

并行离散事件网络仿真系统的研究与实现

李 越¹, 钱德沛², 张兴军², 邢春晓¹, 何 莹²

(1. 清华大学信息技术研究院, 北京 100084; 2. 北京航空航天大学计算机学院, 北京 100083)

摘 要: 互联网的不可控、易变和不可预测的特性给网络协议的开发、调试和性能测试带来极大的困难。网络仿真系统构造可控、可重现网络状况的虚拟网络环境, 可以在新协议新算法部署之前预测其可用性和性能。文中所述的并行离散事件网络仿真系统具有以下特点: (1) 基于并行计算模型 BSP 的实时事件调度引擎-BREEN; (2) 网络应用透明的灵活的交互机制; (3) 重用基于流行仿真工具开发的网络模型。最后以网络路径瓶颈带宽测量和分布协作 web 缓冲系统作为应用实例, 分别进行可用性评估和性能预测。

关键词: 网络仿真; 并行离散事件系统; 实时调度; 交互机制

中图分类号: TP393.01 **文献标识码:** A **文章编号:** 0372-2112 (2006) 02-0365-06

Research and Implementation of a Parallel Discrete Event Network Emulator

LI Y ue¹, QIAN De-pai², ZHANG Xing-jun², XING Chun-xiao¹, HE Ying²

(1. Research Institute of Information Technology, Tsinghua University, Beijing 100084 China;

2. School of Computer, Beihang University, Beijing 100083 China)

Abstract Internet is characterized as a uncontrollable, variable and unpredictable infrastructure, this brings the difficulty of developing, debugging and evaluating network applications. Network emulation system constructs a virtual network environment which has the characteristics of controllable and repeatable network conditions. This makes it possible to predict the correctness and performance of proposed new technology before deploying to Internet. In this paper we present a methodology for predicting the correctness and performance of applications based on the PARNEM, a parallel discrete event network emulator. PARNEM employs a BSP based real time event scheduling engine, provides flexible interactive mechanism and facilitates legacy network models reuse. PARNEM allows detailed and accurate study of application behavior. Comprehensive case studies covering bottleneck bandwidth measurement and distributed cooperative web caching system demonstrate that network emulation technology opens a wide range of new opportunities for examining the behavior of applications.

Key words network emulation; parallel discrete event system; real time scheduling; interactive mode

1 引言

网络技术和网络应用的快速发展, 使得研究人员不断开发新的网络协议和算法, 来适应新的需求。在这个过程中, 需要测试协议的可用性和评价协议的性能。然而互联网的不可控、易变和不可预测等特性给协议的开发、调试和性能测试带来极大的困难。网络仿真系统构造可控、可重现网络状况的虚拟网络环境, 模拟真实网络的各种行为和特性, 为新协议新算法的研究和开发提供方便灵活的手段。它具有以下重要性:

- 预测新协议新算法在不同网络条件下的可用性和性能;

- 比较不同的协议设计和参数对性能的影响;
- 重现触发协议错误的某些特定网络状况以方便调试。

当前国际上比较成熟的网络仿真系统主要有 Nist Net^[1], DelayLine^[2], Hibox^[3], Trace-based network emulation^[4], NS emulator^[5], IP-TNE^[6]。这其中只有加拿大 Calgary 大学和英国 Bath 大学合作开发的 IP-TNE 采用了并行仿真技术, 其内核是一个基于通道 (channel) 的保守同步任务调度器。文献 [6] 给出了相关的论述。

本课题隶属国家 973 项目“海量信息系统概念性试验和验证平台”, 旨在研究并行网络仿真系统结构及关键技术, 实现原型系统。特点有: (1) 设计实现了基于并行计算

收稿日期: 2004-08-01; 修回日期: 2005-09-15

基金项目: 高校博士点基金 (No. 20020006011); 国家自然科学基金 (No. 60473078); 国家自然科学基金 (No. 90412011)

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

模型 BSP 的实时事件调度引擎-BREEN.它是网络仿真系统运行的核心部件,底层平台是广泛使用的工作站机群.并行网络仿真系统用多台宿主机分担外部网络应用产生的负载,提高系统运行的效率,具有更好的实时性.(2)网络仿真系统不是一个孤立的封闭系统,必须提供将外部数据流导入导出的机制.在本系统中我们设计了网络应用透明的灵活的交互机制,支持不同规模的网络应用的开发和调试.(3)网络仿真系统需要提供丰富的模型库,比如协议模型、流量模型和拓扑模型.在本系统中研究人员可以使用网络研究领域流行的仿真工具-NS提供的模型库构建需要的网络模型.我们只对 NS 模型库做了有限的扩展,以支持仿真的并行运行.这样设计的好处是使大量已经开发的基于 NS 的网络模型得以重用.最后以网络路径瓶颈带宽测量和分布协作 web 缓冲系统作为应用实例,分别进行可用性评估和性能预测.

