

# 一种基于虚拟化平台的网络仿真准实验床

黄锦松<sup>1</sup>, 杨 艺<sup>1</sup>, 王文鼎<sup>1,2</sup>

(1. 宽带无线通信与传感网技术教育部重点实验室, 江苏 南京 210003;  
2. 南京邮电大学 通信与信息工程学院, 江苏 南京 210003)

**摘 要:** 由于为通信网络验证实验构建真实环境的代价过大, 而传统的仿真软件又缺乏真实性, 针对这种情况, 文中结合虚拟化平台技术, 设计了一种网络仿真准实验床系统。通过物理资源虚拟化, 将多个全功能网络节点和模拟的通信链路实体部署在虚拟机层面。系统基于 XEN 准虚拟化环境, 引入 GNS3 和 NetEm 软件包, 分别构造真实路由器和调控通信链路, 再由虚拟机模拟网络端节点。实测表明, 单台物理服务器可部署 6 个全功能网络节点和 5 条通信链路, 配置构成一个哑铃型网络的准实验床。分析了 Reno 和 Vegas TCP 控制算法的公平性实验结果, 并与 NS2 软件仿真对比, 证实所设计的准实验床具有高逼真度。

**关键词:** 准实验床; 虚拟化平台; GNS3; NetEm; 逼真度

**中图分类号:** TP39

**文献标识码:** A

**文章编号:** 1673-629X(2015)08-0208-05

doi:10.3969/j.issn.1673-629X.2015.08.044

## A Para-testbed for Network Simulation Based on Virtualized Platform

HUANG Jin-song<sup>1</sup>, YANG Yi<sup>1</sup>, WANG Wen-nai<sup>1,2</sup>

(1. Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education,  
Nanjing University of Posts and Telecommunications, Nanjing 210003, China;  
2. School of Telecommunications & Information Engineering, Nanjing University of Posts and  
Telecommunications, Nanjing 210003, China)

**Abstract:** Because of the high costs of building real network environment for network verification experiments while the traditional simulators are lack of fidelity, in view of this, present a system of para-testbed for network simulation by the techniques of virtualized platform. Aiming at fidelity and scalability of simulation, propose a deployment approach for multiple fully-functional network nodes and communication link emulators on virtual machine layer by physical resources virtualization. The para-testbed is based on XEN server environment to install realistic routers by open source packages GNS3, to control communication links by NetEm, and to simulate network terminals by virtual machines. An experimental para-testbed consisting of 6 nodes and 5 links is to set up over single a machine to simulate a dumbbell typed network. The results on the fairness experiment of the current Reno and Vegas TCP algorithm are reported and compared with that by NS2 evaluations. It is showed that the proposed para-testbed disposes of the fidelity in network simulation.

**Key words:** para-testbed; virtualized platform; GNS3; NetEm; fidelity

## 0 引 言

迄今为止, 通信网络的实验验证手段主要分为两大类: 实验床和网络仿真<sup>[1]</sup>。实验床(例如 Emulab<sup>[2-3]</sup>)通过部署和配置实际的网络环境进行实验, 由真实的物理设备(如终端、路由器、交换机等)组成。由于其实验条件最接近现实, 所以实验床的实验结果最能反映真实情况。但是, 实验床具有明显的成本劣势: 硬件昂贵、配置费时、容量受限、管理繁重, 并且缺

乏可重复性与可扩展性。

网络仿真(例如 NS-2<sup>[4-5]</sup>)通过对网络设备、链路和协议建立计算模型, 通过软件仿真网络行为, 从而实施对网络功能设计和通信优化所需的实验观测。网络仿真器通常不运行真实协议栈, 不需要配备成本较高的网络设备, 只以软件的方式推演现实环境下的网络变化过程。它的优点在于实验环境容易控制, 可重复性高; 构建环境成本低廉, 配置简单, 方便协议行为测

收稿日期: 2014-09-15

修回日期: 2014-12-18

网络出版时间: 2015-07-21

基金项目: 江苏省科技支撑计划(工业)项目(BE2013019)

作者简介: 黄锦松(1989-), 男, 硕士研究生, 研究方向为通信网络技术; 王文鼎, 博士, 教授, 研究方向为信息通信网络技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150721.1433.018.html>

