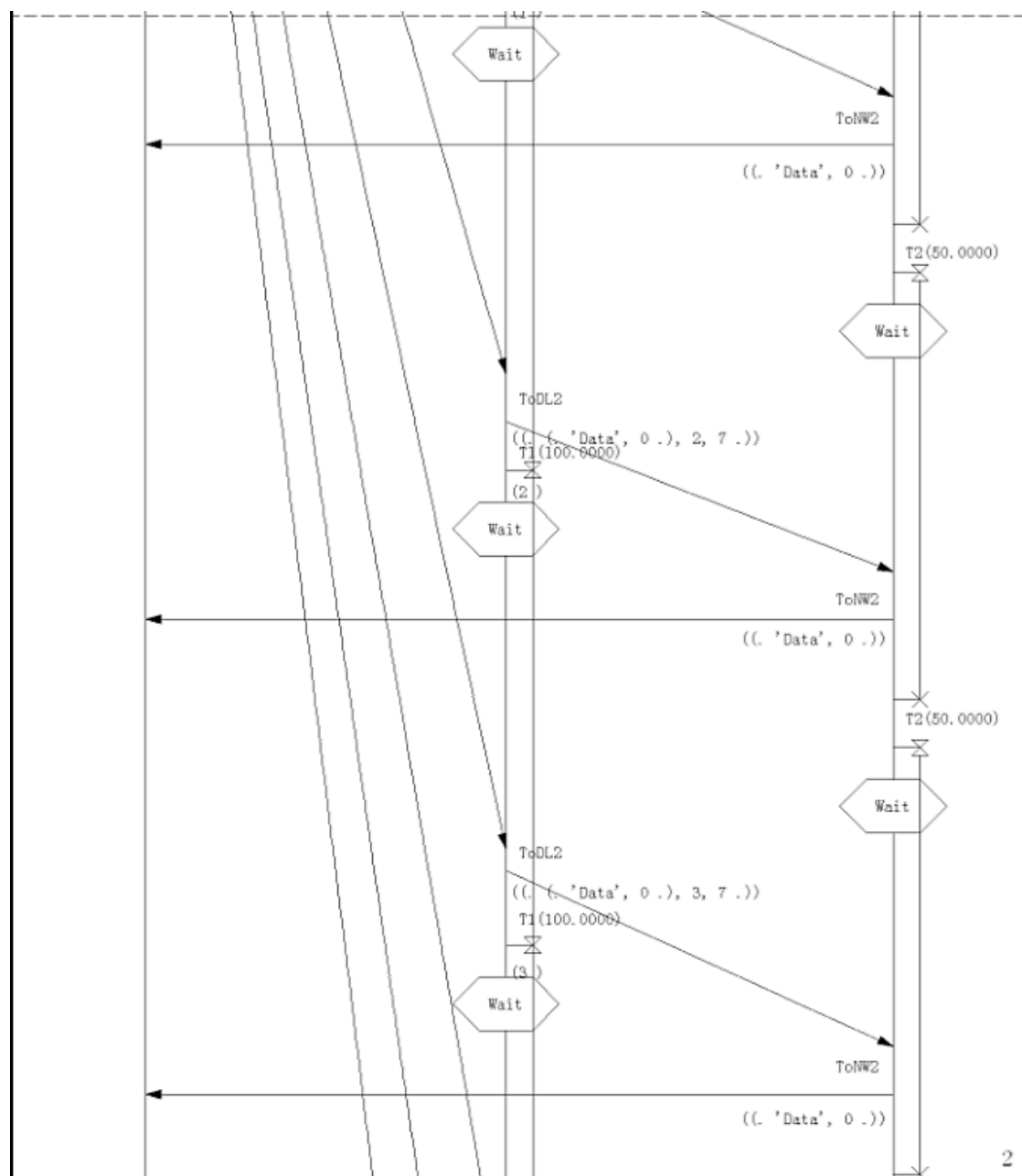


图 15.正常情况下发送 7 个包-2

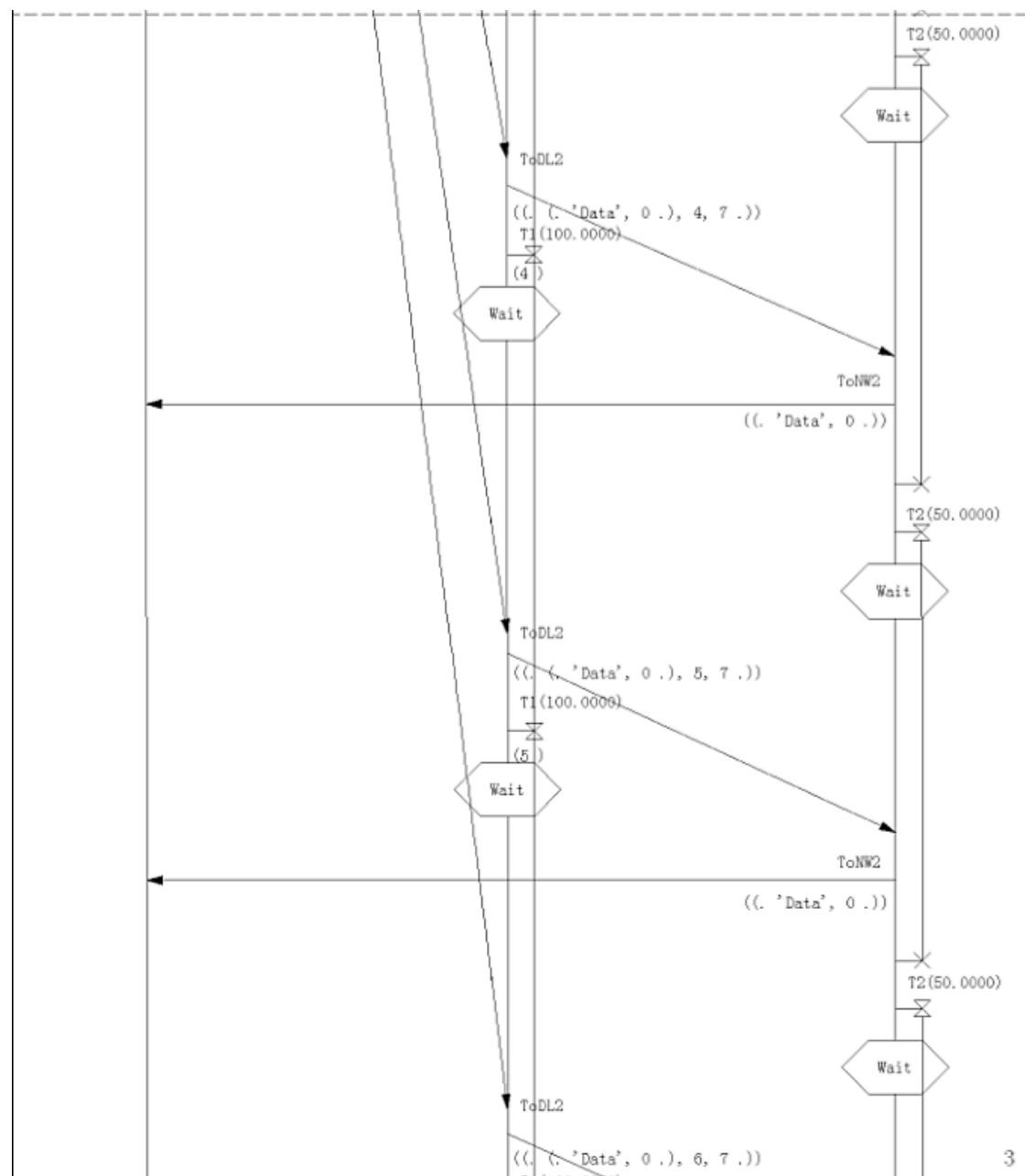