2 并行网络仿真系统结构

模型、模拟、仿真对实际系统的依赖关系依次一个比一个更强,产生的效果也依次一个比一个更逼真.模型所需建立时间最少,但描述系统的精确程度很低,一般用于快速分析.模拟完全利用软件方式,可以输入各种激励,反复实验,描述系统的精度可以很高,一般不能支持实时.仿真通常结合实际系统,利用硬件加速器,实时性强、逼真.网络仿真是提供一种将网络模型和真实网络协议相结合,来评价网络协议性能的方法,这与传统意义上的网络模拟有很大不同.

第一,网络模拟需要首先对网络的各个层次建模,然后进行模型的有效性验证,最后运行模拟得出结论.模型的建立和验证往往非常困难,而且一旦需要对协议的设计做修改,建模和验证的过程就必须重来.网络仿真对网络的某些层次(网络层以下)建立模型(实际上可以采用已经成熟的通用的模型),其他层次(传输层,应用层)由开发人员根据其特定的需求自行设计实现.这相当于构建了一个混合机制,即网络的一部分由模型组成,另一部分是协议的真实实现,网络仿真系统提供二者交互的机制.

第二,实时性是网络仿真的核心特性,也是衡量仿真精度的重要性能指标.仿真系统时钟的推进必须和真实时间同步,这是仿真系统和真实世界交互的前提.而模拟系统采取“下一事件”的事件调度和处理方式,时钟的推进与真实时间没有关系,是虚拟时间.当外部负载过大时,仿真系统时钟往往会落后真实时间,实时性不能保证,影响仿真的精度.为此我们采用并行仿真技术,用多台主机分担外部负载,并行实时事件调度引擎-BREEN 负责时间同步,具体的细节见第 3 节.

NS^[5]是当前网络研究领域最流行的网络模拟器,其特点是提供了丰富的网络模型库,大量已经开发的基于 NS 的网络模型正在被研究人员广泛使用.这些已有的模

型可以在我们的系统中被方便地重用.网络模型被划分为子模型,分配给各仿真进程,如图 1 所示.最优的划分策略应该兼顾时间前瞻量 (lookahead)、负载均衡和通讯开销.我们对 NS 模型库的各个层次做了并行化扩展以支持子模型之间物理和逻辑上的连接.详细的论述参见文献 [7].

我们提供网络应用透明的灵活的交互机制,网络应用产生的数据流被透明地导入到仿真系统中,这样外部网络应用就好像面对一个真实的网络.可以通过主机模式 (Host mode) 或者进程模式 (Process mode)

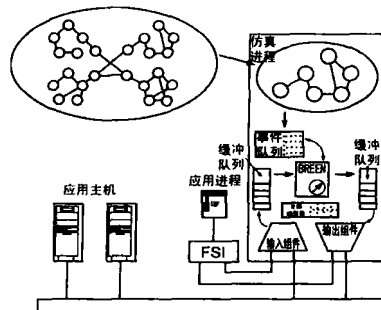


图 1 并行离散事件网络仿真系统结构

与仿真系统交互.主机模式中网络应用和仿真进程运行在不同主机上,支持大规模的仿真实验,比如分布式协同工作的网络应用.小规模的仿真实验可以采用进程模式,网络应用作为一个网络进程运行在仿真主机上,FSI (Fake Socket Interface) 机制负责应用流量的导入导出.输入输出部件 (I/O Component) 负责真实世界和仿真系统网络包格式转换,这个过程中需要查询映射表获取相关的信息.具体的细节见第 4 节.