试。由于网络仿真器的仿真过程过于依赖软件模型,并没有在终端和网络设备运行真实的协议栈,所以仿真结果不一定能准确反映出真实情况,影响其真实性。

近年来得到快速发展的网络虚拟化技术,为通信网络的实验验证提供了一种新的集成手段,以融合传统实验床和网络仿真的技术特点和性能优点。使用真实的硬件配置实验环境,配以仿真软件来代替部分实体功能,在提高了实验环境的可重复性的同时,可以保证实验本身的逼真度。

文中基于 XEN<sup>[6]</sup>、GNS3 (Graphical Network Simulator)<sup>[7]</sup> 和 NetEm<sup>[8]</sup> 设计了一种集成网络准实验床 (Integrated Network Para - Testbed based on XEN, INPTX), 用于创建网络仿真的实验环境,并能够对网络拓扑和网络链路进行灵活控制。INPTX 利用 XEN 对网络节点进行模拟, GNS3 对路由器进行模拟, NetEm 对网络链路进行控制。文中进一步分析了模型实验的可行性和逼真度。

1 虚拟化技术与 XEN 平台

1.1 虚拟化技术

虚拟化技术<sup>[9]</sup> 能够把物理资源统一进行逻辑表示,使资源不再受物理限制的约束。它将物理硬件与操作系统分离开,使得一个物理机上可以同时运行多个不同的操作系统,每个操作系统可以执行多个应用程序。

虚拟化技术按照实现的层次来划分,主要可以分为硬件虚拟化和基于操作系统的虚拟化<sup>[10]</sup>。硬件虚拟化是在基础硬件和客户操作系统之间添加一个虚拟层,称为 Hypervisor。Hypervisor 为每个虚拟机创建一套硬件全仿真实例,然后上面安装客户操作系统。虚拟机运行在 Hypervisor 之上,拥有独立的虚拟化硬件资源和操作系统,与其他虚拟机完全隔离。

Hypervisor 主要分为两类:一类是 Hypervisor 直接运行在硬件层之上,称为裸机型,主要例子有 Citrix XEN 和 Microsoft Hyper-V 等;另一类是 Hypervisor 运行在已有的操作系统上,称为宿主型,主要例子有 VMware Workstation 和 VirtualBox 等。

硬件虚拟化可以分为完全虚拟化和准虚拟化两种。完全虚拟化是指虚拟机模拟了完整的硬件环境,使得客户操作系统不用改动就能安装到虚拟机上,并且客户操作系统也意识不到自己是运行在虚拟的平台上的。由于需要对硬件进行完全模拟,所以对于性能影响比较大。准虚拟化是指通过对客户操作系统进行内核修改,让客户操作系统知道自己运行在虚拟环境下,并且能够配合 Hypervisor 工作,提高性能。XEN 就是通过准虚拟化技术来提高性能的。

1.2 XEN 体系架构

XEN 的体系架构如图 1 所示。

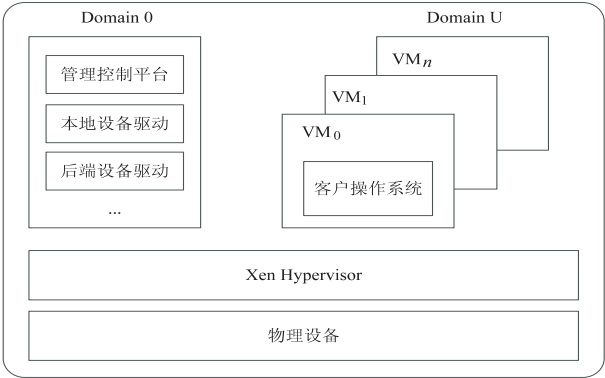


图 1 XEN 体系架构

XEN 的虚拟环境由三个重要部分组成: XEN Hypervisor、Domain 0 和 Domain U。

XEN Hypervisor 是在操作系统和基础硬件之间的一个虚拟层,是整个虚拟化平台的核心。Domain 0 (特权域) 是一个运行在 Hypervisor 之上的特殊的虚拟机。Domain 0 是在 XEN 的内核启动后首先加载的域,负责所有的 Domain U 的创建、关闭、删除等管理工作。Domain U (非特权域) 是运行不同客户操作系统和应用程序的虚拟机。

