



基于 SDN 的链路故障恢复

侯 乐,汪 硕,林毅凯,姚博韬,刘 江

(北京邮电大学 北京 100876)

摘 要:网络的灾备与故障恢复作为网络安全的一部分,一直以来是人们关注的重点,传统网络缺少自动化的网络故障检测与恢复工具。SDN 打破了传统网络设计理念,实现了控制平面和数据平面的分离,并开放了网络的可编程能力,从而提高了网络的灵活性和可管控性。通过 SDN 技术,自动化的网络故障恢复与检测成为了可能。设计了一种基于 SDN 的链路故障检测与自动化恢复方案,通过链路发现协议来获得全网拓扑,监控网络链路状态,并通过 SDN 的集中管控特性和可编程性实现链路故障快速检测和故障恢复。

关键词:软件定义网络;网络安全;链路故障恢复

doi: 10.11959/j.issn.1000-0801.2015158

Link Failure Recovery Based on SDN

Hou Le, Wang Shuo, Lin Yikai, Yao Botao, Liu Jiang

(Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: As part of network security, network disaster backup and recovery has been a focus, and traditional network lacks of automation failure detection and recovery tool. SDN broke the traditional network design idea, the separation of control plane and data plane was realized, and the network of programmable ability was opened. Based on SDN, automatic network fault recovery and detection has become a possibility. A link failure detection and automatic recovery program based on SDN was designed. The entire network topology could be obtained, and network link state could be monitored through link discovery protocol. Through the centralized management and control features of SDN, fast detection and link failure recovery were achieved.

Key words: software defined networking, network security, link failure recovery

1 引言

随着互联网规模爆炸式的增长,网络底层的交换设备已经发展到了上百台的规模。同时网络的业务也变得复杂起来,网络中出现了各种复杂的协议和网络管理策略。从而,调试网络上的故障变成了一项困难的事情,人为操作失误、网络协议因素或者自然因素等都是引起网络故障的

因素^[1]。保障网络的安全与稳定是保障网络正常运行的重要一环,为了检测网络中的故障,网络维护人员常常需要使用一些检测命令与工具,但随着网络设备的不断增加,依靠人工使用工具去检测并恢复网络故障变得越来越低效。鉴于网络需要快速从故障中恢复来保证业务的安全,人们需要一个更加成熟的工具和方法来检测网络中的故障,这个工具可以自动地进行故障检测以及排除^[2]。

收稿日期:2015-01-20;修回日期:2015-06-12

论文引用格式:侯乐,汪硕,林毅凯等.基于 SDN 的链路故障恢复. 电信科学,2015158

Hou L, Wang S, Lin Y K, et al. Link failure recovery based on SDN. Telecommunications Science, 2015158



软件定义网络 (software defined networking, SDN) 是一种数据控制分离、软件可编程的新型网络体系架构, 具有全局控制视图。SDN 可以明确知道网络管理者的意图, 因为策略是通过运行在控制器上的程序下发的。例如交换机对数据分组过滤后产生的分组丢失, 在传统网络中很难判断是管理员下发的安全策略还是网络故障。但是在 SDN 中规则是由程序下发的, 只需要检查转发行为是否匹配程序就可以明确知道是否发生故障。其次, 由于 SDN 控制器集中管控了所有的底层交换机, 控制器可以即时收集到所有底层交换机的转发状态。而在传统网络中, 很难有效地收集到分散在各种交换机的转发信息, 不能有效地进行自动化的错误检测与恢复。可见, SDN 数据分离、集中管控、软件可编程的特点使得自动化的故障恢复成为了可能。

现代化的数据中心或者大规模的园区网等网络中, 由于需要高可靠性的网络保障, 在设计的时候就使用了多条冗余链路来保证网络的可靠性。大型网络每个设备间都存在多条链路, 并且本身的网络节点繁多, 这就造成了网络拓扑极其复杂。当网络中发生某条链路故障或者中断时, 网络管理员需要花费大量的时间来查找出错误链路的位置。当找到故障链路之后, 还需要花费精力来更改路由或者转发表项的配置来保证交换机能够正确使用备用链路转发数据。可见, 一旦发生网络故障, 必将会引起长时间的网路中断, 影响用户的正常使用。利用以上提到的 SDN 的优势, 可以有效地解决链路故障问题, 实现快速定位故障链路, 自动配置路由转发规则并切换到备用链路上。