3 基于 BSP 的实时事件调度引擎-BREEN

并行计算模型 BSP^[8]由 Valient 在 1990 年提出,目的是为各种并行体系结构提供一种通用的、具有可扩展并行性的理论模型.它不仅是一种并行体系结构模型,也是一种并行程序设计模型. BSP 计算过程由一系列被全局同步分隔的超步 (superstep) 组成. BSP 模型具有通用性、程序结构清晰、开发调试方便等优良的特性,已经在科学计算、并行数据库系统等领域获得了广泛的应用.

基于 BSP 模型设计 BREEN 的执行过程为一个包含以下两步处理的循环,可以看出这是一个保守同步算法,用全局同步操作来决定可以“安全”执行的事件.

- 全局同步,计算时间下限 T_b ,即未来可能会收到消息的最小时戳;
- 处理所有时戳不超过 T_b 的事件.

为了计算 T_b ,时间前瞻量的概念非常重要,其含义是从某个子模型的事件被处理,到它对另一个子模型产生影响的一段时间.

3.1 实时的概念

首先所有的仿真进程必须有统一的实时的概念.可以采取的做法是把经过初始对表的本地系统时钟作为实时的一个参考,忽略主机系统时钟的漂移.如果仿真时钟和本机的系统时钟同步,就认为它与实时保持同步.其次仿真系统的时钟不能采取模拟系统的“下一事件”调度方式

来推进,即不能立即处理下一个事件,而是应该在适当的真实时间处理。

3.2 实现

图 2 给出了 BREEN 执行的伪代码。如前所述其执行过程分为两步,第一步计算时间下限 T_b , 然后处理满足条件的事件。其中 TS_i 和 LA_i 分别是第 i 个子模型的下一个待处理事件的时戳和时间前瞻量, TS 是当前子模型的下一个待处理事件的时戳。为了计算将来可能会收到消息的时戳下限 T_b , 需要考虑过渡消息 (transient message) 的影响。过渡消息是指已经发出但尚未到达目的地的事件消息。文献 [7] 介绍了全局同步和捕获过渡消息的两轮缩减算法。在计算 T_b 的时候同样需要考虑实时的影响, 在任一时刻对实时的采样值代表了外来数据包产生事件的时戳下限。一旦时间下限 T_b 计算完成, 那么所有时戳不超过 T_b 的事件就可以被“安全地”处理。如果下一个待处理事件的时戳大于实时的采样值 ($TS > RT$), 那么调度器会等待真实时间到达 TS 时处理该事件, 或者一个新的外部数据包到来。到来的数据包会产生一个时戳为当前时刻的事件插入事件队列。

3.3 T_b 的计算开销

为了计算 T_b , 运行在每台宿主机上的仿真进程需要参与全局的同步和计算。我们的测试环境是四台浪潮 NP320 工作站, 每一台的配置是双处理器 Pentium III 1.13GHz 内存 1.5GB, 用 100M bps Ethernet 互连。每台工作站上运行一个仿真进程。采用负载均衡的模型划分策略, 每一个本地逻辑连接包含 3 跳, 每一个远程逻辑连接包含 3 跳在源子模型, 3 跳在目的子模型, 划分策略保证每个子模型上具有相同的跳数。实验在相对空闲的局域网环境开展, 测试结果表明计算时间下限 T_b 可以在 3 毫秒以内完成。

这是因为固定长度的消息在局域网的传输具有延迟上限。BREEN 中采用的缩减算法保证 T_b 的计算经过两轮处理可以完成, 而且不论有多少过渡消息, 对于 n 个仿真进程, 最多有 $2n \log n$ 个同步消息^[7]。所以 T_b 的计算具有开销上限。

BREEN Execution Loop

```

1  while( unprocessed events remain ) {
2      // first compute  $T_b$ 
3      receive messages from other process in previous cycle;
4      catch transient messages;
5       $T_b = \min(\min(TS_i, Sample\_Realtime()) + LA_i)$ ;
6      // process eligible events
7      while( $TS \leq T_b$ ) {
8          do {
9               $RT = Sample\_Realtime()$ ;
10             if( a new packet comes in )
11                 break;
12             } while( $TS > RT$ );
13             dequeue the next event and process it
14         }
15     }
```

图 2 BREEN 执行过程的伪代码

4 应用透明的灵活的交互机制

网络仿真系统的交互机制提供外部负载导入导出的方式。这种交互首先要对网络应用透明。其次不同特性的网络应用可以选择适合的交互模式, 主机模式或进程模式。输入输出组件负责真实世界和仿真系统网络包格式的转变。