文中在服务器上安装 XCP (XEN Cloud Platform)<sup>[11]</sup> 来提供 XEN 的虚拟化环境。XCP 是 XEN 的一个开放性云平台,其中包含了 XEN Hypervisor、配置好的特权域和完善的平台管理工具。XCP 在 XEN 的基础上添加了很多管理功能,使得用户能够创建和管理资源池,使用高级存储策略和性能监控功能等。

2 INPTX 的设计与实现

2.1 INPTX 体系架构

如图 2 所示, INPTX 体系架构主要分为四个部分: XCP、管理控制平台、网络节点和网络链路。

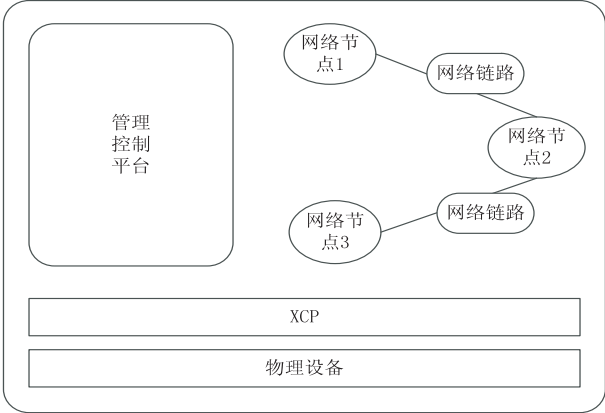


图 2 INPTX 体系架构

(1) 管理控制平台。

管理控制平台使得用户通过 Domain 0 能够直接

访问网络节点,并对网络节点进行配置。Domain 0 与 XCP 提供的管理控制接口交互,通过控制面板模块来管理整个虚拟机环境。文中在此基础上利用 XEN-Center 软件建立一个图形化管理界面,使得实验管理控制更加直观。

### (2) 网络节点。

文中使用 Domain U 中运行的虚拟机模拟网络节点,安装 Ubuntu 作为网络节点的操作系统,网络节点分为两类:虚拟主机和虚拟路由器。虚拟主机节点作为网络终端设备使用,在上面安装应用程序和流量发生器,为网络实验服务。虚拟路由器节点中安装 GNS3 软件,使用 GNS3 中的模拟路由器与虚拟机的网络接口相连,从而与外界通信。虚拟路由器节点相当于把虚拟机当作路由器使用,因为 GNS3 运行真实的 Cisco IOS,所以虚拟路由器具有等同于真实路由器的功能。

### (3) 网络链路。

网络链路用于在任意网络节点之间提供通信链接,并且能够模拟真实链路的带宽、丢包率和延时等特性。文中网络链路的连接功能由 XCP 的网络虚拟化环境实现,而链路的特性由 NetEm 实现。

## 2.2 网络链路模拟

### (1) 虚拟化架构。

XCP 在一台物理机器上运行多个虚拟机,各虚拟机作为网络节点进行通信时需要共享该物理机的网卡功能,XCP 使用虚拟化技术完成转发虚拟网络接口和物理网络接口之间的数据包工作。两个网络节点之间的链路构成如图 3 所示。

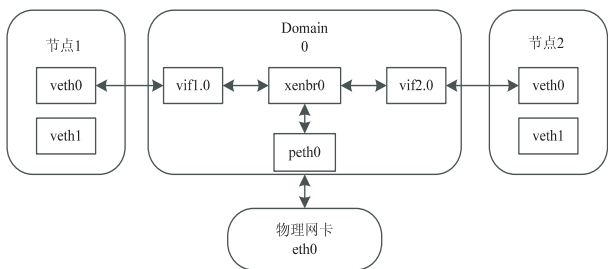


图 3 两个节点间的链路构成

默认的 XCP 网络虚拟化架构由四部分组成:前端设备、后端设备、网桥和物理网卡。在 XCP 中,只有 Domain 0 可以直接访问物理网络接口,Domain U 必须把网络分组发送到 Domain 0 进行处理。XCP 中引入了分离设备驱动模式,在 Domain U 中建立前端设备,在 Domain 0 中建立后端设备,客户操作系统向前端设备发送请求,前端设备通过 XEN Hypervisor 将请求以及 Domain U 的身份信息发送到后端设备。

