



面向开源路由设备的 I2RS 加速技术设计与实现

施仨, 刘德靖

(南通大学电子信息学院, 江苏 南通 226019)

摘要:南向协议接口标准 I2RS(路由系统接口)是传统网络向 SDN(软件定义网络)过渡过程中出现的一种新标准。基于该标准,提出了一种直接利用地址访问内核空间数据的方法,从而能够使南向接口以更高效率获取单点路由设备状态和各类统计信息的方法,给出了方法的实现过程,并对采用本方法提取数据的速度、访问数据出错次数以及从异常中恢复的时间进行了大量、长时间的测试。实验测试结果表明,与现有技术相比,本方法提取信息的速率明显提高,并且在提取信息出现异常时能够快速从异常中恢复。在实际网络环境中的应用表明,该方法发挥了快速可靠提取信息的作用。

关键词:软件定义网络;路由系统接口;南向接口;路由设备;操作系统

中图分类号:TP393

文献标识码:A

doi: 10.11959/j.issn.1000-0801.2016137

A quickly accessed method for I2RS in open-source router

SHI Quan, LIU Dejing

School of Electronic Information, Nantong University, Nantong 226019, China

Abstract: Southbound interface protocol specification I2RS (interface to routing system) appears in the transitional period of traditional network to SDN (software defined networking). Based on the protocol specification, a method which could get the state and statistics information of network's equipments in a more efficient way was put forward. The design was detailed and adequately testing was performed, which included the speed of getting information about equipments' state and statistics, the numbers of the wrong information as well as the recovery time from exception. The experiment results show that the proposed method improves the speed of getting the information about equipments' state and statistics, which is better than the existed method and when some exceptions happen, the proposed method can recover the situation quickly. The application of the real network environment shows that the proposed method can get the wanted information quickly and stably.

Key words: SDN, I2RS, south bound interface, routing equipment, operating system

收稿日期:2016-03-03;修回日期:2016-05-02

通信作者:施仨, sq@ntu.edu.cn

基金项目:国家自然科学基金资助项目(No.61171132);江苏省自然科学基金资助项目(No.BK20151272);江苏省“六大人才高峰”项目(No. 2014-WLW-029);江苏省“333 工程”资助项目(No.BRA2015356);南通市科技项目(No.CP2013001, No.HL2014006, No.BK2014054)

Foundation Items: The National Natural Science Foundation of China(No.61171132), The Natural Science Foundation of Jiangsu(No. BK20151272), The “333” Program Talents of Jiangsu Province(No.BRA2015356), The Six Top Talents of Jiangsu Province Grant(No. 2014-WLW-029), Nantong Science and Technology Platform (No.CP2013001, No.HL2014006, No.BK2014054)

1 引言

软件定义网络(software defined networking,SDN)的南向协议接口有多种标准^[1],比较典型的有 ForCES^[2]、OpenFlow^[3]等,而针对实际应用中非 SDN 设备数量仍占多数的情况,IETF 制定了另一套南向接口标准 I2RS(路由系统接口)^[4]。I2RS 的主要思想是复用现有路由设备,在现有路由设备上开发出一套能够获取其路由信息、状态信息和各种统计信息的监控接口,并根据监控接口获取的各种信息对各个路由设备的路由策略进行动态调整。在现有技术中,实现 I2RS 监测最常见的方法是在路由设备所安装的操作系统(以下简称路由操作系统)的用户空间实现一个代理,由代理收集路由操作系统内用户空间和内核空间的各种信息,并且将其转发给外部的 SDN 控制器。

