doi:10.3772/j.issn.1002-0470.2015.07.005

基于 NS-3 构建网络环境的虚实结合技术的研究①

李广荣②* 王 琨** 张兆心③*

(*哈尔滨工业大学计算机学院 哈尔滨 150001)

(**国家计算机网络应急技术处理协调中心 北京 100029)

摘 要 研究了网络环境的构建。针对构建网络环境的传统试验床技术代价高,网络仿真真实性差的缺点,提出了一种基于网络仿真器 NS-3 的新的网络环境构建技术。为了实现物理网络和仿真网络的对接(简称虚实对接以及物理网络和仿真网络的路由策略,该技术采用基于虚拟网卡桥接的虚实对接设计。为实现虚实之间的数据通信,虚拟网卡桥接采用基于轮询机制的监测算法(PMA),路由策略采用基于广度优先搜索的路由改进算法(BFSIA)。利用 ping 命令和 IRC 僵尸网络对构建网络进行了测试,测试结果表明用这项技术可进行真实的数据包转发。通过设定不同规模的网络拓扑测试了 BFSIA 计算路由所消耗的时间,并和 NS-3 全局路由算法进行了比较,证明了 BFSIA 的优越性。

关键词 网络仿真, 拓扑形式化描述, 虚拟网卡桥接, 路由算法

0 引 言

为了增强网络安全,需要为了更好地研究和分析网络安全事件,提出有效的防御措施,因而需要对安全事件进行重现。由于网络安全事件具有不可控性和易变性,无法在实际网络中进行测试,往往需要搭建虚拟网络环境进行测试,因此如何构建虚拟网络环境成为了网络技术人员所面临的一项重大课题。在真实试验方面,张黎辉等人[1]利用真实主机构建试验床的技术进行了分布式拒绝服务(DDoS)攻击的防御测试,但是测试仅使用了两台主机作为攻击源,无法达到 DDoS 攻击洪范的需求。相关工作还有全晓莉等[2]及冯陈伟[3]提出了基于虚拟化网络试验平台的构建技术,虽然相比张黎辉等人直接利用主机进行测试在规模上有所提高,但需要手动配置 VMWare 虚拟机技术和操作系统虚拟化技术,导致无法进行大规模的网络配置。张云等利用 KVM 虚拟机和快照技术实现的虚拟机的自动化创建和网络部署技术,大大提高了试验床部署的效率,但采取的 DDoS 试验仍以个位数的主机作为攻击源,并且在构建大量虚拟机时需要一定数量的物理服务器作为支撑[4]。为解决试验床造价不菲,规模受到限制的问题,研究人员开始着手于虚拟仿真器的研究。袁晓在文献[5]中介绍了当前主流的 4 种网络仿真器:NS、OPENNET、CORE 以及 Mininet,并给出了各个软件器的仿真规模,相比试验床技术,网络拓扑规模提高了很多。文献[6]使用 NS-3 仿真器构建了机会网络,对 Epidemic 路由协议算法进行了模拟仿真并通过对仿真结果的分析,评价了 Epidemic 路由协议算法在各种环境下的性能。倪晓伟在 NS-3 平台上实现了 DNCA 信道分配算法和改进了的 Ad hoc 按需距离矢量(AODV)路由协议,并通过仿真验证了其理论上的进步性,通过改进的路由

① 国家科技支撑计划(2012BAH45B01),国家自然科学基金(61100189,61370215,61370211)和国家信息安全计划(2014A085,2015A072)资助项目。

② 男,1989 年生,硕士生;研究方向:网络安全;E-mail: 316082475@qq.com

③ 通讯作者,E-mail: heart@hit.edu.cn

(收稿日期:2015-05-18)

协议,有效提高了多信道的信道利用率,降低了信道间的相互干扰,显著提高了网络吞吐量[7]。但是,两者的试验都仅基于 NS-3 仿真器,没有和物理网络进行交互,在真实性上存在一定的差异。针对真实网络试验床试验和虚拟网络仿真试验的各自缺点,华东理工大学的虚实结合试验则较好地将虚拟仿真和远程交互试验结合在一起进行温度检测和控制试验[8]。廖健等[9]也利用桥接器将半实物系统与数学模型进行联合试验,并通过实时共享内存网来连接互联网络设备以保证仿真的实时性。他们虽然完成了虚实结合的目的,但主要基于自动化控制和电路设计方面,没有进行网络案例的测试。

