## RDT实践指导书

《计算机网络》课程组

2022年4月

# 一、实践目的

1. 深入理解可靠数据传输RDT的基本原理。

2. 掌握Stop-and-Wait和Go-Back-N两种协议的设计方法，能够使用FSM图进行设计。

3. 掌握Stop-and-Wait和Go-Back-N两种协议的实现方法。

# 二、实践内容

在仿真环境下编写传输层可靠数据传输代码，实现单向传输的Stop-and-Wait和Go-Back-N两种协议。即教材[1]中Chapter3-Programming Assignments-Implementing a Reliable Transport Protocol。

教材[1]的作者James Kurose等提供了网络仿真过程的模拟、数据报发送和接收、除传输层以外的各层功能等大量基础代码。这些基础代码已经构建起了一套完善的网络仿真环境，如图1所示。下面做简要说明，详见[2]。

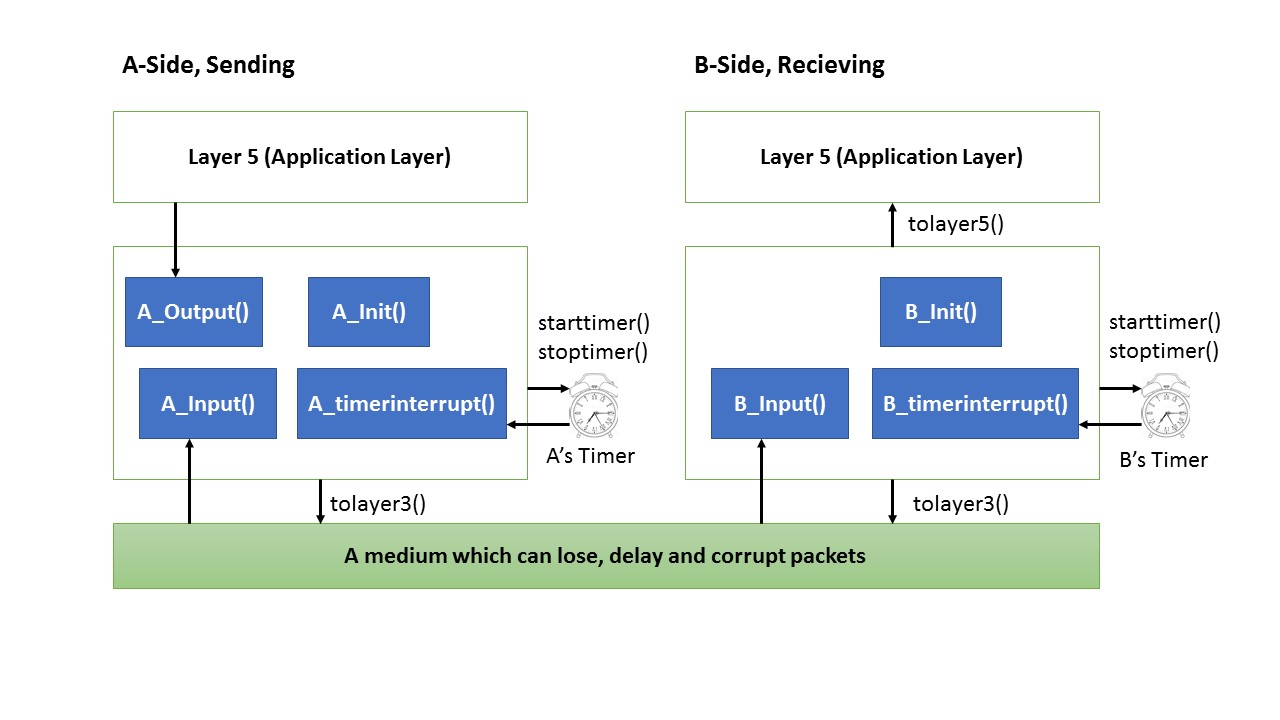


图1 RDT仿真框架图

每个网络节点的传输层代码部分需要自行填补。具体包括节点传输层的初始化操作、节点接收到应用层消息的处理过程、节点接收到网络层数据包的处理过程、节点计时器到时的响应过程等内容。

为方便实现，提供了以下实现好的过程供调用：

**starttimer()和stoptimer ()：**调用此函数启动和停止节点的计时器。

**tolayer3(calling\_entity, packet)**：调用此函数，传入需要发送的packet，将数据包传递至网络层。这之后的传输工作将由框架自动完成。

**tolayer5(calling\_entity, message)**：通过调用此函数，传入message，将消息传递至应用层。这之后仿真框架将检查数据是否完好、是否按序到达。

完成所有节点的算法后，运行网络仿真，记录仿真过程和结果，并对结果进行分析和总结。

# 三、每周工作进度具体要求及评分标准

**3.1进度安排**

第一周：设计和实现单向Stop-and-Wait协议。提交进度报告；协议源码、可执行文件；测试用例和trace文件。

第二周：设计和实现单向Go-Back-N协议。提交进度报告；协议源码、可执行文件；测试用例和trace文件。

提交最终课程设计报告和源码。

**3.2 具体要求和评分标准**

**总体分数分布：**

表1 总体分数分配比例

|  |  |  |
| --- | --- | --- |
|  | **完成内容** | **分值比例**  **（满分100分）** |
| 1 | 第1周任务 | 30% |
| 2 | 第2周任务 | 50% |
| 3 | 最终课程设计报告和源码 | 20% |
| 合计 |  | 100% |