2 SDN 与 OpenFlow 协议

SDN 是一种数据控制分离、软件可编程的新型网络创新架构, 采用了集中式的控制平面和分布式的转发平面, 两个平面相互分离, 控制平面利用控制—转发通信接口对转发平面上的网络设备进行集中式控制, 并提供灵活的可编程能力, 具备以上特点的网络架构都可以被认为是一种广义的 SDN。

OpenFlow 协议诞生于美国斯坦福大学 Clean Slate 计划资助的 Ethane 项目。它是基于网络中“流”的概念设计的一种 SDN 南向接口协议, 通过提取网络通信产生数据分组的共同特征, 把它们抽象成一个个“流”, 使网络设备统一看待这些数据分组。经过多年的发展, OpenFlow 目前已经成为 SDN 的主流南向接口协议之一^[3,4]。

SDN 的核心思想是数据与控制分离, 在 OpenFlow 引入了“流”的概念后, 控制器可以根据某次通信中“流”的第一个数据分组的特征, 使用 OpenFlow 协议提供的接口对数据平面设备部署策略——OpenFlow 称之为流表, 而这次通信的后续流量则按照相应流表在硬件层次上进行匹配、转发, 从而实现了灵活的网络转发平面策略, 网络设备不再受固定协议的约束, 即软件与硬件“解耦”, 体现了 SDN 控制与数据分离的核心思想^[5]。

OpenFlow V1.0 架构原理如图 1 所示。

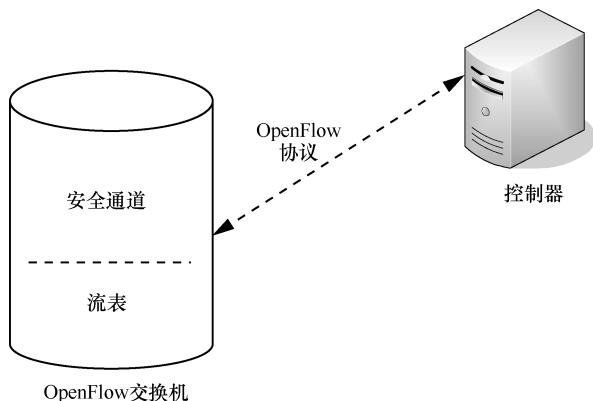


图 1 OpenFlow V1.0 架构原理示意

3 SDN 链路故障恢复机制

3.1 链路发现

网络链路拓扑发现是网络管理系统的核心与基础, 在传统网络中链路发现可以由各网络设备自动进行。与传统网络不同的是, 在 SDN 构架下是由控制器负责链路发现的功能。SDN 控制器主要先后使用了 LLDP (link layer discovery protocol, 链路层发现协议)、广播报文以及 packet-in 消息与 packet-out 消息实现链路发现功能。

LLDP 是一个二层协议, 它提供了一种标准的链路发现方式, 将本端设备的主要能力、管理地址、设备标识、接口标识等信息组织成不同的 TLV (type/length/value, 类型/长度/值), 并以 Ethernet II 格式或 SNAP 格式封装在 LLDPDU (link layer discovery protocol data unit, 链路层发现协议数据单元) 中, 并将封装好的帧发送给自己直连的其他设备, 这些设备收到这些信息后, 将其以标准 MIB (management information base, 管理信息库) 的形式保存起来, 以供网络管理系统查询机判断链路的通信状况^[6-9]。

SDN 链路发现过程如图 2 所示。

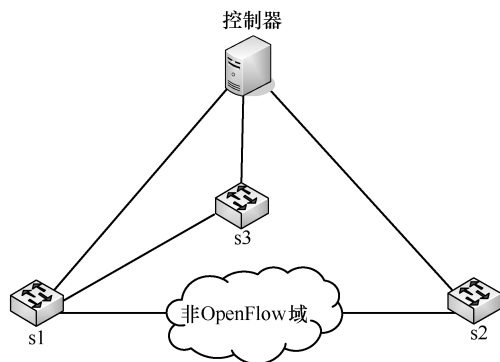


图2 SDN 链路发现过程

当控制器在执行链路发现过程时,首先会向所有与控制器直接相连接的 OpenFlow 交换机发送带有 LLDP 信息的 packet-out 分组,该消息命令交换机将 LLDP 数据分组发送给所有端口,如果这个 OpenFlow 交换机相邻的设备仍然是 OpenFlow 交换机,由于交换机中没有可以匹配的流表项,就会通过一个 packet-in 消息将此数据分组发送给控制器。通过这一步骤,控制器就可以得到所有两两直连的 OpenFlow 交换机拓扑信息。例如图 2 中,经过这一步后,控制器可以知道 s1 与 s3 是直接相连的。此时控制器只能认为 s1 和 s2 以及 s2 和 s3 之间不是直接相连的。并不知道它们之间是否有非 OpenFlow 交换机。