图 16.正常情况下发送 7 个包-3

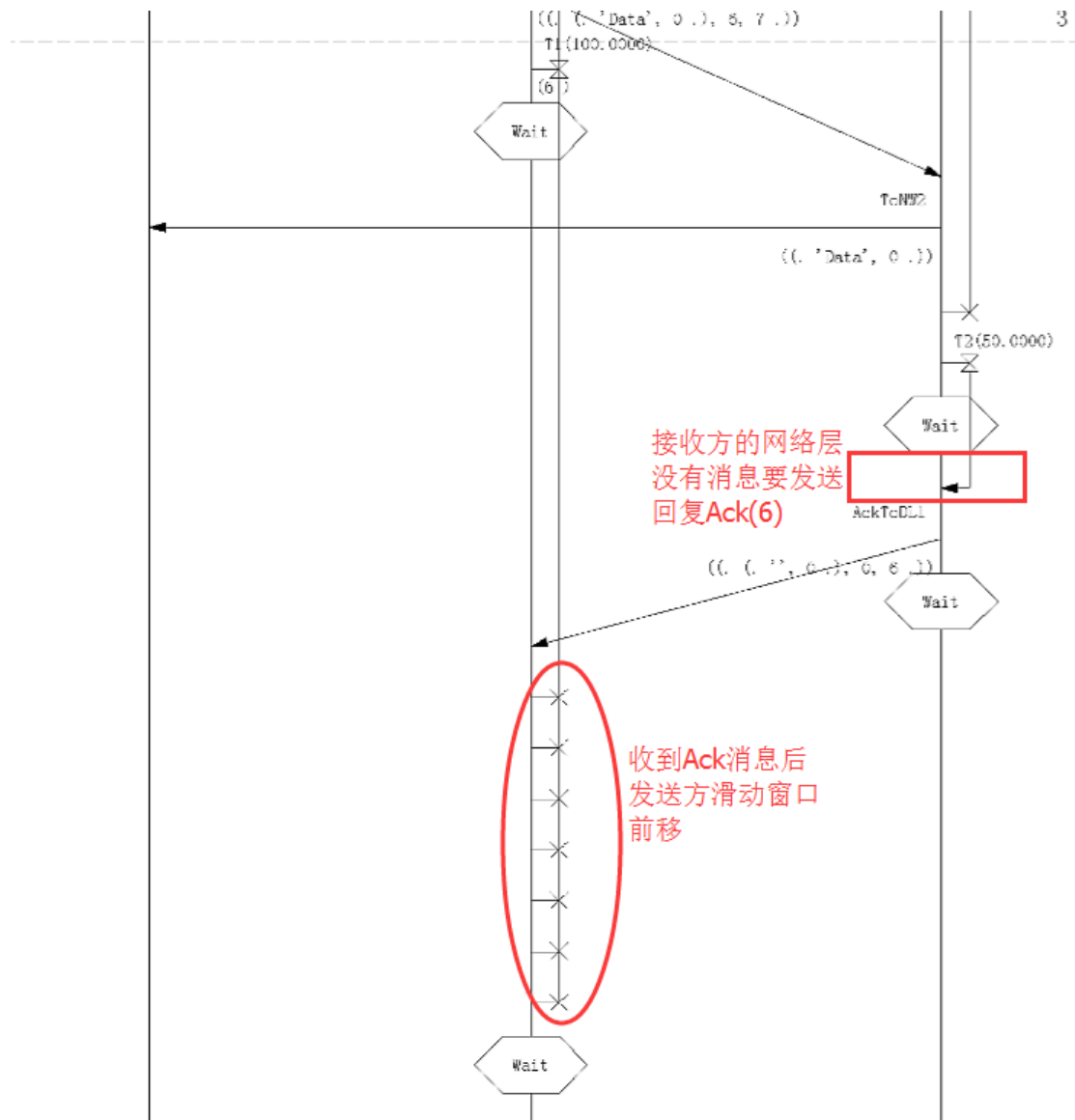


图 18. 正常情况下收到 7 个包

4.2 接收方网络层有数据要发送

