

# 对等网络的仿真研究

张少娴

(南京邮电大学 计算机学院, 江苏 南京 210046)

**摘要:** P2P网络和P2P相关技术一直是计算机网络研究的一大热点。P2P网络具有大规模、高动态性的特点,这使得在真实环境中实际运行、测试一个它需要付出巨大的努力和昂贵的开销。而P2P模拟器具有的优势能为P2P网络研究带来很大的方便,研究者们只需要将精力集中于想要关注的性能,而不必花费大量的时间和精力去编写实验工具。文中深入研究了PeerSim模拟平台上模拟P2P环境的构建方法,并在其上实现了一个简单的负载均衡算法。最后,通过实验来说明PeerSim模拟器的使用方法,展示了PeerSim组件化的构造思想和灵活的配置机制。

**关键词:** P2P(Peer-to-Peer); 模拟; P2P模拟器; PeerSim

**中图分类号:** TP393.07

**文献标识码:** A

**文章编号:** 1673-629X(2012)11-0141-04

## Simulation Research of P2P Network

ZHANG Shao-xian

(School of Computer, Nanjing University of Posts and Telecommunication, Nanjing 210046, China)

**Abstract:** P2P (Peer-to-Peer) network and P2P technology have always been a hotspot in computer network research. The characteristics of large-scale and high dynamicity of P2P networks make that running and testing a P2P network in an actual network environment requires huge efforts and expense cost. However, the advantages of P2P simulators provide much convenience for P2P network research. Researchers can focus only on those properties they want to know, rather than spending a lot of time and energy in developing experiment tools. It carries out a deep research of the construction methods of P2P network environment on the PeerSim framework and successfully implements a simple load-balancing algorithm on PeerSim. Finally, the experiments are given out to demonstrate the using methods, the component-based structure and the flexible configuration mechanism of PeerSim simulator.

**Key words:** P2P(Peer-to-Peer); simulation; P2P simulator; PeerSim

## 0 引言

如今, P2P网络技术的应用已经渗透到各个领域, 如文件共享、多媒体传输、实时通信、协同工作、分布式数据存取、分布式计算、搜索引擎等。

在计算机网络和分布式系统的研究中, 研究者们往往倾向于实际的网络实验, 它真实、可靠、不含过多的理论假设。然而, 这并不是P2P网络实验的正确选择, 因为具有大规模、高动态性的实际P2P网络构建起来相当困难, 并且需要昂贵的开销<sup>[1]</sup>。因此, 模拟(Simulation)是P2P网络研究现实可行的测试、评价、比较和认证机制。P2P模拟器的开发为P2P领域的研究者带来了实验上的便利, 人们只需要将精力集中于那些想要关注的性能, 而不必花费大量的时间去编写

实验工具。比较有代表性的P2P模拟器有: p2psim, PlanetSim, Neurogrid, Overlay Weave, PeerSim, 3LS, GnutellaSim, NDP2PSim<sup>[2-8]</sup>等。

文中将以PeerSim开发者提供的PeerSim文档和BISON项目的相关文档研究PeerSim模拟器的模拟过程, 分析了PeerSim模拟的生命周期, 并详细介绍了PeerSim模拟环境的实现以及文中要模拟实现的负载均衡算法, 最后通过实验介绍了PeerSim模拟器的使用方法和性能, 并对仿真实验的结果进行分析。

## 1 PeerSim 模拟器

PeerSim模拟器用Java语言编写, 它现在作为一个开源项目发布在SourceForge<sup>[9]</sup>上, 供P2P研究者们使用和改进。PeerSim它由两个模拟引擎<sup>[10]</sup>组成: 一个是循环驱动(Cycle Driven)的, 另一个是事件驱动(Event Driven)的。这两个引擎由许多简单的、可扩展的插件组成, 并提供了一个非常灵活的配置机制, 能够对不同的拓扑结构和动态环境等进行模拟。文中选用的是循环驱动的模拟引擎。

收稿日期: 2012-02-27; 修回日期: 2012-05-29

基金项目: 国家自然科学基金(61003040); 中国博士后科学基金(20100471353)

作者简介: 张少娴(1972-), 女, 广西南宁人, 讲师, 硕士, 研究方向为数据库、信息安全。

基于循环的模拟过程的生命周期<sup>[11,12]</sup>(见图 1)如下:

第一步,读取配置文件。配置文件中包含了模拟器的属性和模拟过程中涉及的所有组件对象的所有参数。

第二步,模拟器启动网络创建过程,初始化网络结点和结点内的协议。所有结点都拥有相同的协议。协议的所有实例在网络中构成了一个队列,而每个结点都有协议的一个实例。

第三步,网络创建完之后,开始初始化过程。初始化过程将设立各个协议的初始状态,架构起网络拓扑结构,并为结点建立邻居结点类表。初始化过程是由控制(Control)组件对象执行的。