**每周的具体要求和评分标准如表2所示。课程设计报告的要求和评分标准参照进度报告。**

表2 每周具体要求和评分标准

|  |  |  |  |
| --- | --- | --- | --- |
|  | **任务点** | **完成内容及评分标准** | **分值比例**  **（满分100分）** |
| 进度报告 | 1 | 协议需求分析。  分别阐述两种协议的需求和所适用的网络场景。 | 5% |
| 2 | 协议设计。  使用FSM工具分别设计两种协议的发送端和接收端的工作流程，并给出相应FSM图。注：只用ACK不用NAK。 | 15% |
| 3 | 协议实现。  根据FSM图在仿真框架下分别实现Stop-and-Wait和Go-Back-N两种协议。分别说明两种协议实现的具体方法、主要数据结构，给出流程图或者伪代码。 | 30% |
| 4 | 实验结果及分析。  1）能够按要求给出协议功能测试用例，并展示结果。测试要求详见下文“补充说明”部分。  2）能够对结果进行合理的分析说明。 | 20% |
| 5 | 结论。  总结本周任务完成情况，按照周进度任务要求详细填写完成进度表格。 | 5% |
| 源码 | 1 | 能够运行并实现本周协议的所有功能。结果可重复。 | 20% |
|  | 2 | 代码整洁，可读性强 | 5% |
| 合计 |  |  | 100% |

**补充说明：**

**1、报告格式要求**

周进度报告请参照“计算机网络课程设计周进度报告模版”；

课程设计报告请参照“计算机网络课程设计报告模版”。

**2、协议测试要求**

**协议功能测试：**

（1）设计测试用例，测试两种协议的基本功能。

合理选择仿真参数(发送数据包个数、数据包发送间隔、出错率、丢包率等)，模拟出以下情况：

1）无error无loss的情况下，数据包能够正确发送和接收；至少要发送20个数据包。

2）没有loss，只有error的情况。

a)数据包出现error；

b)ACK包出现error；

能够模拟出上述两种情况，正确检测出error，并做出相应处理。

3）没有error，只有loss的情况。

a)数据包出现loss；

b)ACK包出现loss；

能够模拟出上述两种情况，正确检测出loss，并做出相应处理。

4）有error，有loss的情况。

能够模拟出所有情况，并做出相应处理。

注意：仿真框架中layer5和layer4之间没有数据缓冲区，需要自行设计缓冲区并防止该缓冲区溢出，合理考虑缓冲区大小和发送数据包间隔的设置。

对于stop-and-wait协议，建议设置较大的数据包发送间隔，以确保发送端有足够的时间把当前数据包正确发送出去。

对于Go-Back-N协议，合理选择传输层的发送缓冲区的大小和数据包发送间隔，确保layer5发送下来的数据包都能存储在传输层的发送缓冲区，不会导致缓冲区的溢出。

参数的设置可参考[2]。

（2）测试结果展示和分析说明。

使用仿真框架提供的tracing功能记录每个测试用例的运行过程， tracing level=2。 在输出文件中用不同颜色标出相应事件，并对结果进行解释说明。

**3、最终RDT课程设计报告和源码说明**

鼓励同学们不断改进自己的源码和实验报告。持续提升自己的编程能力和文档写作能力。

课程设计报告在进度报告的基础上撰写。具体要求：

1）在课程设计报告中，增加一章“改进内容”。简要列出对各部分所做的改进工作,包括源码的改进。

2）具体的改进在文档正文中用“黄色”字底标出。

**评分：**

1）如果没有任何改进，则按照进度报告和源码的得分折合成这部分的分值。

2）如果有改进，根据表2中的评分标准给分。

# 四、实践方式

1. 单人独立完成。
2. 分两周完成。每周都要提交本周的工作内容，确保按计划推进实验工作。
3. 在智慧树平台上提交实践报告（周进度报告和课程设计报告）和源码；
4. 必须使用C编程语言。
5. 推荐使用Linux操作系统，也可以使用其它操作系统。需要在实践报告中详细说明自己所用的操作系统类型、版本以及编译环境等。
6. 在规定的网络仿真基础代码上实现算法。不能修改仿真框架的源代码；除了数据包外，全局变量不能由发送方和接收方共同使用，因为实际网络中，发送方和接收方通常运行在不同的物理设备上。
7. 鼓励同学们就编程实现问题积极开展讨论和交流，但是需要独立编程实现。禁止任何形式的抄袭，一经发现，抄袭与被抄袭者均以零分处理。源码和报告文档都会使用查重软件查重。

**关于诚信问题：**

1、源代码和报告提交后，将由学校查重系统或相关查重软件查重，**如有雷同，本次实践总分记为0分。**

2、本次实验的源码**3年内禁止公开**：禁止在学生中间流传，禁止放到网上公开。请同学们严格遵守此项要求。

# 参考资料

[1] Computer Networking: A Top Down Approach，8th Edition, J. Kurose & Keith Ross。

[2] RDT-programmingAssignment.pdf.