当发送方发送一个帧给接收方时，接收方有数据要发送，则进行捎带确认，即接收方将数据与 Ack 消息一起发送给发送方。

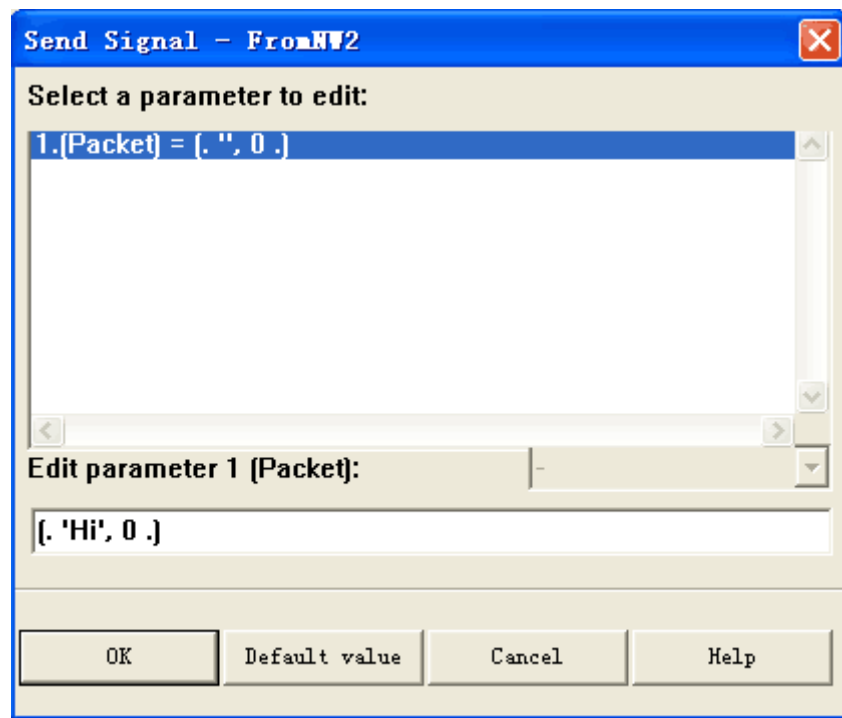