针对当前试验床技术规模低、仿真试验真实性差以及虚实结合试验没有进行网络案例测试等问题,本研究基于 NS-3 仿真器利用虚拟网卡桥接技术将 NS-3 仿真器和物理网络进行连接,构造了虚实对接的通信模型,将物理网络中的流量导入到 NS-3 仿真器中,同时将 NS-3 仿真器中的流量导出到物理网络中,实现了物理网络和 NS-3 仿真网络结合的目的,即虚实结合试验,并通过仿真节点加入物理网络中的互联网中继聊天(Internet relay chat, IRC)服务器的测试用例证明了模型的可行性。

1 基于 NS-3 的虚实结合模型

虚实结合模型包括拓扑形式化描述、虚实对接、路由策略。拓扑形式化描述用于完成实际网络拓扑到仿真可识别网络拓扑的转换;虚实对接完成物理网络和仿真网络边界节点的连通;路由策略以试验拓扑为输入,计算物理网络和仿真节点之间的路由并进行路由配置,在虚实对接基础之上实现物理网络和整个仿真网络之间的互联互通。虚实结合模型如图 1 所示。

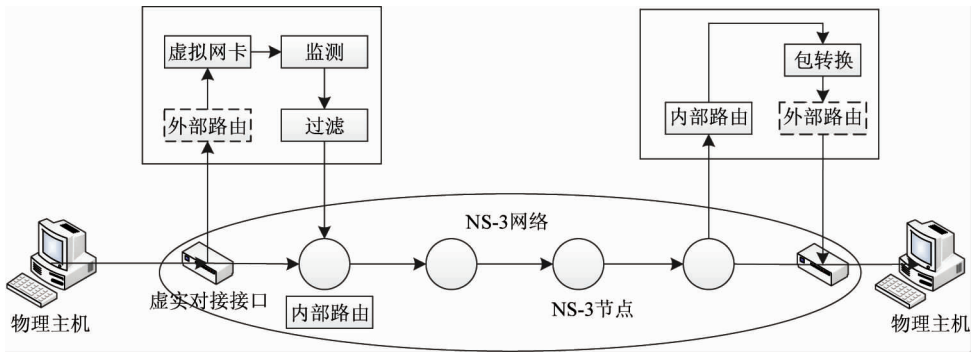


图 1 虚实结合模型图

1.1 拓扑形式化描述

在根据实际网络构建用于网络安全事件重现的仿真网络环境时,需要首先对实际网络中的各个网络元素例如主机、路由器、链路等进行描述。使用形式化描述处理网络环境数据,可以解决不同系统异构网络环境数据的兼容性问题。形式化处理后,实现数据归一化处理,进而可以转换成仿真系统可以处理的数据结构。形式化描述主要包括网络中主机的形式化描述、网络中路由器的形式化描述以及链路的形式化描述:

(1)主机的形式化描述

主机分为物理节点和 NS-3 中的仿真节点。主机主要包括节点的序列号、IP 地址、节点所在的区域,即为物理节点还是仿真节点。同时为了和路由器进行区分,还需要一个字段用于标示主机和路由器节点。因此,对主机的形式化描述为四元组 H = (H _ ID,H _ IP,H _ TYPE,H _ AREA),其中 H _ ID 为主机序列号,按顺序递增,表示网络中的特定主机;H _ IP 参数定义了节点分配的 IP 地址;H _ TYPE 指定节点为路由器节点还是主机节点;H _ AREA 指定当前节点为物理节点还是仿真节点。

(2)路由器的形式化描述

仿真网络中的路由器节点包括终端路由器和非终端路由器。路由器的形式化描述除了 IP 地址为

多个,其它和主机描述基本相同。因此,对路由器的形式化描述为 R = (R _ ID,R _ IPL,R _ TYPE,R _ AREA),其中 R _ ID、R _ TYPE 和 R _ AREA 的参数意义和主机描述相同,R _ IPL = (R _ IP1,R _ IP2,……)为路由器端口的 IP 地址列表。