4.1 主机模式 (Hostmode)

主机模式中网络应用和仿真进程运行在不同主机上, 支持大规模的仿真实验。在第 6 节以分布协作 web 缓冲系统作为应用实例的实验中采用了主机模式。问题是在局域网环境下如何使应用流量经过宿主机而不直接到达目的主机。可以采用的解决方法是为每一台应用主机指定一个虚拟 ip 地址, 通过配置其路由表使应用流量 (目的地址是虚拟 ip 地址) 经过宿主机。这样可以保证只有参与仿真实验的应用流量经过宿主机, 不会影响应用主机之间的其他流量。用移植性较好的 PCAP 库^[9] 读入数据包, 它提供了一套简单的对内核过滤器编程的接口。过滤器设置为将所有目的地址位于指定的虚拟地址空间的数据包读入仿真器。这样只需要生成一个过滤器, 具有很好的可扩展性, 同时也减轻了用户配置网络建模脚本的负担^[10]。数据包的写出采用原始套接口, 因为 IP 包头的相关字段已经在输出组件中做了适当的修改, 不需要操作系统的协议栈对 TCP 包头和 IP 包头有任何的操作。采用原始套接口的另一个原因是它已经被多数的操作系统支持, 具有较好的移植性。

4.2 进程模式 (Process mode)

在进程模式中, 网络应用作为一个进程运行在宿主机上, 适合小规模仿真实验。图 3 描述了应用进程和仿真进程的交互机制

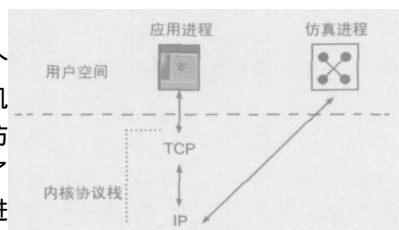


图 3 FS 通信机制

(FSI), 可以看出 FSI

将系统模块与用户空间进程的通讯机制和套接口通讯相结合, 是一种进程间的混合通讯机制。实现上是在内核协议栈的 ip 层加载系统模块, 其主要功能是在内核空间进行包过滤。系统模块与仿真进程通过 netlink^[11] 通讯。

4.3 输入输出组件

与输入输出组件的处理过程关系密切的数据结构是全局映射表。它主要维护两个映射关系: 一是应用进程和仿真器内部网络节点的对应关系, 每一个应用进程都和一个网络节点绑定; 二是真实 IP 地址和虚拟 IP 地址的对应关系。为实现映射表的快速查找, 可以采用某种哈希算法, 同时构造缓冲器以实现热点记录的快速命中, 缓冲器的维护选择 LFU 替换算法。

输入组件收到应用数据包后,利用包头信息查询全局映射表获得分别与源应用和目的应用绑定的源节点和目的节点。查询获得的信息被包含在与网络仿真器兼容的仿真数据包中,该数据包同时包含一个指向真实数据包的指针。一个与仿真数据包对应的事件被插入事件队列,事件的时戳就是当前实时时刻。一旦该事件触发,就表明与事件对应的数据包从源节点发出。仿真数据包替代真实数据包在仿真网络里经过一系列路由到达目的节点。由于网络模型只对网络层以下建模,如图4所示,所以不会影响真实数据包传输层包头的字段,对于IP头的某些字段则会产生影响(比如TTL),所以这些字段必须包含在仿真数据包中。

输出组件和输入组件分别实现为独立的线程。已经到达目的节点的数据包被放进输出队列。

输出组件从队列中读包,将其目的IP地址改为真实IP地址,源IP地址改为虚拟IP地址,重新计算检查和。将数据包发出。

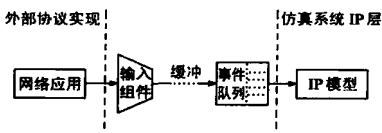


图4 外部网络应用流量进入仿真器

5 仿真器的验证

模型的正确性和仿真器的实时能力是影响仿真精度的关键因素。本节我们将对仿真器的延迟模型、带宽模型以及丢包模型的正确性进行验证,并通过强度测试实验说明串行仿真和并行仿真实时能力的差异。