图 19.接收方的网络层有数据要发送

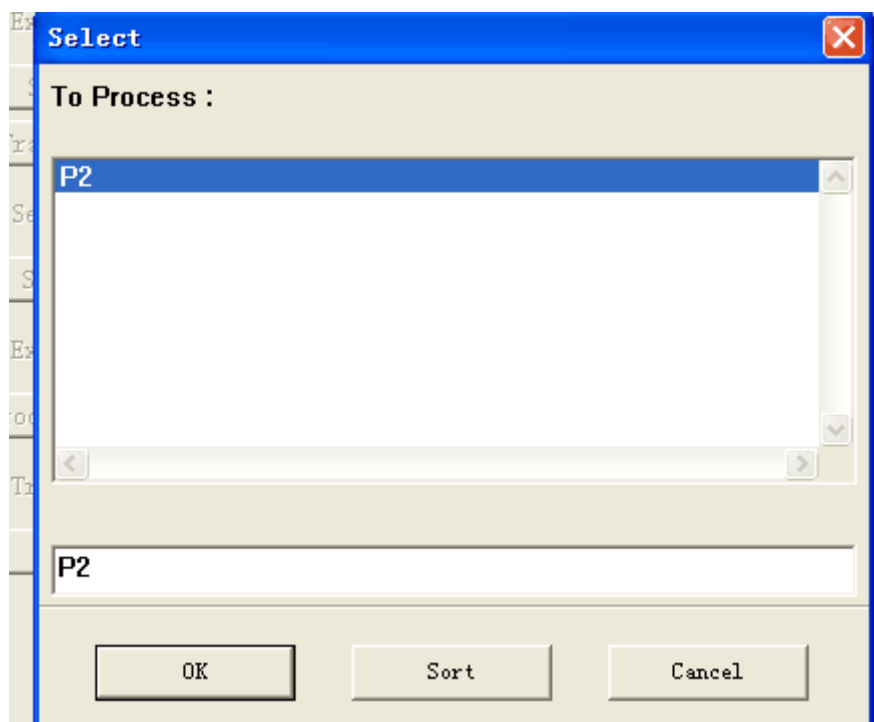


图 20. 将要发送的消息传给 P2 进程