第四步,初始化阶段之后,开始模拟过程。循环驱动的模拟引擎在每一轮循环中调用一次所有的组件(协议和控制组件),直到达到设定的循环数,或者有一个组件决定终止模拟过程。

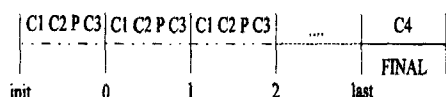


图 1 基于循环的 PeerSim 模拟生命周期

其中,C 指控制(Controls)组件,P 指协议(Protocols)。底下的数字指模拟过程的循环数。在最后一个循环中,可以运行一个 Control 对象以获得模拟系统的快照。

## 2 PeerSim 模拟环境实现

用 PeerSim 模拟器实现协议(或算法)模拟要做两个工作:一是依据 PeerSim 框架编写协议(或算法);二是编写对应的配置文件。

协议的编写中,仅编写出协议自身是不够的,还需要根据 PeerSim 框架编写其他相应的组件,这些组件主要有:

### (1) 初始化组件(Initializer Component):

初始化组件完成的工作主要有两个:一是初始化网络结点和网络拓扑结构。二是设置网络结点的访问顺序和结点的数值分布状态等。

### (2) 控制组件(Control Component):

控制组件可以分为两类,动态组件和观察员组件。动态组件完成的工作是在每个结点上执行协议规定的操作,模拟结点动态行为。观察员组件则为监视模拟过程中的结点状态变化或系统属性值改变提供了机制。

配置文件是一个 ASCII 码的纯文本文件,由一系列的命令行组成。每条命令行都以键-值对(key-val-

ue)的形式出现。启动模拟过程时,需指定模拟过程要用到的配置文件,命令格式如下:

```
java - cp <class-path> peersim. Simulator
config-file. txt
```

其中,config-file. txt 为配置文件名。使用该命令时只要将用到的实际的配置文件名替代 config-file. txt。类搜索路径 class-path 不是必须的,只有在没有将 peersim 的类目录加入到系统的 class-path 的情况下才需要指明 class-path。否则,模拟器会自动搜索路径以找到相对应的类。

配置文件内容可以按功能分成几部分:模拟属性参数,协议组件,初始化组件,控制组件。

模拟属性参数(Simulation Properties)主要指随机种子、模拟循环次数、网络的大小、结点访问次序等。协议组件实例和参数的定义以“protocol”开头,相对地,初始化组件和控制组件的实例和参数定义分别以“init”和“control”开头。协议组件(Protocols Component)包括模拟过程要模拟的协议和用来辅助该协议的其他协议,如拓扑管理协议。初始化组件(Initializers Component)完成拓扑结构的初始化和结点数值的分配。初始化拓扑结构是为网络结点建立连接,使它们构成一个拓扑结构。指定了初始化组件之后,必须显示地声明协议用到的拓扑管理协议。控制组件(Controls Component)负责对被模拟系统的状态和属性进行监视,并将数据输出到控制台或一个指定的文件。

组件实例的定义语句的格式为:

```
<protocol | init | control>. string_id [full_path_]
classname
```

组件参数的定义语句的格式为:

```
<protocol | init | control>. string_id. parameter_name
parameter_value
```

## 3 Load Balance 算法设计

这里将在 PeerSim 模拟一个简单的负载均衡算法。该算法的任务是经过一系列的数值交换,最终使网络上的结点所拥有的负载达到平衡。该策略是负载差值策略<sup>[13,14]</sup>,为网络上的每个结点维持两个数值:负载(load)和配额(quota)。负载指该结点拥有的可传输数量,配额指该结点在每一次循环中所能传输的最大数量,如果配额为 0,则结点不能进行数值传输。网络上的每一个结点会在网络中寻找离自己最“远”的邻居结点,并与它进行数值交换,交换的数值不能超过两个结点配额值中较小的那个。邻居结点的“远”和“近”是通过比较两个结点的负载差值。如果该结点在网络上找到了一个结点,该结点的负载和自己的负载差额最大,那么这个结点就是最“远”邻居结点。

在每一次循环末尾,结点的配额都会被重新设置为默认值,这样保证了在一定的循环次数下,所有的结点的负载最终能够达到一个平衡水平。

为了实现对该算法的模拟,设计了如下类:

BasicBalance 类实现了负载差值传输策略。BasicBalance 类的主要函数 nextCycle(Node node, int protocolD) 实现了 CDProtocol 接口的方法,协议的主体功能由该方法实现。该方法利用底层 Linkable 协议(如 SimpleNewscast 和 IdleProtocol)提供的机制选择最“远”的邻居,完成结点负载交换。