5.1 延迟模型验证

包延迟是网络的基本行为。在仿真器中,用户指定链路的带宽和传播延迟,从而确定路径的端到端延迟(在本实验中,可以适当地设置网络模型参数以排除排队延迟的影响)。仿真数据包里包含一个时间戳,该时戳反映了数据包经历仿真网络时的延迟。当数据包到达输出组件,其时戳是未来的某个时刻,数据包在该时刻被仿真器发出。这样数据包在仿真网络里经历的延迟完全由网络模型控制,保证了延迟模型的正确性。如果当数据包到达输出组件时,其时戳小于当前的实时时刻,即过去的某个时刻,那么说明仿真器引入了网络模型以外的附加的延迟,仿真精度受到影响。定义滞后比例为所有滞后的包个数(到达输出组件时的时戳小于当前的实时时刻)占总包数的百分比。滞后比例可以作为衡量仿真系统实时性的指标。

开展强度测试实验比较串行仿真和并行仿真实时能力的差异。并行仿真的运行环境是四台浪潮NP320工作站,用100M bps Ethernet互连。网络模型包含136个节点,包调度策略FIFO,路由协议Distance Vector生成恒定的背景流量。外部应用流量包括4组UDP server/client,包大小为100字节,包速从100pps(packet per second)增加到1600pps,每次实验运行30分钟,输出组件统计滞后比例。

图5显示了串行和并行仿真的实验结果,可以看出并行仿真的实时性在很大程度上优于串行仿真。

5.2 带宽模型验证

带宽模型根据当前数据包的大小和带宽值计算发送延迟,在该延迟时间内带宽模型处于锁定状态,随后到来的数据包等待直至解锁才会被处理。

验证带宽模型的正确性相对简单,可以通过加载UDP冲击流量并做相应的统计就能够判断其正确性。

5.3 丢包模型验证

丢包模型分为拥塞相关丢包和拥塞无关丢包两类。拥塞相关的丢包模型包括各种队列管理和包调度策略,比如FIFO, RED, CBQ等等。拥塞无关的丢包模型用于模拟网络出错引起的丢包行为。这两种类型的丢包模型都可以通过逐包跟踪的方式来验证其正确性。

6 应用实例

以网络路径瓶颈带宽测量和分布协作web缓冲系统作为应用实例,仿真实验说明网络仿真系统作为新协议新算法的验证测试平台,在部署到Internet之前预测其可用性和性能,为研究人员开发正确、高效、健壮的网络协议和算法提供支持。设计仿真实验时需要考虑网络应用的不同特性,比如规模、负载压力以及对仿真系统实时能力的要求。

6.1 网络路径瓶颈带宽测量

路径瓶颈带宽是重要的网络性能参数。带宽测量在网络管理、网络规划设计和提高网络应用程序性能等方面具有重要意义。现有的带宽测量工具大多基于包对算法(packet pair),但是测量结果的精度如何,是否在可接受的误差范围内,往往很难判断,所以需要验证带宽测量工具和测量算法的可用性。网络路径瓶颈带宽测量工具与通常的网络应用的工作方式一样,位于网络路径两端的应用程序用TCP或UDP流量通信,进行相应的计算。好的测量工具通常只给网络带来很小的附加流量,对网络行为的影响也比较小。由此可见网络路径瓶颈带宽测量工具作为仿真实验的应用实例具有以下特点:小规模,生成负载压力小,对仿真系统的实时能力要求比较低。基于以上的考虑,我们采用进程模式设计仿真实验。

实验选择带宽测量工具IGMP^[12],实验目的是验证该测量工具在不同网络条件下的可用性。在仿真系统中建立的网络模型如图6所示,三条链路的容量带宽分别是

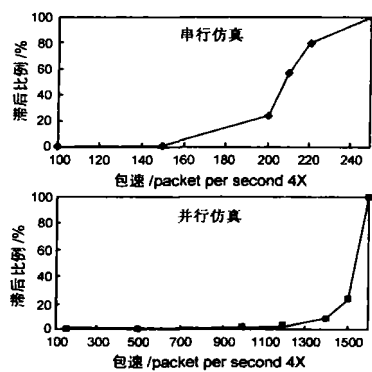


图5 串行仿真和并行仿真的实时性比较

20M bps 10M bps 20M bps A s和 A d分别是测量工具 IG I/ PTR的源端和目的端. 生成背景流量 T1, T2, T3模拟网络的负荷, 这样背景流量的变化将会改变路径 < H1, H4> 上的瓶颈带宽. 为了计算方便, T1, T2, T3 都选择 CBR (Constant B it R ate) 流量.