在路由操作系统内,对于用户空间,代理一般通过共享内存的方式来获得各种信息;对于内核空间,代理获取信息的方法有多种,常见的有 Sysctl 方式^[5]、Mmap 方式^[6]、Proc 文件系统方式^[7]和 UNIX domain socket 方式^[8]。上述几种通信方式虽然都能在路由操作系统内完成内核空间和用户空间的数据交互工作,但不同通信方式适合的应用场景不同,开销也不同。例如,Sysctl 方式通常被设计用来供用户在用户空间动态调整一些内核可配置信息,典型应用场景是一些通信量较小的用户空间和内核空间的交互;Mmap 方式用来将一段连续的内存映射到用户空间,其只适合像数组一样的占用连续一大段内存的数据结构,但不能很好地支持链式存储结构,典型的应用场景是磁盘文件的读写;Proc 文件系统方式是一种类似于 Mmap 的方式,适合于读取内核的一些参数或者发送简单开关指令给内核模块;UNIX domain socket 方式则是一种全双工的内核和用户空间通信方式,适合于用户空间和内核空间大数据量数据的交互,然而传输的是无结构的信息,并且数据交互过程需要进行多次复制操作。

当前,在 SDN 实际应用中存在这样一种特殊需求的 I2RS 监测接口,需要大量地从路由操作系统内核空间中提取状态、统计等各类信息,并且这些信息是有结构的。此时,上述几种内核空间通信方式都不能很好地满足提取这种类型数据的要求。本文针对这种特殊需求的应用,提出了一种直接利用地址访问路由操作系统内核空间数据的方法,并对所提方法的效率以及顽健性进行了测试。下面给出具体分析与设计过程。

2 设计原理分析

路由操作系统有软件路由操作系统和硬件路由操作系统,例如,参考文献[9]提出了基于 NetBSD 改写的软件路由操作系统,参考文献[10]提出了基于 FPGA 技术的硬件路由操作系统,本文的路由操作系统是基于 FreeBSD 10.0 操作系统改写而成的软件路由操作系统,因此与 FreeBSD 10.0 操作系统相同,在操作系统的用户空间不能直接通过地址访问内核空间的数据。在 FreeBSD 10.0 操作系统中,设计这一策略的目的是限制用户程序的访问权限,避免用户程序对内核空间数据错误的修改而导致整个系统运行崩溃。虽然,用户空间不能直接利用地址访问内核空间数据,但操作系统也添加了多种用户空间和内核空间数据交互方式。然而,禁止直接用地址访问内核空间的数据,第1节所述方法就需要一些额外的开销供用户空间和内核空间进行数据通信,在本文中,代理只会读取信息,不会对内核空间的数据进行写操作。因此,从用户空间直接利用地址访问内核空间数据成为减小开销的一个安全可行方案。

在路由操作系统中,程序用虚拟地址访问数据,虚拟地址经过地址转换映射到物理地址,最终程序通过物理地址访问到实际的数据^[11]。而对用户空间或者内核空间地址的判断工作,就发生在虚拟地址转换成物理地址的过程中。转换过程中,MMU(内存管理单元)发现地址不合法,产生一个异常,在异常中,具体判断出现地址的不合法属于哪一种类型,通过修改内核源码,在对用户空间直接利用地址访问这一异常处理分支中,去掉其对内核空间地址的保护措施,最终能够从用户空间利用地址直接访问内核空间的数据。

3 设计与实现

本文设计的系统拓扑如图1所示,整个拓扑网络由 SDN 控制器和路由设备组成。SDN 控制器通过 I2RS 接口获取路由操作系统状态、统计等各类信息,图1中虚框部分为路由设备的内部结构示意图。一台路由设备主要由路由设备硬件和路由操作系统两部分组成,路由操作系统进一步可以分为用户空间和内核空间。代理就运行在路由操作系统的用户空间中,而需要获取的路由设备的大部分状态、统计等各类信息则存在于内核空间。

现有技术中路由操作系统内用户空间和内核空间通



在实际路由操作系统中,各种状态、统计等信息零星分布在内存的不同功能模块内,而不是集中在图1所示的数据模块中,并且各条记录的地址会动态变化。例如,反应路由接口与反应路由表信息的数据在内存中存放的地址是不同的。路由表添加了一条记录,新记录会被分配到一个新的存储空间中,其拥有了一个新的内存地址。因此,为了方便代理利用地址直接访问,能正确获取内核收集来的各种信息,本设计在内核空间中添加了一个线程,用来定时收集存储在内核空间零散分布的状态、统计等信息,并且设计了一套数据结构以规范内核空间线程收集的数据存储,使其结构化。同时,为了保证代理访问内核空间数据的可靠性,设计了一套用户空间和内核空间的访问同步机制以及访问超时检查机制以确保所设计方案的可靠性和实用性。