(3)链路的形式化描述

链路关系主要有链接的节点序列号、链接的 IP 地址、链路的延迟、带宽和类型等。因此,本文中对链路的形式化描述为 L = (L _ SRCNO,L _ DSTNO,L _ SRCIP,L _ DSTIP,L _ DELAY,L _ BINDWIDTH,L _ TYPE),其中 L _ SRCNO 和 L _ DSTNO 表示参与链接的两个节点的源序列号和目的序列号,目的序列号根据链路类型不同可为整型也可为列表;L _ SRCIP 和 L _ DSTIP 对应源目的 IP 和目的 IP,目的 IP 同目的序列号一样可为字符串也可为列表;L _ DELAY 和 L _ BINDWIDTH 分别表示链路的延迟和带宽;L _ TYPE 指定链路的类型为仿真网络内部链路还是仿真网络和物理网络之间的链路,用来决定是否创建虚实接口。

构建如图 2 所示网络拓扑,对主机、路由器以及链路关系的拓扑形式化描述如下:

```
node[
    {
        "ip": "10.1.2.1, 10.1.3.1",
        "area": 0,
        "type": 1,
        "no": 0
    },
    {
        "ip": "10.1.1.1, 10.1.1.2",
        "area": 0,
        "type": 1,
        "no":1
    },
    {
        "ip": "10.1.1.2",
        "area": 0,
        "type": 0,
        "no":2
    },
    {
        "ip": "10.1.1.2",
        "area": 0,
        "type": 0,
        "no":3
    },
    {
        "ip": "10.1.1.2",
        "area": 0,
        "type": 0,
        "no":3
    },
    {
        "ip": "10.1.3.2",
        "area": 1,
        "type": 0,
        "no": 5
    },
]
```

```
link[
    {
        "delay": 0.01,
        "datarate": 5000.0,
        "src _ no": 1,
        "dst _ no": [
            2,
            3,
            4
        ],
        "src _ ip": "10.1.1.1",
        "dst _ ip": [
            "10.1.1.2",
            "10.1.1.3",
            "10.1.1.4",
        ],
        "type": "csma"
    },
    {
    {
        "delay": "2ms",
        "datarate": "10Mbps",
        "src _ no": 0,
        "dst _ no": 1,
        "src _ ip": "10.1.2.1",
        "dst _ ip": "10.1.2.2",
        "type": "p2p"
    },
    {
        "delay": "2ms",
        "datarate": "10Mbps",
        "src _ no": 0,
        "dst _ no": 5,
        "src _ ip": "10.1.3.1",
        "dst _ ip": "10.1.3.2",
        "type": "tap"
    }
]
```

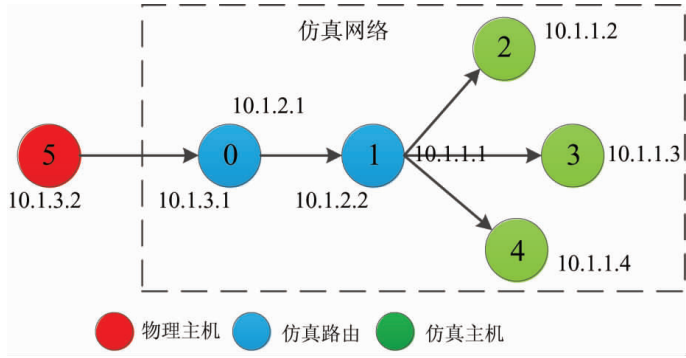


图 2　网络拓扑图

1.2　基于虚拟网卡桥接的虚实对接设计

完成物理网络和仿真网络的对接需要实现两个过程。一个是物理网络和仿真网络临界区域的对接实现，以下简称虚实对接；另一个是物理网络和仿真网络的路由实现。基于以上两个过程，本研究设计了如图 3 所示的虚实对接架构。该架构主要包括虚拟网卡桥接和路由模块。其中虚拟网卡桥接用于完成虚实对接，而路由模块主要包含物理网络和仿真网络的外部路由以及 NS-3 仿真网络的内部路由，完成物理网络和仿真网络内部之间的路由通信。