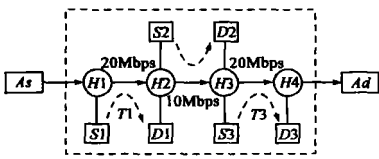
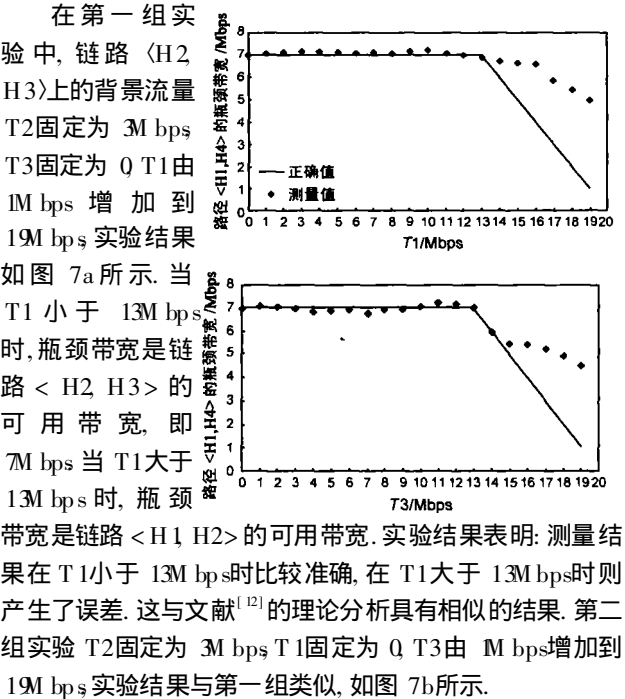


图 6 网络路径瓶颈带宽测量实验的网络模型



6.2 分布协作 web 缓冲系统

Web 缓冲技术可以减小用户的访问延迟, 缓解服务器超载及网络拥塞, 被认为是提高 WWW 性能的重要方法. Web 缓冲课题组提出了近邻分层的协作 web 缓冲模型^[13], 及其原型系统 NHCW. 限于篇幅, 该系统的工作机制不再赘述.

实验的目的是分析 NHCW 系统的重要性能指标 (如平均访问延迟、命中率) 随系统参数的变化规律, 进一步的实验可以预测最优的系统参数. NHCW 系统具有大规模, 生成负载压力大, 对仿真系统的实时能力要求比较高的特点, 所以在设计仿真实验时采用了主机模式.

利用课题组开发的可视化网络建模平台构建仿真实验的网络拓扑, 如图 8 所示. 该平台提供可视化编辑网络模型, 生成仿真实验脚本的能力. 网络模型包括 136 个节点, NHCW 系统包括 web 服务器, 热点目录服务器, 12 个缓冲代理服务器, 每个缓冲代理服务器连接 3 个客户端. 缓冲代理服务器的文档替换算法是 Size 算法, 客户端采用 trace 驱动方式, 选择 NLANR trace^[14]. 希望从初步的实验

中获得 NHCW 系统的平均访问延迟和命中率随缓冲空间大小的变化规律, 实验结果如表 1 所示. 可以看出, 缓冲空间增加时, 客户端的平均访问延迟下降, 命中率上升.

表 1 分布协作 web 缓冲仿真实验结果 (每次实验运行 12 小时, 共 4 次)

| | | | | |
|-------------|------|------|------|------|
| 相对缓冲大小 (%) | 0.5 | 5 | 15 | 30 |
| 平均访问延迟 (ms) | 2727 | 2538 | 2414 | 2358 |
| 命中率 (%) | 5.4 | 7.3 | 9.1 | 10.1 |