通过对内核空间中需要进行收集的信息进行详细分析,内核空间中的存储结构可以用模型—实例的模式进行

由于接口模型中有接口名称、接口转发分组数量两个字段,而静态路由模型有静态路由命中次数 1 个字段,因此两个模型实例的数据结构是不相同的。为了能够以一种统一的方式管理各种模型实例,采用了图 2 所示的数据结构,以达到不同的数据结构用同一存储结构进行管理的目的。在图 2 的结构中,存储每一个模型实例的数据时,每个实例并不是在实例内部存放统计信息数据,而是通过指针指向一块单独的内存区域。这样的存储结构,对于每一个模型,

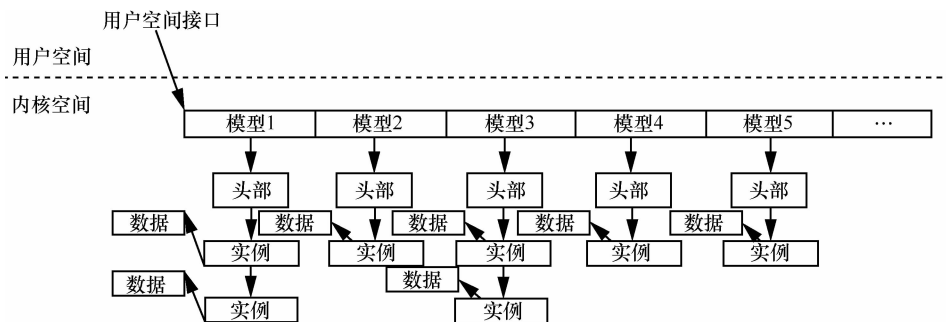


图 2 内核存储结构

其模型和实例部分的结构是完全相同的,不同的仅是其数据部分。因此,在这套存储结构上寻找模型及实例可以用相同的方法,不同的仅仅是取每一个模型数据的方法。

3.2 读写同步机制设计

路由操作系统用户空间进程通过存储结构首地址去获取内核空间各模型的数据,内核空间线程通过存储结构首地址对各模型数据进行更新。然而,由于路由设备内操作系统进程调度以及处理器多核等原因,对同一块区域的读写,会出现数据不一致的情况。为了保证数据的一致性,必须对路由操作系统内用户空间的读进程和内核空间的写线程进行同步操作。本文所面对的同步问题,并不是路由操作系统中用户空间的临界资源同步读写,或者内核空间临界资源同步读写,上述两种同步方案在路由操作系统中已经有实现的接口。本文涉及的临界资源同步是关于用户空间进程和内核空间线程的同步,由于原路由操作系统不提供在用户空间直接用地址访问内核空间数据的方法,即用户空间进程不能从用户空间访问内核空间的数据,造成现有路由操作系统不提供用户空间和内核空间的同步方法。为了实现用户空间进程和内核空间线程的同步操作来保证数据的一致性,本文设计了如图 3 所示的同步访问方案。

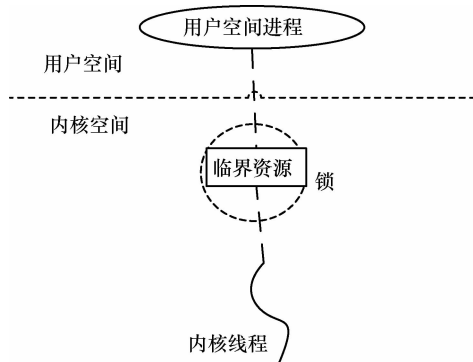


图3 用户进程及内核线程的同步

图3所示的临界资源即各模型实例的统计数据,为了能够对访问临界资源的用户空间进程和内核空间线程进行同步操作,本设计给位于内核空间的临界资源加了一把锁。当用户空间进程准备读取临界资源时,该进程首先触发内核的一段程序执行,该段被触发的程序给临界资源加上锁并返回一个操作结果。若返回的操作结果表示获取锁成功,则用户空间进程开始读取临界资源中的信息;否则等待,直到获取锁进行读取数据。内核空间线程的写操作与用户空间的进程读操作类似。