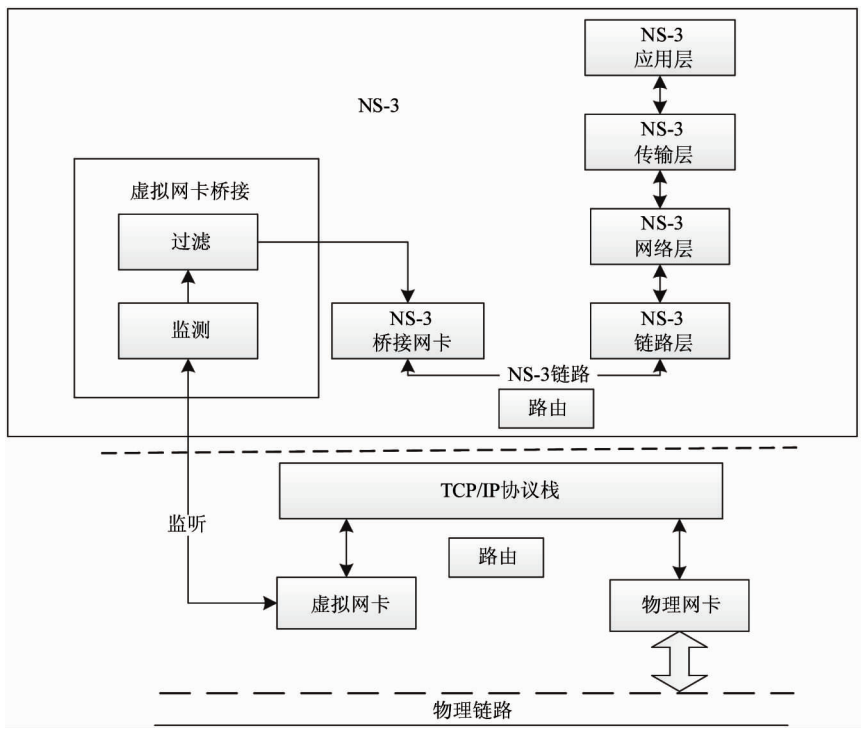


图 3　虚实对接设计

1.2.1　虚拟网卡桥接

虚拟网卡桥接主要基于虚拟网卡技术，利用桥接的方式将 Linux 主机中的虚拟网卡与 NS-3 仿真网卡进行桥接，使物理网络中的数据包能够转发到被桥接的 NS-3 仿真网卡中，同时也能够使仿真网络的数据包被转发到物理网络中，进而达到虚实对接的目的。在虚实对接中，绑定桥接网卡的 NS-3 节点仅用于物理网络和仿真网络间数据包的转发，不参与仿真应用的绑定。

虚拟网卡桥接主要包含虚拟网卡模块、监测模块和过滤模块三个部分。虚拟网卡用于内核态和用户态交互的接口，监测模块负责监测虚拟网卡，过滤

模块用于过滤来自物理网络中的数据包,并转换为仿真数据包。

(1)虚拟网卡模块

虚拟网卡是一种基于 Linux 或 Unix 的虚拟网络设备,既可以作为点对点设备,也可以作为以太网设备。作为虚拟网卡驱动,其驱动程序的数据接收与发送并不直接和物理网卡进行交互,而是通过用户态来转交。在 Linux 下,完成核心态和用户态之间数据的交互有多种方式:可以利用 socket 创建特殊套接字来实现数据交互,也可以通过 proc 文件系统来创建文件进行数据交互,还可以使用设备文件的方式,访问设备文件会调用设备驱动相应的驱动程序。本文采取的虚拟网卡驱动就是利用设备文件实现用户态和核心态的数据交互。

(2)监测模块

虚拟网卡仅仅提供了用户态和核心态之间数据交互的接口,而何时进行交互则由监测模块来决定。监测模块首先选择监听的描述符,接着采取轮询的方式进行监听,一旦发现监听的描述符有事件发生,就通知响应的程序进行处理。监听的描述符主要有两个,一个是虚拟网卡对应的描述符,用于从虚拟网卡中读取数据;另一个是管道描述符,用于结束当前的监听,由管道的另一头进行通知。本文采取基于轮询机制的监测算法(polling monitor algorithm, PMA),算法实现如下:

```
INPUT
    监听描述符 tap_fd, m_evpipe[0]
BEGIN
    定义监听集合 rfds
    FD_ZERO (&rfds)
    FD_SET (tap_fd, &rfds)
    FD_SET (m_evpipe[0], &rfds)
    for (;;) do:
        readfds ← rfds
        select (nfds, &readfds, NULL, NULL, NULL)
        if FD_ISSET (m_evpipe[0], &readfds) do:
            read((m_evpipe[0])
            break
        end if
        if FD_ISSET (m_fd, &readfds) do:
            read(tap_fd)
            Notify()
        end if
    end for
END
```