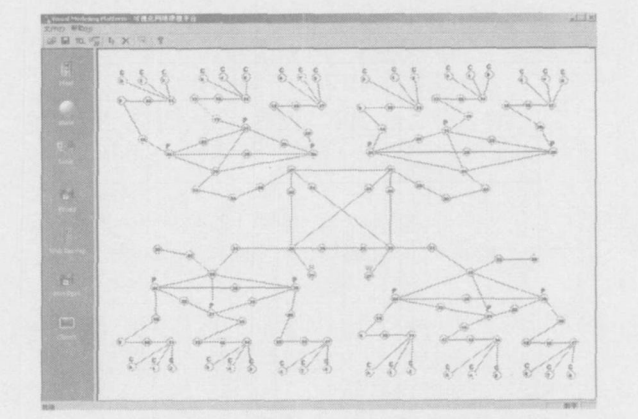


图 8 分布协作 web 缓冲仿真实验网络模型

7 总结

为了在新协议新算法部署到 Internet 之前预测其可用性和性能, 我们研究实现了并行离散事件网络仿真系统. 本文详尽地论述了该系统的体系结构, 以及实时事件调度引擎, 交互机制的设计思想和细节. 通过对仿真系统的延迟模型、带宽模型、丢包模型的验证, 说明了模拟网络行为的正确性和有效性. 以网络路径瓶颈带宽测量和分布协作 web 缓冲系统作为应用实例, 分别开展仿真实验, 说明网络仿真系统可以为研究人员开发正确、高效、健壮的网络协议和算法提供支持.

参考文献:

[1] M Carson D Santay. NIST Net a linux-based network emulation tool[J]. ACM SIGCOMM Computer Communication Review. 2003, 33(3): 111- 126

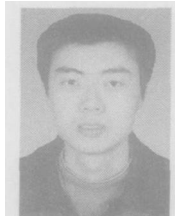
[2] D B Ingham, G D Parrington. Delay line A wide-area network emulation tool[J]. Computing Systems 1994, 7(3): 313- 332

[3] J S Ahn, P B Danzig Z Liu, L Yan. Evaluation of TCP Vegas Emulation and experiment[A]. Proceedings of ACM SIGCOMM [C]. Boston: ACM Press 1995 185 - 195

[4] B D Noble, M Satyanarayanan, G T Nguyen, R H Katz. Trace-based mobile network emulation[A]. Proceedings of ACM SIGCOMM [C]. Cannes, France: ACM Press 1997. 51- 61.

- [5] K Fall Network Emulation in the VINT/NS Simulator [A]. In Proceedings of the fourth IEEE Symposium on Computers and Communications [C]. Egypt: IEEE Press, 1999: 59–67.
- [6] R Simmonds, R Bradford, B Unger. Applying parallel discrete event simulation to network emulation [A]. Proceedings of the 14th workshop on Parallel and Distributed Simulation [C]. Bologna, Italy: ACM Press, 2000: 15–22.
- [7] 李越, 钱德沛. 基于 NS 的分布式并行网络模拟器 [J]. 电子学报, 2004, 32(2): 246–249.
Li Yue, Qian Depei. A distributed parallel network simulator Based on NS [J]. Acta Electronica Sinica, 2004, 32(2): 246–249.
- [8] L G Valient. A bridging model for parallel computation [J]. Communications of ACM, 1990, 33(8): 103–111.
- [9] LBNL's Network Research Group. PCAP Packet Capture Library [EB/OL]. <http://www-ng.ee.lbl.gov/ng.html>
- [10] 李越, 钱德沛, 等. 网络仿真器 NS 问题分析及改进方案 [J]. 系统仿真学报, 2005, 17(11): 555–559.
- [11] Russell Rusty. Netfilter Hack ing HOW TO [EB/OL]. <http://netfilter.kerneNotes.org>
- [12] Ningning Hu, Peter Steenkiste. Evaluation and characterization of available bandwidth probing techniques [J]. IEEE Journal on selected areas in communications, 2003, 21(6): 879–894.
- [13] 张兴军. 分布协作 web 缓冲机制及关键算法研究 [D]. 西安: 西安交通大学研究生院, 2003.
- [14] NLNR. Weekly Access Logs at NLNR's Proxy Caches [EB/OL]. <ftp://ftp.ircache.net/Traces/>

作者简介:



李越男, 1978 年出生于西安, 2005 年获得计算机系统结构专业博士学位, 现为清华大学信息技术研究院助研, 主要研究方向包括网络仿真, 网络性能评价和测量. E-mail: yuelichina@vip.sina.com.



钱德沛男, 1952 年出生于上海, 北京航空航天大学计算机学院教授, 博士生导师, 研究方向为计算机体系结构, 计算机网络等.