3.3 读写机制超时检查设计

正确工作状态下,用户空间进程和内核空间线程的工作流程是:首先,用户空间进程每隔 5 s 申请一次读锁,申请成功后读取临界区的数据并释放读锁,然后内核空间线程每隔 5 s 申请一次写锁,申请成功后更新数据并释放写锁。然而,在实际的运行测试中,出现了两种特殊情况:一是用户空间某个进程异常退出,没有释放读锁,导致内核空间线程长时间不能获取写锁,如图 4 所示;二是用户空间有太多的进程读取临界区数据,导致内核空间长时间不能获取写锁,其示意如图 5 所示。

上述两种情况无论哪一种发生,都会导致内核空间线程不能及时获取写锁,从而不能更新临界区的数据。为了

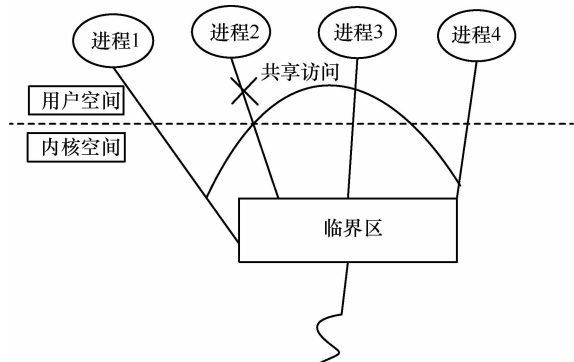


图 4 用户进程异常终止

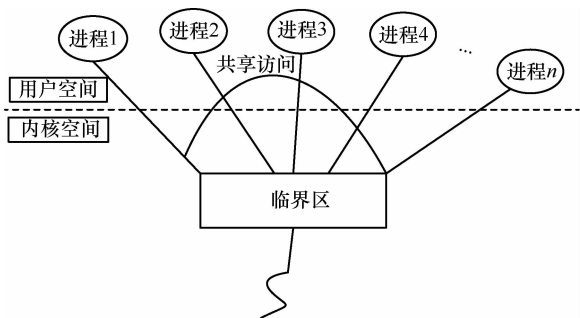


图5 多进程同时访问

解决上述的异常问题,本文在内核空间中设计了超时检查功能。超时检查功能的主要思想是设计一个计数器,当内核空间线程获取写锁失败时,计数器开始计时,在计数器计时开始后的一段固定时间内,如果内核空间线程一直不能获取写锁,则说明用户空间访问的进程发生异常,需要在内核空间强制释放所有读锁。在上述强制释放读锁的过程中,很有可能释放掉用户空间正常进程的读锁,若此时内核空间线程又立刻获取写锁进行更新数据,则仍然存在着读写冲突问题。为了解决这个问题,本文为锁设置了一个特殊状态:“等待态”。在“等待态”下,有两个特点,一是内核空间线程不再每隔 5 s 获取一次写锁,而是不断尝试获取写锁,直到获取写锁为止;二是内核空间线程能够在读锁全部释放的情况下,获取写锁,但是用户空间进程申请读锁会失败。在引入“等待态”后,内核线程更新数据时间应该是:

更新数据时间 = 5 s (内核设置的更新时间) + 尝试获取锁的时间 (1)