通过监测模块,每当有数据包的目的地址为仿真网络中的节点,虚拟网卡桥接都能立刻从虚拟网卡中进行读取,并将数据包交由过滤模块,并最终发送到 NS-3 网络中,减少丢包率,如果使用 TCP 协议进行传输,则没有丢包。

(3)过滤模块

物理网路中的数据包是 TCP/IP 协议通信传输中的基本数据单位,包含着 OSI 模型中的各层信息,依据协议的不同,包的类型可以分为很多种。而 NS-3 中的数据包则由一个个类构成,以指针指向的方式进行传递,因此,需要对物理网路中的数据包进行转换才能交由 NS-3 仿真器进行处理。由于虚拟网卡虚拟的是以太网设备,因此本文实现的过滤功能主要获取物理网络中的以太网数据包或者符合 802.2 协议的数据包,并将数据包转换为 NS-3 的仿真包,再交由 NS-3 的网卡进行处理。

1.2.2 路由策略

虚实对接虽然完成了物理网络和 NS-3 仿真网络间的通信,但是这种通信仅限于物理网络和 NS-3 仿真网络中绑定虚拟网卡的节点间,而 NS-3 网络中的路由计算也只能获取仿真节点间的路由信息,无法获得到达物理网络的路由信息,因此,这并非是真正意义上的虚实对接。为了完成仿真网络内部的所有节点和物理网络的连通,还需要对 NS-3 仿真器中的节点进行一定的路由配置。

本文采取基于最短路径的静态路由,依据各链路的代价都为 1,采取基于广度优先遍历的路由改进算法(breadth first search improved algorithm, BFSIA)。路由计算以整个网络拓扑为输入,即网络拓扑包括物理主机和仿真主机,得到 NS-3 内部节点和物理网络间的路由。由于物理网络的规模远远小于仿真网络,该算法选取物理主机作为起点,通过广度优先的方式遍历整个网络拓扑,最终得到整个仿真网络和物理网络间的路由信息。算法实现如下:

```
Input:拓扑图 Graph,源节点 srcnode,目的节点列表 dstnodes
Output:路由表 Route
BEGIN
    定义遍历列表 visit、路径列表 path、访问队列 queue
    nodenum = Graph.node
    path.append(srcnode)
    Enqueue(queue,srcnode)
    while ! Empty(Q) do:
        i ← Front(queue)
        DeQueue(queue)
        for j in Graph.adjacent(j) do:
            if visit[j] = = false do:
                visit[j] ← true
                path[j] ←path[i]
                path[j].append(j)
            end if
            if j in dstnodes do:
                dstnodes.delete(j)
                if dstnodes is null do:
                    break
                    end while
                end if
            end if
        end for
    end while
END
```

计算结束后,得到一张整体路由表,其中包含所有仿真节点到达物理节点的路由信息。依据获得的路由表,在仿真开始前根据节点 ID 将相应的路由信息配置到对应的仿真节点中,同时在物理主机中设置静态路由,将所有到达仿真网络的数据包交给桥接虚拟网卡,这样就可以完成整个物理网络和仿真网络的互联互通。

2　试 验

本研究共进行 3 组试验,虚实通信测试,用于表明物理网络和仿真网络之间可以完成实时通信;NS-3 节点加入 IRC 服务器试验,用于验证通过桥接虚拟网卡可以完成实际网络案例的测试;路由计算效率测试,用于表明不仅可以测试试验给出的网络拓扑还可以进行大规模的网络拓扑测试。