图 3 中,前端设备驱动在节点 1 和节点 2 中创建网络接口,文中把每个节点中的第 Y 个网络接口命名为 vethY。后端网络接口在 Domain 0 中与前端网络接

口相连,文中用 vifX.Y 命名。其中,X 是指节点的 ID,Y 是对应的虚拟网卡号,即 vethY。XCP 默认使用网桥模式对数据包进行转发,实现对物理网卡的共享。文中用 xenbrN 来命名网桥,连接后端网络接口和物理网卡,并把物理网卡 ethN 重命名为 pethN。

### (2) 模拟链路。

NetEm 提供网络链路的模拟功能,它通过与 tc<sup>[12]</sup>的配合使用,能够准确模拟链路延时、分组丢失等情况。NetEm 已经集成于 Linux 2.6 及以上的内核版本中,它的链路模拟功能是通过设置队列规则实现的,队列规则存在于协议输出和网络设备之间。NetEm 命令行功能已嵌入到 tc 之中,用户通过在 tc 命令中增加参数来设置队列规则,设定分组延时、丢包,模拟数据包的重复乱序等。

## 3 网络模拟实验

### 3.1 实验环境

INPTX 是建立在物理服务器构成的虚拟化环境之上的,通过安装 XCP1.0 来提供 XEN 的虚拟化支持。网络节点统一安装 Ubuntu 12.04 作为操作系统,内核版本为 3.2.0-34-virtual,每个节点均分配 1 G 内存和 1 个虚拟 CPU。

实验过程中网络节点使用默认的 TCP 协议栈配置,TCP 发送缓冲区的默认大小为 16 kB,TCP 接收缓冲区的默认大小为 85.3 kB。每个网络节点安装有 GNS3 和 NetEm 软件,并且使用 iperf 来产生业务流,测试 TCP 连接的最大带宽。为便于分组监测,使用 tcpdump 截获在网络中传输的数据包,使用 tcptrace 来分析截获的数据包,使用 tcprobe 来记录 TCP 连接的参数。

### 3.2 网络链路性能模拟实验

创建网络节点 Node-1 和 Node-2,在连接网络节点之间的链路上添加 NetEm 模块。

对于链路不作任何限制的情况下使用 iperf 软件测试两个节点间的 TCP 连接性能。把 Node-2 作为服务器端,让它监听在 5001 TCP 端口。Node-1 作为客户端,使用 iperf 软件与 Node-2 建立连接,从 Node-1 向 Node-2 发送数据包,测试带宽。可以使用 ping 命令测试两节点间的延时和丢包率。

#### (1) NetEm 控制链路延时。

同样把 Node-2 作为服务器端,在 Node-1 中使用 NetEm 和 tc 的相关命令,对链路参数 delay 进行修改,设置链路时延。该命令的作用是单向的,只对从 Node-1 发出的数据包有效。

#### (2) NetEm 控制链路丢包率。

实验环境不变,在 Node-1 中用 NetEm 和 tc 的相



关命令对链路参数 loss 进行修改,设置其丢包率。可以通过 ping 命令来验证该设置是否生效,同样该命令也是单向有效的。

(3) NetEm 控制链路带宽。

在相同的实验环境下,在 Node-1 中通过 NetEm 和 tc 相关命令对链路参数 rate 进行修改,设置链路带宽,可以通过 iperf 来测试。

3.3 哑铃模型模拟实验

为验证 INPTX 的可行性和真实性,配置建立如图 4 所示的哑铃模型拓扑,由 4 个网络节点和 2 个模拟路由器共同组成。路由器功能由 GNS3 模拟实现,使用 NetEm 设置 Router-1 和 Router-2 之间的链路带宽为 1 Mbps 左右,时延为 10 ms。