其总体工作流程如图 6 所示。

在图 6 中,内核空间线程需要对临界资源的信息进行更新,它首先申请写锁,若此时上述两种异常情况的一种发生或两者同时发生,内核空间线程申请写锁时会失败。在上述异常情况下,会设置锁的状态为“等待态”,并且计算距离第一次发生异常所经过的时间,如果时间超过了指定值(这个值与具体应用有关,本文目前设置为 15 s),则内核空间的写线程会强制释放所有的读锁。释放完之后,内核开始申请写锁,由于此时锁的状态为“等待态”,并且读锁的计数已经恢复到 0,所以写锁申请成功,然后更新数据并设置锁的状态为正常态,释放写锁。

4 方案运行测试

本文从时间提取的数据发生错误次数、异常状况下恢

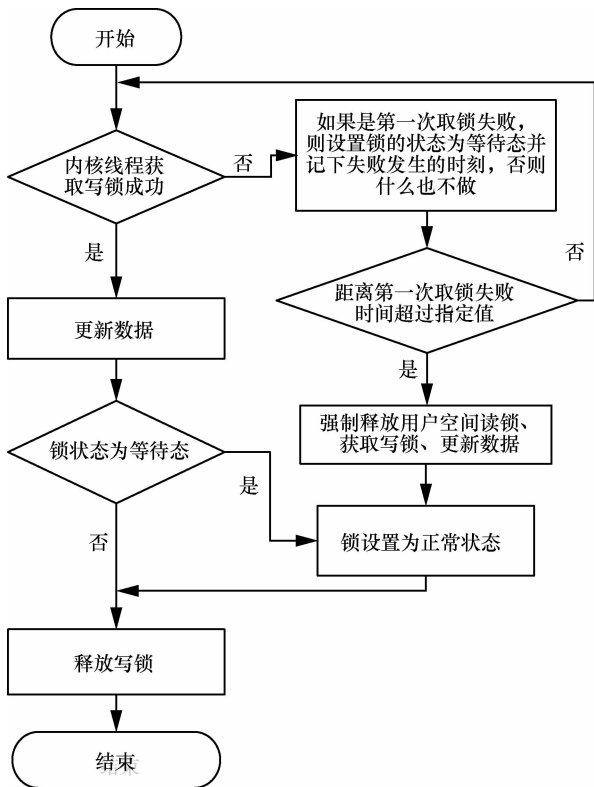


图6 超时检查机制

复时间等角度对本设计方法的性能及可靠性进行了评估。使用的测试平台为 AMD64 3.7 GHz 处理器、内存 4 GB 的通用型计算机,路由操作系统为基于 FreeBSD 10.0 内核修改的测试系统。

4.1 运行时间测试

本文提到多种内核空间与用户空间通信方式,例如 Sysctl、Mmap、Proc、UNIX domain socket 等。在设计测试方案时,由于 Mmap 与 Proc 方式适用于从内核空间与用户空间传递一整块信息,而本文所传输的信息是链式的,因此 Mmap 与 Proc 不适合本文所设计的方案。Sysctl 能够用来将一小段内存传递到用户空间,对于本文的链式方案也不适用。UNIX domain socket 是一种常用的内核空间和用户空间通信的方式,能够将内核中的链式信息传递到用户空间,因此在系统测试时,将本文的设计方案与 UNIX domain socket 通信方案进行对比。图 7 给出了获取相同的内核空间信息时,两种程序所需要的时间。图 7 的测试结果表明,直接用地址访问方式获取数据,其获取内核静态路由统计信息的时间比用 UNIX domain socket 方式少,说明本方案在提取内核空间信息效率方面比 UNIX domain socket 方式要高。

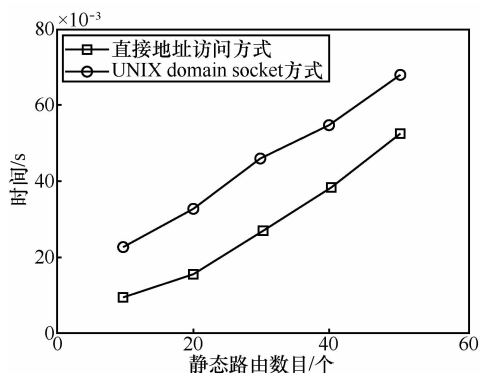


图7 获取数据时间测试比较曲线

4.2 读写问题验证

(1) 同步方案有效性的验证

路由操作系统中,由于进程调度以及多核等原因,访问临界区进程越多,访问数据越容易出现数据不一致问题(即本文所说的错误数据),图8给出了在未使用同步方案的情况下,随着时间的变化,不同数目进程访问临界区时,所产生的错误次数变化;而图9则给出了在使用不同方案的情况下,产生错误次数的变化。