如果经过上面一步,控制器发现自己所控制的交换机并不是两两互相连通的,就会向所有被其控制的交换机下发命令,要求交换机发出广播分组并发送除交换机和控制器相连的端口之外的其他端口。广播分组从 OpenFlow 交换机发出后,如果网络中存在非 OpenFlow 域,广播分组将从这个网络域的一端进入并穿越,达到与该非 OpenFlow 域连接的其他 OpenFlow 交换机。因为在接收到广播分组的 OpenFlow 交换机中并没有对应的流表项可供广播分组匹配,所以该广播分组将被上传到控制器,从而告知控制器在网络中存在非 OpenFlow 域。如果控制器没有收到 OpenFlow 交换机上传的广播分组,则相应两个 OpenFlow 交换机属于断开关系。例如图 2 中,经过这一步骤,控制器就可以知道 s1 与 s2 直接存在非 OpenFlow 域,而 s2 和 s3 之间是断开的关系。通过上面的步骤,控制器实现了链路发现的过程。

3.2 链路故障检测

链路故障出现的时候会引起某段链路上出现大量的分组丢失或者数据分组无法传送,造成网络的传输速率下降,影响网络的正常使用。造成链路故障的原因多种多样,

但主要包括端口故障、流表条目错误和链路负载过重引起的大量分组丢失或者链路出现断路 3 种。

SDN 控制器可以通过不同的手段有效地检测到这 3 种不同原因引起的链路故障。首先,当交换机转发端口发生故障的时候,端口状态会发生变化,例如一个端口的状态从 up 变为 down 时,会触发 port-status 消息报告端口的信息以及此时发生的状态变化给控制器。此时控制器根据 port-status 消息可以判断某个转发端口出现了故障。其次,交换机拥有对所有交换机的转发流表,同时也能检测到所有交换机的转发行为。交换机通过对下发的流表进行检测、分析,可以判断出某段链路是否存在流表下发错误,同时也能发现某段链路是否存在拥塞。当出现流表逻辑错误引发的链路故障时,控制器可以快速修复流表的错误解决故障。最后,对于链路断路的情况,控制器需要定时发送 LLDP 分组的 packet-out 消息并根据反馈回来的 packet-in 消息来不断监测交换机的状态和更新网络拓扑。这样控制器中就会形成整个网络的全局资源视图,以实现控制器的集中管控功能^[7]。这是 SDN 与传统网络的区别之一,利用集中管控这一优势,就更容易地获得优化算法结果,从而达到监测和管理网络链路故障的目的。

3.3 链路故障恢复

在 SDN 中已经存在许多工具用来检测由转发表项错误引起的链路故障^[8-14]。所以本文主要解决由于链路断路或者转发端口问题而引起的链路故障,在网络设计或者组网的时候,往往会设计多条备用的冗余链路来应对链路故障,可以有效地利用冗余链路,使网络快速从故障中恢复出来。当控制器发现链路故障并定位到故障链路之后,就会删除交换机中旧的转发规则,并下发新的规则到交换机中保障数据分组能够正常转发。链路故障恢复如图 3 所示。

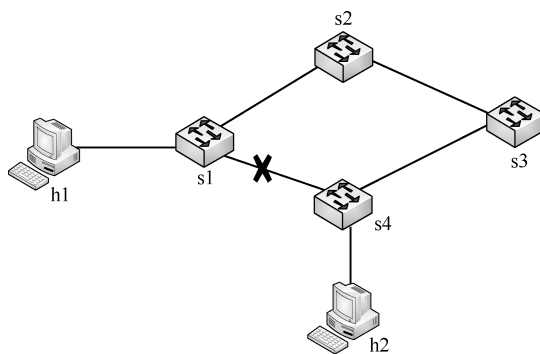


图3 链路故障恢复



例如图 3 中,为了更加直观,省略了控制器以及一些其他设备,在链路正常时,h1 与 h2 通信只需要通过 s1-s4 这条链路,当这条链路出现故障时,控制器检测到故障,向相应的交换机下发新的匹配流表项,这样 h1 与 h2 相互通信时可以通过 s1-s2-s3-s4 这条链路,使得通信流量正常转发,从而实现了链路故障恢复的目的。

