《通信网理论与技术基础》教材配套实验

M/M/1 队列

实验指导书

1 实验目标

利用 OMNeT++软件，构建 M/M/1 队列的仿真，并对顾客的平均排队长度、平均等待时间、平均系统内停留时间四个性能参数进行数值分析验证。

2 实验背景

在网络中，节点需要依次处理到达的数据包。当业务繁忙时，数据包需要排队等待，排队论可以很好地分析数据包从到达节点至离开的全过程。M/M/1 是最简单的排队模型，利用肯德尔记号写全为 $M/M/1/\infty/\infty/FIFO$ 。其到达过程为一个参数为$\lambda(>0)$的泊松过程，服务时间是参数为$\mu(>0)$的负指数分布，系统仅有一个服务员，队列长度无限制。

3 实验方法提示

3.1 理论值计算

(1) 排队强度

排队系统强度 ρ 用来判断系统的稳定性。当系统强度 $\rho<1$时，排队长度有限，系统稳定；当 $\rho\geq1$时，排队长度趋于无穷，系统不稳定。排队强度理论值是：

$$\rho=\frac{\lambda}{\mu} \qquad \text{（式 1）}$$

其中λ代表顾客的到达率，μ代表服务台的服务率。

(2) 排队长度

排队长度有三种观测方式，分别是随机取 t 时刻观测、顾客到达时观测（不包括刚到达的顾客）和顾客被服务完毕将离开时所观察到的人数（不包括正在离去的顾客）。理论上，当顾客的到达是泊松流时，三种观测方法观测的平均排队长度相同。平均排队

长度$\bar{k}$的理论值是：

$$\frac{\rho}{1-\rho} \quad \text{（式 2）}$$

(3) 等待时间$\bar{\omega}$

等待时间表示顾客从进入排队系统直至去服务台接受服务的时间。平均等待时间的理论值是：

$$\frac{\rho}{1-\rho}\cdot\frac{1}{\mu} \quad \text{（式 3）}$$

(4) 系统内停留时间$\bar{s}$

系统内停留时间表示顾客从进入系统直至离开的总时间。平均系统内停留时间的理论值是：

$$\frac{1}{\mu(1-\rho)} \quad \text{（式 4）}$$

3.2 仿真

(1) 仿真图

编写简单模块 Source、Server 和 Sink，并构建拓扑结构实现 M/M/1 仿真。Source 模块负责产生到达率为 lambda 的泊松流，统计量有一个，是随仿真时间产生的总的顾客数和在顾客到达时观察排队长度（不包括刚到达的顾客）。Server 模块负责按照服务率 mu 服务顾客，统计量有四个，分别是随机时间观察排队长度、顾客被服务完毕之后观察排队长度、顾客等待时间和当前系统中的顾客数目。Sink 模块负责模拟顾客离开，统计量有一个，是顾客系统内停留时间。

仿真图如下：

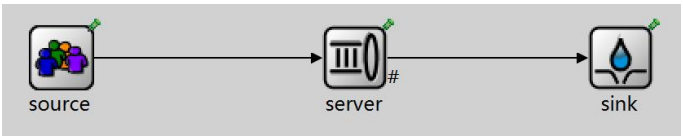

图 1 “M/M/1”排队模型的仿真图

(2) 参数设置

在表 1 中显示了仿真中各个模块需要设置的关键参数，参数“lambda”代表 Source 模块中泊松流的到达率；参数“mu”代表 Server 模块服务顾客的服务率；参数“pkCount”代表 Server 模块随机时间观察排队长度的总次数；参数“QtenvEndTime”代表仿真结束

的时间。设置“QtenvEndTime”参数，是用于程序在 0 秒至“QtenvEndTime”内，随机生成“pkCount”个时间点，以便观测排队长度。

表 1 “M/M/1”排队模型的仿真参数设置

简单模块	参数名
Source	lambda
	mu
Server	pkCount
	QtenvEndTime

(3) 代码提示

① cMessage 类

类 cMessage 用于创建消息。在排队论的仿真实验中，利用 cMessage 类来模拟数据包。类 cMessage 的构造函数拥有两个参数：对象名字和消息类别。在创建时，如果没有设置消息类别，如下面的代码所示，则默认消息类别为 0。

```
cMessage* msg1 = new cMessage("call");
```