试验使用两台物理主机,其中一台为 Windows 主机,用于运行 IRC 服务器,另一台为 Linux 主机,用于运行 NS-3 仿真器。试验使用的网络拓扑如图 4所示,其中竖线左边的为 Windows 主机,IP 地址为 172.29.153.54,竖线右边的为运行 NS-3 的 Linux 主机,IP 地址为 172.29.153.3,NS-3 中仿真节点分布以及 IP 地址分配如图 4 所示,其中仿真节点 6 绑定桥接虚拟网卡。

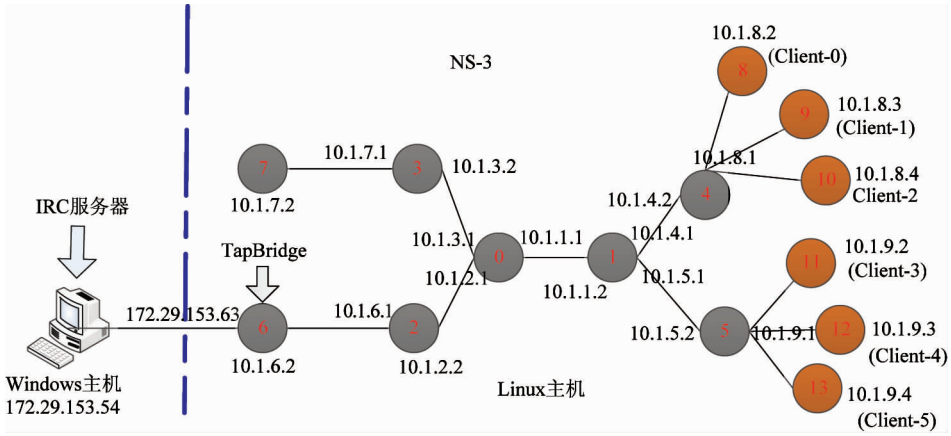


图 4　试验拓扑

2.1　虚实通信测试

本节利用 ping 命令进行虚实通信测试,并对实际延迟时间和预计延迟时间进行比较,以证明实时性。在 NS-3 中,链路延迟主要来自于点对点链路之

间的延迟,因此预计延迟时间 = 物理主机到达虚拟网卡的往返时间 + 点对点链路延迟时间 × 链路条数 ×2。在进行实际时间统计时,由于每一次 ping 命令的延迟时间都是变化的,本文采取对同一目的进行 30 次 ping 命令的延迟时间统计,并取平均值作为当前目的延迟时间的统计方法。试验中利用物理主机 172.29.153.54 ping 仿真节点 10.1.8.2,经过 3 条点对点链路,多次测试到达虚拟网卡的时间为 0.8ms,得到的预计延迟时间和实际延迟时间如图 5 所示。