现实中网络的拓扑可能很复杂,如果计算并重新写入受损路径上所有的交换机的转发表会带来巨大的开销。所以在新链路的计算时,只考虑受影响的两台交换机,如 s1 和 s4 之间的链路,只要修正了受影响交换机之间的链路,就可以保证所有的数据分组都能够正常地转发。一种可行的方法是使用 Dijkstra 算法计算出受影响交换机之间的一条最短路径并下发路由。

当计算好路由后,控制器要先收集到受影响的交换机中的转发表,然后根据计算出的新的路径,改写转发表中对应的转发端口,最后把新的转发表项下发到受影响的交换机中。通过以上的过程就可以保证所有的数据分组都能通过备用链路转发,使网络能够从故障中自动恢复,保证用户的正常使用。

4 链路故障恢复仿真验证

4.1 Mininet 下基于 Floodlight 的网络拓扑

Floodlight 是一款基于 Java 语言的开源 SDN 控制器,遵循 Apache 2.0 软件许可,支持 OpenFlow 协议。Floodlight 的日常开发与维护工作主要由开源社区支持,其中包括很多来自 Big Switch Networks 公司的工程师,同时 Floodlight 也是该公司商业控制器产品的核心部分。Floodlight 作为免费的开源控制器提供了与商业版本相同的 API,使得开发者可以把 Floodlight 上的程序快速移植到商用版本的控制器上^[67]。

在 Floodlight 控制器中,有一个 LinkDiscovery 模块可以感知到当前网络交换机之间的连接关系。其原理是通过定时地向交换机下发 LLDP 报文,根据交换机上传的 packet_in 报文来获知当前网络中交换机之间的连接情况^[15]。监听这一模块的更新消息,程序可以迅速地获知网络中的链路开启和关闭的情况。在实际实验中,程序就是通过获知备用路由被启用了之后,才在备用路由上下发相应的流表。在 Mininet 仿真环境下,建立网络拓扑图,如图 4 所示。

在图 4 中,服务器与交换机 s4 相连接,两台客户端与交换机 s1 相连接,链路 1 (s3-s4-s1) 为主用链路,链路 2

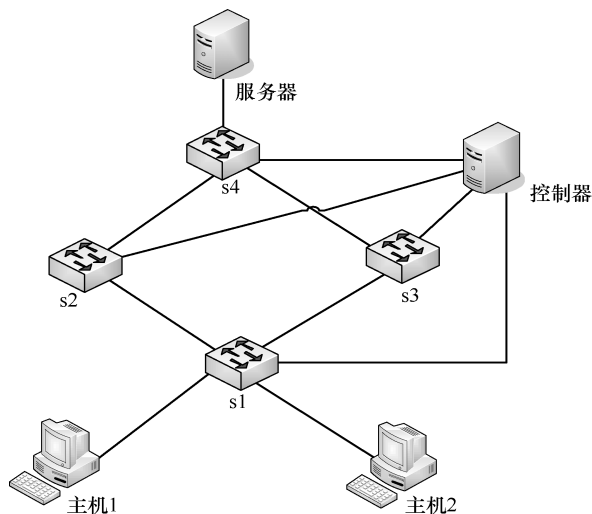


图 4 网络拓扑

(s3-s2-s1) 为备用链路,服务器通过主用链路由 2 台客户端提供视频流服务,而控制器实时监控网络链路状态,当主动断开主用链路时可以向备用链路相关交换机下发流表,实现流量的正常转发。

4.2 视频流下链路故障恢复仿真

使用基于 RTP 与 RTSP 的 VLC player 在多链路的 SDN 环境中对客户机进行多播服务,实现基于用户优先级的备用路由服务,并用可视化界面显示。

当用户向 VLC 服务器请求多播视频服务时,会进行 RTSP 握手以建立连接,其中报文记录了视频的端口和音频的端口。控制器将传来的数据分组里的内容解析成字符串,提取出服务器和用户约定好的端口的信息,这样就可以对用户加以区分。这样,当使用主链路进行通信时,两台客户机都可以正常地观看到视频。当主链路故障断开后,通过控制器就可以启用备用链路给对户优先级高的 VIP 用户,从而实现了 VIP 用户的链路恢复。而对普通的用户不做处理,这样普通用户就无法继续观看视频。仿真界面如图 5 所示。

在此仿真中,链路 1 (s3-s4-s1) 为主用链路,链路 2 (s3-s2-s1) 为备用链路,设置左边用户为普通用户,右边用户为 VIP 用户。当链路正常时,开启 VLC 程序接收 RTSP 视频流服务进行实时视频传输,此时 2 个用户均能观看视频。