② 基于信号的统计量

仿真信号可以用于揭示模型的统计特征，下面给出在 Server 模块中，构建基于信号的统计量来统计系统中顾客总数的思路。

- 在 Server.h 文件中构建代码

第一行代码“**int currentCustomerCount**”，定义了数据类型是整型、变量名是“currentCustomerCount”的变量。该变量的作用是在 Server.cc 文件中记录当前系统中的顾客总数。

第二行代码“**simsignal_t currentCustomerCountSignal**”，表示信号 ID 是 currentCustomerCountSignal。

```
// 当前系统中的顾客总数（包括正在服务的）
int currentCustomerCount;
// 构建信号 ID
simsignal_t currentCustomerCountSignal = registerSignal("currentCustomerCount");
```

- 在 Server.ned 文件中构建代码

第一行代码“**int currentCustomerCount =default(0)**”定义了变量

currentCustomerCount 的默认值 0。

第二行代码“**@signal[counterTotal](type="int")**”定义了一个名称是 currentCustomerCount 、类型是 int 的信号。

第三行代码“**@statistic[currentCustomerCount](title="the total number of arrived jobs"; record=vector;source=currentCustomerCount;interpolationmode=none)**”，基于信号 currentCustomerCount 声明了统计量 currentCustomerCount。表 2 具体解释了利用关键字设置的信息。

```
// 当前系统中的顾客总数（包括正在服务的）
int currentCustomerCount = default(0);
// 当前系统内顾客数（包括正在服务的）
@signal[currentCustomerCount](type="int");
@statistic[currentCustomerCount](title="current customer count";record=vector;
source=currentCustomerCount;interpolationmode=none);
```

表 2：@statistic 属性的具体设置

@statistic 属性中的关键字	作用	在本次实验中设置的具体信息	具体信息的意思
title	更长更具描述性的统计量名称；结果可视化工具可以将其用做表格标签。	current customer count	指定统计量名称是当前系统内的顾客数，并在绘图时作为标题。
record	设置记录模式。	vector	vector 可以将输入数据 currentCustomerCount 和时间戳一起记录，并将其输出到向量结果文件中。
source	指定统计量的输入信号。	currentCustomerCount	指定了统计量 currentCustomerCount 的输入信

			号是 currentCustomerCount。
interpolation-mode	定义了如何在需要的地方插入信号	none	不插入信号

- 在 Server.cc 文件中构建代码

在“void Server::initialize()”函数中构建如下代码。

第一行代码“currentCustomerCount = par("currentCustomerCount ")”，左边的 currentCustomerCount 是在 Server.h 文件中定义的，右边的 currentCustomerCount 是在 Server.ned 文件中定义的。这句代码的意思是将 NED 文件中定义的 currentCustomerCount 的默认值赋值给 Sink.h 文件中定义的 currentCustomerCount 。

第二行代码“currentCustomerCountSignal = registerSignal("currentCustomerCount")”，将信号名称 currentCustomerCount 和信号 ID currentCustomerCountSignal 相关联。在程序运行过程中，为了提高程序运行效率，一般利用信号 ID 而不是信号名称识别信号。

```
// 当前系统中的顾客总数（包括正在服务的）
currentCustomerCount = par("currentCustomerCount");
// 当前系统中的顾客总数（包括正在服务的）的信号
currentCustomerCountSignal = registerSignal("currentCustomerCount");
```

在 Server.cc 中，利用 emit()函数记录当前系统内的顾客数。emit()函数的参数分别是信号 ID 和信号值。

```
emit(currentCustomerCountSignal, currentCustomerCount);
```

4 实验步骤

1）编写 Source、Server 和 Sink 简单模块；

2）利用三个简单模块构建复合模块，设置参数，进行仿真；

3）仿真结束后，利用基于信号的统计量绘图，将仿真值与理论值进行比较。

5 提交要求

学生需要提交一份实验报告，并将实验报告、源代码、数据打包成压缩包上交。其中实验报告内容包括但不限于：

1）研究背景与问题描述

2）总体思路与方案论证

3）结果分析与结论

4）课程学习收获与体会建议