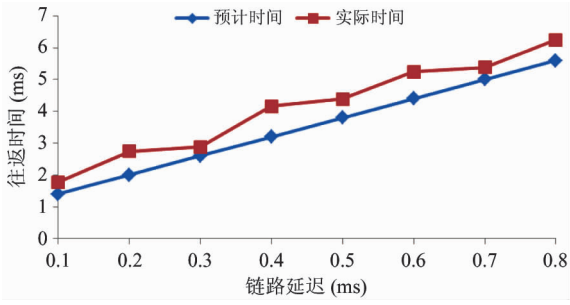


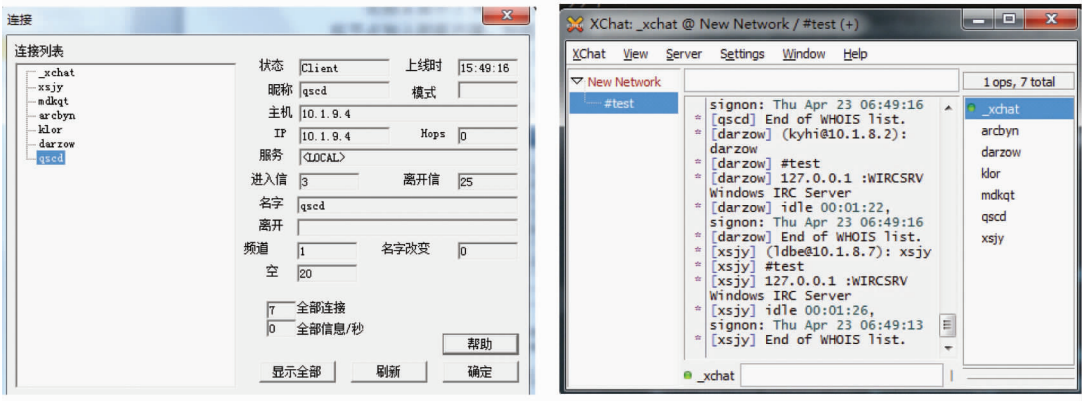
图 5　虚实通信结果图

试验中预计时间为斜率固定的直线,而实际时间为一条回归直线,两条直线都依据链路延迟呈递增趋势。实际延迟时间基本多于预计延迟时间 0.5ms左右,多余时间为 ping 应用解析数据包和回应数据包的处理时间,因此,通过虚拟网卡桥接技术不仅可以实现虚实网络之间的通信,且可以达到实时性。

2.2　NS-3 节点加入 IRC 服务器

为了表明虚拟网卡桥接不仅可以完成虚实通信,还可以在虚实通信的基础上像试验床技术和网络仿真技术一样进行具体的网络试验,本文通过在 Windows 主机上安装 Wircsrv IRC 服务器和 xchat IRC 客户端,构造了 NS-3 仿真节点和 xchat 加入 IRC 服务器的试验,试验结果如图 6 所示。

图 6(a)显示了 Wircsrv 当前 IRC 客户端的连接状况,其中_ xchat 为物理客户端,其它为仿真网络节点加入的客户端,如图6(a)显示的qscd用户对应的 IP 地址为 10.1.9.4,即仿真网络中节点 13 的 IP 地址。图 6(b)为 xchat 客户端显示界面,表明仿真客户



(a) Wircsrv IRC 客户端加入统计　　(b) xchat 客户端显示界面

图 6　仿真节点加入 **Wircsrv** 试验结果

端不仅可以加入物理主机的 IRC 服务器还可以和 xchat 处以同一频道,说明仿真客户端和物理客户端之间没有差异,即真正实现了虚实结合。

图 7 通过设定不同的链路延迟时间,统计仿真节点加入 IRC 服务器的时间。试验结果基本趋于一条直线,表明仿真节点加入 IRC 服务器的时间主要取决于 IRC 仿真节点客户端和物理 IRC 服务器对数据包的处理时间,而数据包在中间链路传输的时间相对较小,并且当链路延迟等值递增时,IRC 服务器的加入时间基本也处于等值变化中,说明仿真节点可以完成实时加入物理 IRC 服务器中。

2.3　路由计算效率测试

本节路由计算测试不同大小的网络拓扑和经过路由计算生成路由表的时间关系,并和 NS-3 的全局路由算法进行对比。由于 NS-3 的全局路由算法在对包含 csma 链路类型的拓扑进行计算时,会出现环

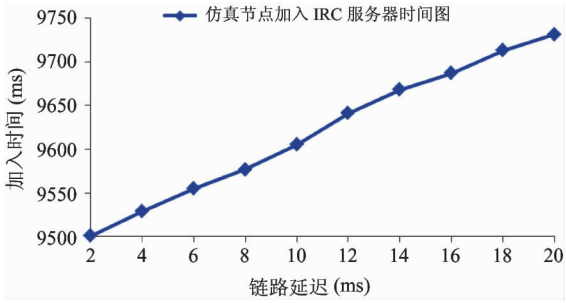


图 **7** 仿真节点加入 **IRC** 服务器时间图

路计算错误的问题,因此试验采取的拓扑全都为路由器节点,并使用点对点链路类型作为各个路由器节点之间的链路连接,试验结果如图 8 所示。

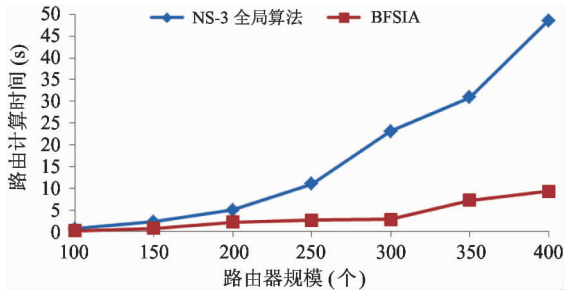


图 **8** 网络拓扑与路由计算时间关系图

路由计算时间指从读取拓扑到生成整个网络拓扑可配置路由表的时间。NS-3 全局算法和基于广度优先搜索的路由改进算法(BFSIA)的路由计算时间都随着网络拓扑规模的扩大呈递增趋势,并且 BFSIA 的效率一直优于 NS-3 全局算法,伴随着拓扑规模的扩大,时间差也会变得越来越大。