4 个网络节点共同竞争 Router-1 和 Router-2 之间的瓶颈带宽。将 Node-2 作为服务器端,Node-1 作为客户端,在 Node-1 和 Node-2 之间建立 TCP 连接,用作实验。然后在 Node-3 和 Node-4 之间引入 TCP 业务流用于竞争瓶颈带宽,通过分析比较在相同 RTT 情况下 Vegas<sup>[13]</sup> 和 Reno<sup>[14]</sup> 两种算法的公平性来验证 INPTX 的可行性。

作为对照实验,通过 NS2 软件建立相同的哑铃拓扑,链路性能、业务流设置都与 INPTX 中相同,通过分析对比 NS2 和 INPTX 的仿真结果,验证该准实验床的逼真度。

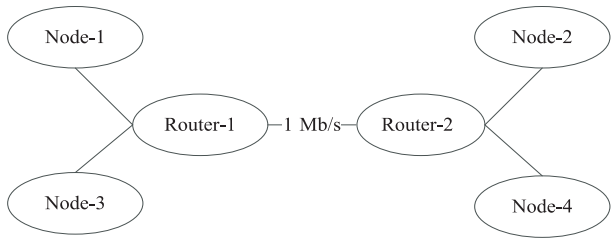


图 4 哑铃模型拓扑

对照实验的计算对象为 Vegas 和 Reno 算法的公平性。实验中 Node-1 和 Node-2 之间的 TCP 连接先建立起来,并且使用 Vegas 算法,而 Node-3 和 Node-4 之间的业务流在第 4 s 时开始,且使用 Reno 算法,实验结果如图 5 和图 6 所示。

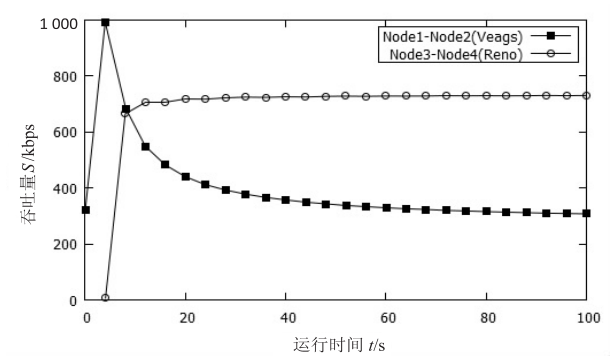


图 5 Vegas & Reno 公平性验证(NS2)

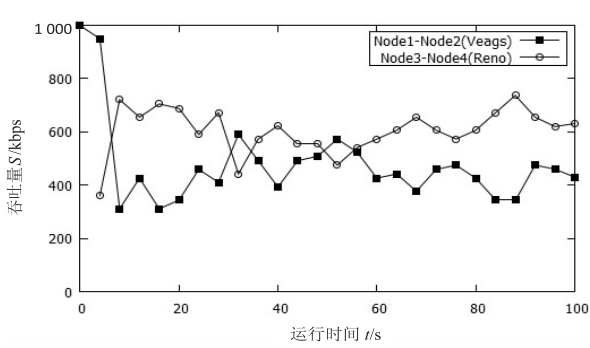


图 6 Vegas & Reno 公平性验证(INPTX)

从图 5 和图 6 可以看出,无论是在 NS2 仿真软件中还是在 INPTX 准实验床中,Vegas 和 Reno 算法具有相似的竞争效果。NS2 仿真结果显示,两条业务连接,各自占用带宽之和,趋于 1 Mbps。INPTX 实验结果表明,占用带宽之和保持为 1 Mbps。两种仿真均证实 Vegas 与 Reno 共享所设的瓶颈链路的带宽(1 Mbps)。

在 NS2 仿真结果中,Reno 算法最后占用带宽稳定在 700 kbps 左右,而 Vegas 算法占用的带宽稳定在 300 kbps 左右。在 INPTX 仿真结果中,Reno 和 Vegas 算法竞争时间较长,但是平均占用带宽分别为 630 kbps 和 440 kbps。Vegas 与 Reno 在带宽争用的主导性方面,NS2 与 INPTX 的结果是一致的。