断开主用链路,并对 VIP 用户启用备用链路,普通用户不做处理,此时,可以看到,VIP 用户可以继续观看视频,而普通用户无法继续观看视频。结果界面如图 6 所示。

根据上述实例,通过标记 VIP 与普通用户以及控制器

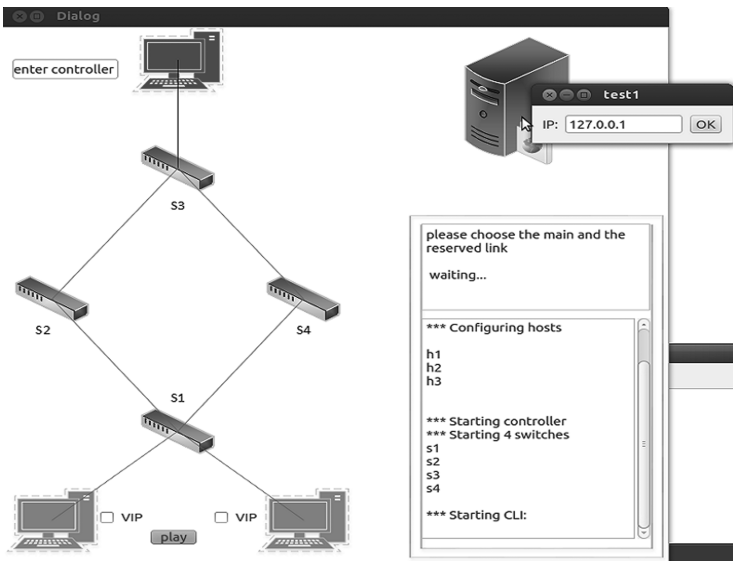


图 5 仿真界面

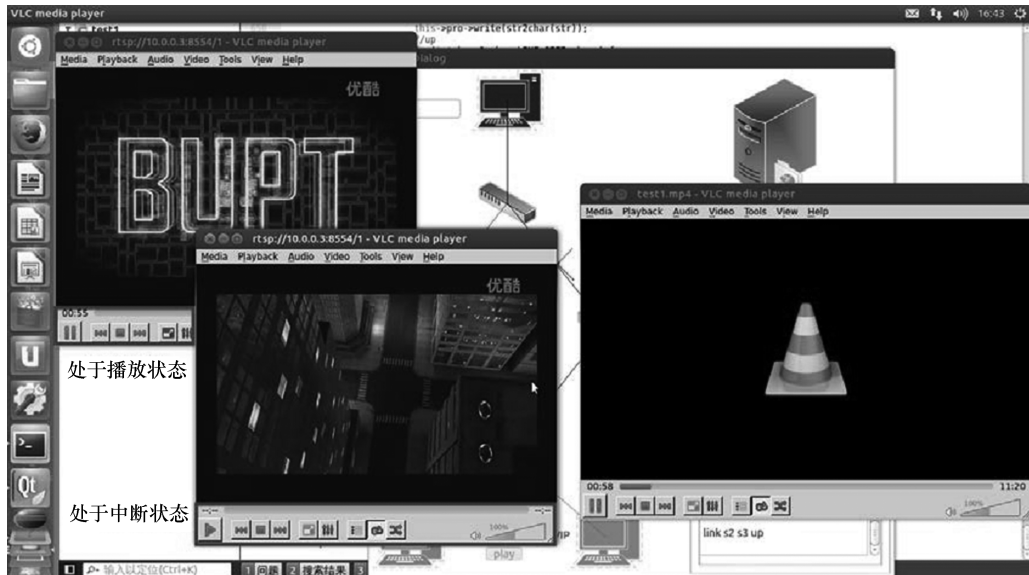


图 6 结果界面

的作用,在主用链路中断时,VIP 用户仍然可以继续观看视频,实现了 VIP 用户的链路故障恢复。可见,在 SDN 中拥有冗余链路时,可以有效地对链路故障进行检测与恢复。

5 结束语

本方案实现了在有冗余链路的情况下,对不同优先级的用户差异化进行链路故障的恢复,具有一定的实用价值。但其中也有很多不足之处,比如网络拓扑简单,没有考虑更为复杂的拓扑环境等。在链路恢复的时候使用了最少条数的链路作为冗余链路而没有考虑网络的负载情况,在