这是由于虽然两者的输入都为整个网络拓扑,NS-3 全局路由算法对整个网络拓扑中的任意两点都进行了路由计算,并且在每一个节点上都存储了整个拓扑的全局路由表,即每一个节点上的拓扑信息是相同的;而 BFSIA 根据仿真应用绑定在终端主机上的前提,只对终端路由器节点之间的路由进行了计算,并且每个节点上只存储和自己自身相关的路由信息,并没有进行整张路由表的存储,因此在计算和存储上都相对 NS-3 全局路由算法节省了时间。

3 结 论

本文通过对网络环境构建问题的研究,结合试验床和网络仿真器的优点,提出了一种新型的网络环境构建技术。用此技术可以解决传统试验床研究代价高、网络仿真真实性差的缺点。通过虚实通信试验验证了实现物理网路和仿真网络之间通信的可行性。在基于虚实通信的基础之上又进行了仿真节点加入 Wircsrv 服务器的试验,表明虚实结合可以完成具体网络案例的测试。最后进行了路由测试试验,验证了网络试验拓扑的可扩展性。

参考文献

[1] 张黎辉,段海新,戴世冬. DDoS 攻击防御试验床的设计与实现. 计算机工程,2008,34(13):118-120

[2] 全晓莉,周南权,余永辉. 基于虚拟仪器技术的网络实验系统的研究. 计算机工程与设计,2011,32(9):3227-3230

[3] 冯陈伟. 利用 VMware 构建虚拟网络平台. 信息系统工程,2009,8:78-81

[4] 张云,唐积强,闫健恩等. 基于 KVM 虚拟化的网络环境自动构建技术研究. 高技术通讯,2014,24(10): 1037-1043

[5] 袁晓,蔡志平,刘书昊等. 大规模网络仿真软件及其仿真技术. 计算机技术与发展,2014,(7): 9-12

[6] 刘东亮,马春光. 基于 NS-3 的机会网络路由协议仿真. 信息网络安全,2014,(5): 52-58

[7] 倪晓伟. 基于 NS-3 无线自组网多信道路由协议研究[硕士论文]. 北京邮电大学,2013

[8] 王华忠,姚俊,程华. 一种虚实结合温度控制远程实验系统的开发. 华东理工大学学报,2012,38(2):205-209

[9] 廖建,彭健,章乐平等. 一种虚实结合的联合试验系统及方法. 计算机测量与控制,2014,22(11): 3650-3653

Research on a technique of combining the virtual with the true for construction of network environments based on NS-3

Li Guangrong[*], Wang Kun[**], Zhang Zhaoxin[*]

([*]School of Computer, Harbin Institute of Technology, Harbin 150000)

([**]The National Computer Network Emergency Response Technical Team Coordination Center of China, Beijing 100029)

Abstract

The problem of building network environments was studied. In consideration of the shortcomings of low truthfulness in network simulation and high cost of the traditional testbed technique for net-environment building, a novel net-environment construction technique based on the network simulator NS-3 was proposed. To realize the linkage of the physical network and the simulation network (called "the linkage of the virtual with the true" for short) and the routing strategy for the two, the new technique uses the design of the linkage of the virtual and the true based on the virtual network card's bridging. To realize the data communication between the simulation network and the physical network, the virtual network card's bridging uses the polling monitor algorithm (PMA), and the routing strategy uses the breath first search improved algorithm (BFSIA). The constructed network was tested by using the ping command and the IRC botnet, and the test results show that the proposed technique can forward the real packet. The consumption time of BFSIA routing was tested by setting the different sizes of network topology, and the comparison with the NS-3 global routing algorithm proved the superiority of BFSIA.

Key words: network simulation, formalization of topology, virtual network card's bridging, routing algorithm