对比图 5 和图 6,Reno 和 Vegas 争用带宽的比重随时间变化上存在明显差异。NS2 中是在第 10 s 左右完成竞争,吞吐量在数值上仅有一次变迁,之后两者就相对平稳。在 INPTX 中,Reno 和 Vegas 从约 10 s 开始交替,50 s 之后结束。图 5 曲线表现出平衡走势,而图 6 中曲线存在明显的抖动。

图 6 所示的吞吐量随时间变化关系,反映了实际环境中影响吞吐量的因素不仅仅只有拥塞窗口,还有可能是对端接收缓存大小或是网络节点上应用程序的执行过程差异影响等。然而在 NS2 中是把这些因素全部理想化的,所以 NS2 仿真的结果不能完全反映真实情况。INPTX 由全功能网络节点和模拟通信链路配合使用,能够再现真实的实验环境。图 6 中 Reno 和 Vegas 曲线多次相交,清楚地揭示了两种算法竞争带宽的细节,60 s 到 100 s 之间的曲线仍有起伏,但是整体趋势可以看出 Reno 算法对链路带宽的抢占性要明显强于 Vegas<sup>[15-17]</sup>。

4 结束语

文中分析了现有的通信网络的实验验证手段的优缺点,基于 XEN、GNS3 和 NetEm 设计了一种集成网络准实验床(INPTX)。INPTX 中通过 XENCenter 连接 Domain 0 管理各个虚拟机。通过虚拟化技术在一台服务器上运行多个全功能网络节点,在网络节点上安装

GNS3 软件,运行真实的协议栈,在逻辑上模拟真实路由器,以提高其真实性,由 tc 和 NetEm 配合使用控制链路性能。通过实验验证了 INPTX 对链路、网络节点的控制能力,通过分析 Reno 和 Vegas 算法的公平性实验结果,与 NS2 软件仿真对比,证实所设计的准实验床具有高逼真度。

#### 参考文献:

- [1] Siaterlis C, Masera M. A survey of software tools for the creation of networked testbeds[J]. International Journal on Advances in Security, 2010, 3(1): 1-12.
- [2] White B, Lepreau J, Stoller L, et al. An integrated experimental environment for distributed systems and networks[C]//Proceedings of the 5th symposium on operating systems design and implementation. [s. l.]: [s. n.], 2002: 255-270.
- [3] 秦董洪, 陈智勇, 杨家海. 基于 Emulab 的网络仿真实验平台研究[J]. 实验室科学, 2013, 16(3): 92-95.
- [4] Mahrenholz D, Purushothaman I. The network simulator ns-2[EB/OL]. (2011-11-04)[2014-04-15]. <http://www.isi.edu/nsnam/ns>.
- [5] 马元飞. NS2 条件网络性能分析实践[J]. 电脑与电信, 2013(1): 39-41.
- [6] Zhang Enming. Xen project, a linux foundation collaborative project[EB/OL]. (2014-05-20)[2014-06-24]. <http://wiki.xensource.com/wiki>.
- [7] Grossmann J, Saraiva F J. GNS3, graphical network simulator[EB/OL]. (2014-03-25)[2014-07-02]. <http://www.gns3.net>.

(上接第 207 页)

入研究,基于设计所得硬件搭建实验平台,对数据采集功能作了反复测试。文中给出的方案能够有效减小数据采集系统的体积并极大地扩展其系统功能,对于飞行器状态监控与故障诊断系统的进一步设计具有一定的参考价值。

#### 参考文献:

- [1] 何 宾. Xilinx All Programmable Zynq-7000 SoC 设计指南[M]. 北京:清华大学出版社, 2013.
- [2] 陆佳华, 江 舟, 马 岷. 嵌入式系统软硬件协同设计实战指南[M]. 北京:机械工业出版社, 2013.
- [3] 王 冰. 基于 STM32 和 FPGA 的多通道多功能数据采集器[D]. 成都:西南交通大学, 2012.
- [4] 王正浩. 嵌入式 8 通道高速数据采集器的设计[D]. 哈尔滨:哈尔滨理工大学, 2010.
- [5] Wang Jiannong, Wang Wei. The common data acquisition system based on Arm9[C]//Proc of international conference on electronic measurement & instruments. [s. l.]: [s. n.],

gns3.net.