以后的工作中可以把网络的负载情况考虑进去,选择一条时延最小的链路作为备用链路。

综上所述,链路故障恢复涉及网络的可靠性,相比传统网络架构,SDN 架构在链路故障检测与恢复方面存在优势,在 SDN 下进行链路的检测与管理、链路故障的恢复很有必要,也很有前景。

参考文献

- 1 Zeng H Y, Kazemian P, Varghese G, *et al.* Automatic test packet generation. Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies(CoNEXT),Nice, France, 2012



- 2 Heller B, Scott C, McKeown B, *et al.* Leveraging SDN layering to systematically troubleshoot networks. Proceedings of HotSDN 2013, Hong Kong, China, 2013
- 3 ONF. Software-Defined Networking: the New Norm for Networks. ONF White Paper, 2012
- 4 ONF. OpenFlow Switch Specification Version 1.3.1, 2012
- 5 McKeown N, Anderson T, Balakrishnan H, *et al.* OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2)
- 6 雷葆华, 王峰, 王茜. SDN 核心技术剖析和实战指南. 北京: 电子工业出版社, 2013
Lei B H, Wang F, Wang X. SDN Core Technology Analysis and Actual Combat Manual. Beijing: Electronic Industry Press, 2013
- 7 黄韬, 刘江, 魏亮等. 软件定义网络核心原理与应用实践. 北京: 人民邮电出版社, 2014
Huang T, Liu J, Wei L, *et al.* The Core Principle and Application of SDN. Beijing: Posts & Telecom Press, 2014
- 8 Heller B, Handigol N, Jeyakumar V, *et al.* Where is the debugger for my software-defined network. Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN), Helsinki, Finland, August 2012
- 9 Kazemian P, Varghese G, McKeown N. Header space analysis: static checking for networks. Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2012), San Jose, CA, USA, 2012
- 10 Khurshid A, Zou X, Zhou W, *et al.* VeriFlow: verifying network-wide invariants in real time. Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2013), Lombard, IL, USA, 2013
- 11 Kuzniar M, Peresini P, Canini M, *et al.* A soft way for OpenFlow switch interoperability testing. Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT), Nice, France, 2012
- 12 Mai H H, Khurshid A, Agarwal R, *et al.* Debugging the data plane with anteater. Proceedings of the ACM SIGCOMM 2011 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Toronto, ON, Canada, 2011
- 13 Scott C, Wundsam A, Whitlock S, *et al.* How did We Get into This Mess Isolating Fault-inducing Inputs to SDN Control Software. Technical Report UCB/EECS-2013-8, 2013
- 14 Wundsam A, Levin D, Seetharaman S, *et al.* OFRewind: enabling record and replay troubleshooting for networks. Proceedings of USENIX Annual Technical Conference, Portland, OR, USA, 2011
- 15 Project Floodlight. <http://www.projectfloodlight.org/floodlight>, 2015
- 16 Kempf J, Bellagamba E, Kern A, *et al.* Scalable fault

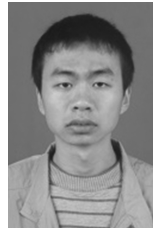
management for OpenFlow. Proceedings of 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 2012: 6606~6610

- 17 Huang W Y, Chou T Y, Hu J W, *et al.* Automatic end to end topology discovery and flow viewer on SDN. Proceedings of the 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Victoria, BC, Canada, 2014: 910~915

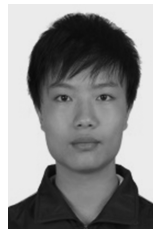
[作者简介]



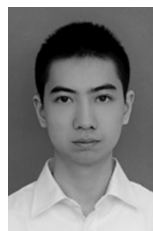
侯乐, 男, 北京邮电大学硕士生, 主要研究方向为软件定义网络、网络管理。



汪硕, 男, 北京邮电大学博士生, 主要研究方向为软件定义网络、网络虚拟化、网络管理。



林毅凯, 女, 北京邮电大学在读, 2014 年 2~8 月在北京邮电大学信息与通信工程学院未来网络理论与应用实验室参与软件定义网络课程实践, 研究课题为 SDN 在流媒体服务中的应用。



姚博韬, 男, 北京邮电大学在读, 2014 年 2~8 月在北京邮电大学信息与通信工程学院未来网络理论与应用实验室参与软件定义网络课程实践, 研究课题为 SDN 在流媒体服务中的应用。



刘江, 男, 博士, 北京邮电大学讲师, 主要研究方向为网络体系架构、网络虚拟化、软件定义网络、信息中心网络。