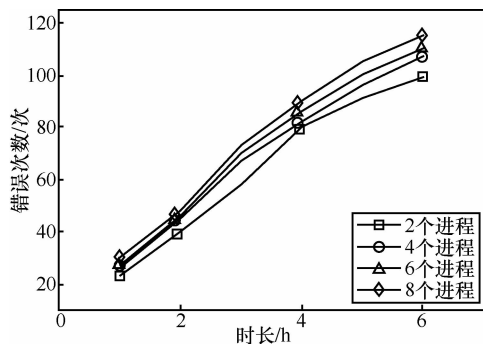


图8 未使用同步方案时的数据出错次数

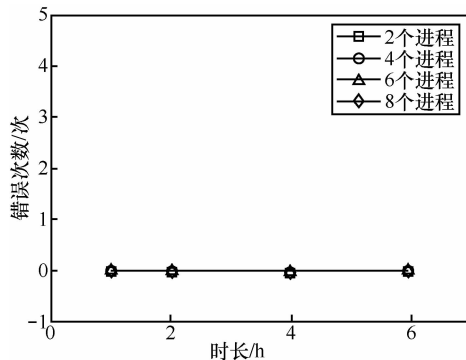


图9 使用同步方案时的数据出错次数

在图8中,由于未使用同步方案,可以看出,不管同时测试的进程数目有多少,错误次数都会随着时间的变化不

断增多;而随着同时运行的进程数目的增多,其数据错误也会明显增多,说明并发进程数目越多,的确会增加数据的不一致问题。图9中使用同步方案后,相比于图8可以看出,随着时间的变化,错误的次数始终为0;即使并发进程数目增多,其次数仍然为0。因此,说明了尽管在进程增多不确定性增大的情况下,本文同步方案都能够解决数据不一致的问题,证明了同步方案的有效性。

(2) 超时检查机制有效性验证

为了验证本文设计的读写机制超时检查方案是否能够保证数据的及时更新,设计了图4和图5出现的两种异常情况。

首先,模拟图4所示的用户进程异常终止情况,在用户空间同时开启5个进程,此时内核空间锁的计数为5,人为终止其中任意一个进程,若没有超时检查,内核空间锁的计数应该始终大于或等于1;引入超时检查功能后,内核空间锁的计数能够回归到0,说明超时检查机制起了作用。

其次,模拟图5所示情况,表1给出了在进程数为10的情况下,数据的平均更新时间与内核设置的更新时间之间的关系,从表1中可以看出,随着内核设置的更新时间的变化,数据的平均更新时间都是等于“内核设置的更新时间+比较短的时间”,与第3.3节分析的“更新数据时间=5s(内核设置的更新时间)+尝试获取锁的时间”一致。表2给出了随着进程数目的变化,内核空间数据在使用与不使用同步情况下的平均更新时间。结果表明,采用超时检查方案后,进程数目从5个调整到20个,其数据平均更新时间为5.02s,而未使用超时检查的程序时,在5个进程时,时间也为5.02s,分析其原因,是由于5个用户空间进程不是很多,在5s时间段内能够访问完所有数据,不会影响到内核空间线程获取写锁,而从开启到10个线程时,其平均更新时间不确定,这里不确定的意思是指更新时间没规律且跳跃性很大,例如会呈现5s、40s、15s这样变化很大的时间,分析其原因,由于没有超时检查机制,其与用户空间进程是通过竞争关系获取锁,随着用户空间进程数目的增多,虽然它获取锁的概率越来越小,但是也有获得到锁的概率,什么时候能够成功获取锁,是一个随机事件,所以其时间呈现这种跳跃性的变化。