- [8] Hemminger S. Network emulation with netem[C]//Proceedings of the 6th Australia's national Linux conference. [s. l.]: [s. n.], 2005.
- [9] 何秀森. 虚拟化技术基本原理及应用[J]. 电信网技术, 2014(6): 16-18.
- [10] Calarco G, Casoni M. On the effectiveness of Linux containers for network virtualization[J]. Simulation Modelling Practice and Theory, 2013, 31: 169-185.
- [11] Citrix CloudPlatform. XCP overview[EB/OL]. (2013-01-20)[2014-01-27]. [http://wiki.xensource.com/wiki/XCP\\_Overview](http://wiki.xensource.com/wiki/XCP_Overview).
- [12] Hubert B, Graf T, Maxwell G, et al. Linux advanced routing & traffic control[EB/OL]. (2012-05-20)[2013-01-20]. <http://lartc.org>.
- [13] 秦 军, 袁翰林, 陈 迪. 异构无线网络中 TCP Vegas 算法的研究与改进[J]. 计算机技术与发展, 2012, 22(4): 88-92.
- [14] 牛 磊, 王 峰, 刘冬冬, 等. TCP Reno 拥塞控制的改进算法[J]. 福建电脑, 2014, 30(2): 7-9.
- [15] Feng W, Vanichpum S. Enabling compatibility between TCP Reno and TCP Vegas[C]//Proceedings of IEEE symposium on applications and the Internet. [s. l.]: IEEE, 2003: 301-308.
- [16] 肖 锴, 章国安, 杨 云. 基于 NS2 的 TCP Vegas 算法的仿真与研究[J]. 实验技术与管理, 2012, 29(8): 92-95.
- [17] 刘国柱, 高文娟. 基于 TCP Reno 和 TCP Vegas 拥塞控制性能研究[J]. 计算机工程与设计, 2011, 32(2): 434-437.

2011: 324-327.

- [6] 李文刚. 基于 FPGA 的高速高阶 FIR 滤波器设计[D]. 成都:电子科技大学, 2005.
- [7] 刘文峰, 李程远, 李善平. 嵌入式 Linux 操作系统的研究[J]. 浙江大学学报:工学版, 2004, 38(4): 447-452.
- [8] Stevens W R, Rago S A. Advanced programming in the UNIX environment[M]. [s. l.]: Addison-Wesley, 2014.
- [9] Jones M T. GNU/LINUX application programming[M]. [s. l.]: Charles River Media, 2008.
- [10] 刘 磊, 张凤荔, 秦志光. 基于 U-boot 构建嵌入式 Linux 的 Bootloader[J]. 计算机应用研究, 2007, 24(12): 238-240.
- [11] Bovet D P, Cesati M. Understanding the Linux kernel[M]. [s. l.]: O'Reilly Media, 2002.
- [12] 邵长彬, 李洪亮. 用 Busybox 制作嵌入式 Linux 根文件系统[J]. 微计算机信息, 2007, 23(10-2): 48-50.
- [13] Corbet J, Kroah-Hartman G, Rubini A. Linux device drivers[M]. [s. l.]: O'Reilly Media, 2005.
- [14] 张曦煌, 柴志雷. 嵌入式 Web 服务器中 CGI 的特点及实现[J]. 小型微型计算机系统, 2003, 24(11): 2046-2048.

一种基于虚拟化平台的网络仿真准实验床


作者:

黄锦松, 杨艺, 王文鼎, HUANG Jin-song, YANG Yi, WANG Wen-nai

作者单位:

黄锦松, 杨艺, HUANG Jin-song, YANG Yi (宽带无线通信与传感网技术教育部重点实验室, 江苏 南京, 210003), 王文鼎, WANG Wen-nai (宽带无线通信与传感网技术教育部重点实验室, 江苏 南京 210003; 南京邮电大学 通信与信息工程学院, 江苏 南京 210003)

刊名:

计算机技术与发展 

英文刊名:

Computer Technology and Development

年, 卷(期):

2015(8)

引用本文格式: 黄锦松. 杨艺. 王文鼎. HUANG Jin-song. YANG Yi. WANG Wen-nai 一种基于虚拟化平台的网络仿真准实验床[期刊论文]-计算机技术与发展 2015(8)