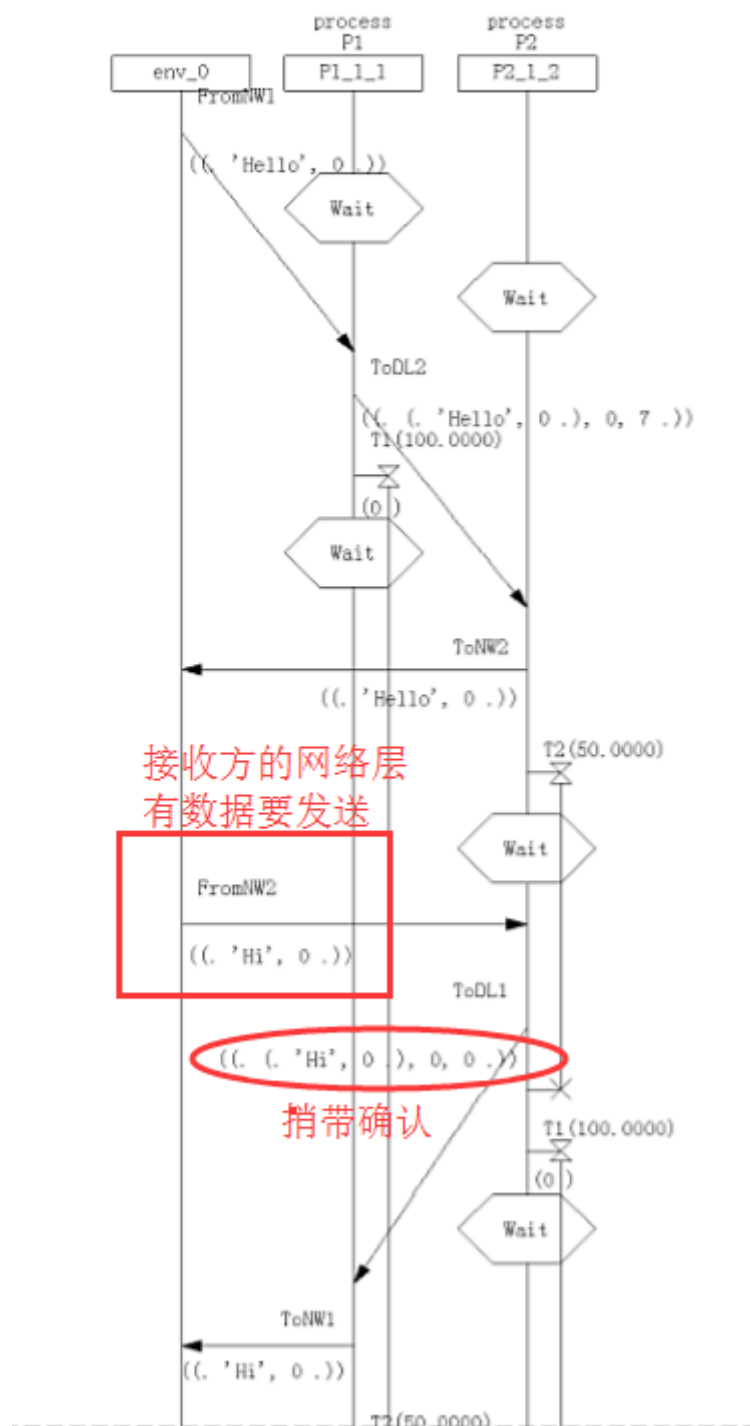


图 21. 接收方的网络层有数据要发送

4.3 发送方的帧出错

发送方发送 3 个帧 0, 1, 2 给接收方的数据链路层, 依次为有效帧, 错误帧, 有效帧。