表1、表2的结果相互对比说明,使用超时检查机制后,能够使更新时间维持在一个稳定的时间段内,证明了超时检查机制所设计的有效性。



表 1 数据平均更新时间与内核更新间隔时间关系

内核设置的时间/s	使用超时检查机制时的数据平均更新时间/s
5	5.03
10	10.06
15	15.03
20	20.01
25	25.03

表 2 数据平均更新时间与进程数的关系

进程数/个	使用超时检查机制时的数据平均更新时间/s	未使用超时检查机制时的数据平均更新时间/s
5	5.03	5.02
10	5.19	不确定
15	5.25	不确定
20	5.30	不确定
25	5.28	不确定

5 结束语

本文针对 SDN 控制器需要提取路由设备统计信息这一要求,通过对 I2RS 接口的研究,设计了一套利用地址直接获取内核空间信息的方法。与 UNIX domain socket 方法相比,本设计在提取相同数量的内核信息时所需的时间更少。本文还设计了一种用户空间和内核空间访问临界资源的同步方案,实现了对临界资源的可靠访问,在实际应用中发挥了稳定提取信息的作用。然而,在 SDN 南向接口获取数据的过程中,影响其获取信息效率的因素往往不只是一个,本文所设计的方案只是针对控制器需要提取路由设备信息时,优化了数据从内核空间到用户空间的路径,但是由路由设备发往控制器这条路径并没有得到优化,因此,在这条路径上的优化尝试将是进一步研究的内容。

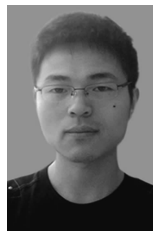
参考文献:

- [1] Stanford university clean slate program [EB/OL]. [2015-02-12]. <http://clean.slate.stanford.edu/>.
- [2] IETF. Forwarding and control element separation (ForCES) forwarding element model: RFC5812[S]. 2010.
- [3] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks [J]. ACM Communications Review, 2008, 38(2): 69-74.
- [4] HARES S, WHITE R. Software-defined networks and the interface to the routing system (I2RS) [J]. IEEE Internet Computing, 2013, 17(4): 84-88.
- [5] Sysctl [EB/OL]. [2016-01-11]. <http://www.freebsd.org/cgi/man.cgi?query=sysctl&sektion=3>.
- [6] IEEE standard for information technology-portable operating system interface (POSIX(R)) [EB/OL]. [2016-01-11]. <http://standards.ieee.org/findstds/standard/1003.1-2008.html>2001.
- [7] Proc file system [EB/OL]. [2016-01-11]. <https://www.kernel.org/doc/Documentation/filesystems/proc.txt>.
- [8] GREG A. Woods [EB/OL]. [2016-01-11]. <http://archives.neohapsis.com/archives/postfix/2000-09/1476.html>.
- [9] 徐恪, 吴建平, 江勇, 等. 通用路由器软件体系结构研究综述[J]. 小型微型计算机系统, 2001, 22(4): 385-392.
XU K, WU J P, JIANG Y, et al. Research on software architecture of general-purpose routers: a survey [J]. Mini-micro Systems, 2001, 22(4): 385-392.
- [10] 钟雪燕. 基于 FPGA 的 SpaceWire 路由器的设计与验证[D]. 南京: 南京航空航天大学, 2013.
ZHONG X Y. Design and verification of SpaceWire router based on the FPGA [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2013.
- [11] JACOB B, MUDGE T. Software-managed address translation[C]// The 3rd International Symposium on High Performance Computer Architecture, Feb 1-5, 1997, San Antonio, Texas, USA. New Jersey: IEEE Press, 1997: 156-167.

[作者简介]



施俊(1973-),男,博士,南通大学电子信息学院教授,主要研究方向为智能信息处理。



刘德靖(1990-),男,南通大学电子信息学院硕士生,主要研究方向为智能信息处理、计算机网络。