BasicBalance 协议的作用是初始化网络结点。为了建立一个覆盖网拓扑结构,还需要为结点建立连接,并维护一个邻居结点列表。为此必须有一个扩展了 WireGraph 类的负责建立结点间连接的协议和一个实现了 Linkable 接口的拓扑管理协议。

PeerSim 的 dynamics 包里提供了许多建立结点连接的协议,如 wireByMethod, WireFromFile, WireKOut, WireStar, WireWS 等。这些协议都扩展了 WireGraph 类。运行过程中,会选取一个 Linkable 协议,以指定的方式加入结点连接构成一个特定的拓扑结构。文中将在算法中选用 WireKOut 协议。该协议将随机选择结点,并建立它们的连接,即加入的连接是随机的。

拓扑管理协议维护网络的拓扑结构,并为结点提供访问邻居结点的机制。本算法将采用 Newscast 这一流行的拓扑管理协议,为其他协议提供搜索邻居结点的机制。PeerSim 系统提供了一个简单的 Newscast 协议:example. newscast. SimpleNewscast。除 SimpleNewscast 外,PeerSim 还提供了其他拓扑管理协议,如 IdleProtocol,它保存结点之间的连接关系,并为结点提供方法访问邻居结点。因此在配置文件中,将指定该协议作为网络的拓扑初始化工具。

由于本算法的操作是在结点保存的数值上进行的,因此在初始化阶段应考虑结点数值的分布方式,是线性分布,还是峰值分布,或者是其他分布方式。

类 LinearDistributionInitializer 实现了 Control 接口,实现了对结点的数值的线性分配。该类根据设定的数值上界和下界,以线性分布的方式为每一个结点分配数值。该类从配置文件中读取指定的数值上界和下界,然后以线性增加的方式为每个结点分配数值。

此外,还可以利用 PeerSim 提供的其它类来实现不同的数据分布,如实现峰值分布的类 PeakDistributionInitializer(example. aggregation. PeakDistributionInitializer)。

本算法涉及结点的两个数值,因此提供两个观察员类,监视结点负载和配额的变化情况,从而得知每一次循环后网络达到的平衡水平。QuotaObserver 类负责

监视每个结点的配额值。LBObserver 负责监视每一次循环的负载平衡过程的状态。此外,LBObserver 类有一个 show\_values 参数,如果该参数在配置文件中被定义,那么 LBObserver 将打印出所有结点的负载。

根据上面的设计,可以编写出配置文件 config-loadbalance.txt(位于 peersim-1.0.3/example 文件夹下)。在验证实验中可以修改其中的部分配置参数,对比不同的参数设置带来不同的模拟效果。

## 4 仿真试验及结果分析

本实验将模拟 BasicBalance 算法,配置文件为 loadbalance\_example1.txt(位于 peersim-1.0.3/example 文件夹下)。参数设置如下:模拟循环次数 cycles 设置为 30,网络大小 size 设置为 20。结点的传输配额均为 100。数值分布选择峰值分布,最大值为 1000;连接建立协议选择 WireKOut 协议,参数 k 设置为 20,表示每个结点将建立 20 个连接。拓扑管理协议选择 SimpleNewscast,参数 cache 设置为 20,表示该协议为每个结点保存 20 个邻居结点。

实验步骤如下:

(1) 进入 MS-DOS 模式。

(2) 用 cd 命令进入 peersim-1.0.3 所在文件夹。

(3) 键入以下命令:

```
java - cp " peersim-1.0.3. jar; jep-2.3.0. jar;  
djep-1.0.0. jar" peersim. Simulator example/loadbalance  
_example1. txt
```

(4) 启动 PeerSim。

在键入命令之后,PeerSim 模拟器开始运行。它首先加载配置文件,接着 PeerSim 判定所选用的模拟引擎是“CDSimulator”,即循环驱动的模拟引擎。然后,PeerSim 调用 CDSimulator 开始循环模拟:

循环开始后,可以从 cycle0 的第一个 control.lbo 输出中看到结点负载经过峰值分布的结果,20 个结点中只有一个结点分到负载值 1000,其余结点的初始负载值都为 0。那么所有结点的负载总和为 1000,达到负载平衡后每个结点的负载值应为  $50 = 1000/20$ 。

对比 cycle0 和 cycle1 的输出可以发现,已经有结点交换了负载值。负载值为 1000 的结点传输给另一个结点 100 个数值,得到数值 900 和 100 的两个结点。cycle2 中有许多结点进行了负载交换。对比 cycle1 和 cycle2 的输出,可知数值变换呈现很大的随机性。

观察 control.qo 的输出可以发现,每个循环的输出都是一样的。这是因为控制组件 ResetQuota 在每次循环的末尾会重置每个结点的配额,设置为默认的 100。所以 control.qo 对配额的统计结果不会发生变化。