则接收方将收到的错误帧丢弃, 发送方计时器超时, 重发该帧。

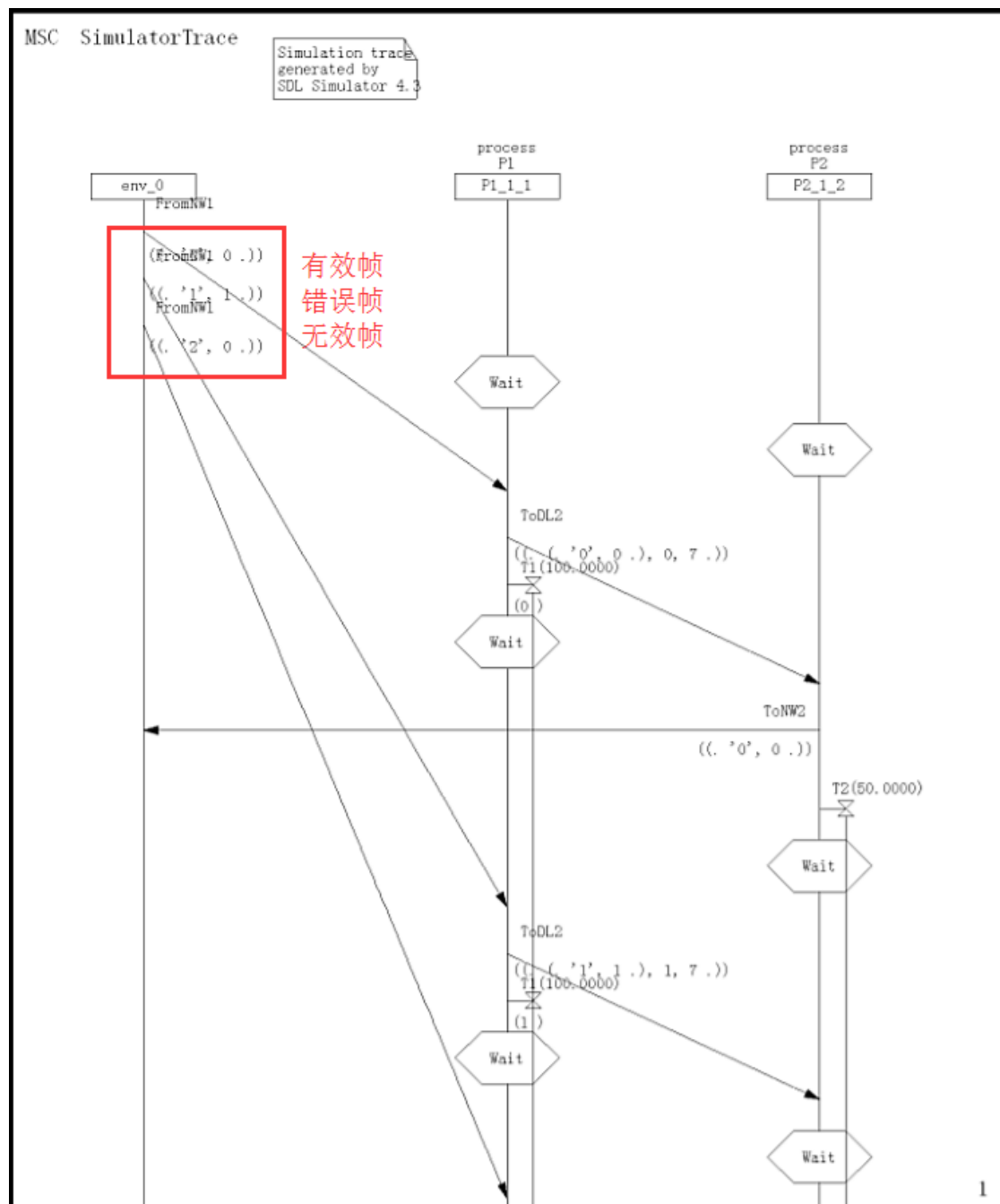


图 22.发送方发送了错误帧

