

# 软件定义网络的测量方法研究<sup>\*</sup>

戴冕<sup>1,2</sup>, 程光<sup>1,2</sup>, 周余阳<sup>1,2</sup>

<sup>1</sup>(东南大学 网络空间安全学院, 江苏 南京 211189)

<sup>2</sup>(教育部计算机网络和信息集成重点实验室(东南大学), 江苏 南京 211189)

通讯作者: 程光, E-mail: gcheng@njnet.edu.cn



**摘要:** 测量技术是状态监测、性能管理、安全防御等网络研究的基础,在网络研究领域具有重要地位.相较于传统网络,软件定义网络在标准性、开放性、透明性等方面的优势给网络测量研究带来了新的机遇.测量数据平面和测量控制平面的分离,启发了通用和灵活的测量架构的设计与实现;标准化的编程接口,使得测量任务可以快速地开发和部署,中心化的网络控制可以基于反馈的测量结果实时地优化数据平面的硬件配置和转发策略,数据平面基于流表规则的处理机制支持对流量更加精细化地测量.但是,软件定义网络测量中额外部署的测量机制造成的资源开销与网络中有限的计算资源、存储资源、带宽资源产生了矛盾,中心化的控制平面也存在一定的性能瓶颈,这是软件定义网络测量研究中的主要问题和挑战.分别从测量架构、测量对象两方面对当前软件定义网络测量研究成果进行了归纳和分析,总结了软件定义网络测量的主要研究问题.最后,基于现有研究成果讨论了未来的研究趋势.

**关键词:** 软件定义网络;网络测量;网络监测;网络管理;网络安全

**中图法分类号:** TP393

中文引用格式: 戴冕,程光,周余阳.软件定义网络的测量方法研究.软件学报,2019,30(6):1853-1874. <http://www.jos.org.cn/1000-9825/5832.htm>

英文引用格式: Dai M, Cheng G, Zhou YY. Survey on measurement methods in software-defined networking. Ruan Jian Xue Bao/Journal of Software, 2019,30(6):1853-1874 (in Chinese). <http://www.jos.org.cn/1000-9825/5832.htm>

## Survey on Measurement Methods in Software-defined Networking

DAI Mian<sup>1,2</sup>, CHENG Guang<sup>1,2</sup>, ZHOU Yu-Yang<sup>1,2</sup>

<sup>1</sup>(School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China)

<sup>2</sup>(Key Laboratory of Computer Network and Information Integration of Ministry of Education (Southeast University), Nanjing 211189, China)

**Abstract:** Measurement technology plays an important role in the field of network researching as it is the foundation of researching on network monitoring, network management, and network security. Compared with the legacy network, software-defined network brings new research opportunities due to its advantages of standardization, openness, and transparency. The separation of measurement data plane and measurement control plane motivates the design and realization of more general and flexible measurement frameworks. Standardized API enables quick development and deployment of measurement tasks. Logically centralized controller enables the network to optimize the hardware configuration and forwarding strategy in real time by analyzing the measurement results pulled from SDN switches. The forwarding mechanism based on flow table enables data plane to provide more fine-grained measurement data of network

\* 基金项目: 国家重点研发计划(2017YFB0801703); 国家自然科学基金(61602114, 61872080, 61502098); 江苏省自然科学基金(BK20151416); 赛尔网络下一代互联网技术创新项目(NGII20170406)

Foundation item: National Key Research and Development Program (2017YFB0801703); National Natural Science Foundation of China (61602114, 61872080, 61502098); Natural Science Foundation of Jiangsu Province (BK20151416); CERNET Innovation Project (NGII20170406)

收稿时间: 2018-05-21; 修改时间: 2018-12-07; 采用时间: 2019-01-20; jos 在线出版时间: 2019-03-27

CNKI 网络优先出版: 2019-03-27 16:54:37, <http://kns.cnki.net/kcms/detail/11.2560.TP.20190327.1654.013.html>

traffic. However, challenges are also brought due to the contradiction between extra network overhead and the limited computing, memory, bandwidth resources in network. The centralized control plane may also suffer from performance bottlenecks. This paper reviews current researches on measurement methods of software defined networking from two aspects: framework and object, and then concludes the main research tasks and challenges. In the end, the argument about future research trend is elaborated based on existing research experiences.

**Key words:** software defined network; network measurement; network monitor; network management; network security

测量技术是状态监测、性能管理、安全防御等网络研究工作的基础,在网络研究领域具有重要地位.近年来,互联网业务和应用飞速发展,尤其是高清音、视频服务逐渐成为主流,使得网络中的数据规模急剧膨胀.由于高速链路上的网络流量具有实时性、连续性、无界性的特点,即网络无法复现已经通过的流量,这使得处理算法只能对网络流量进行一次计算,因此,实时地处理海量规模的流量需要大量的计算和存储资源,这给网络测量工作带来了极大的困难<sup>[1]</sup>.

为了使用有限的计算和存储资源对高速流量进行测量,传统网络测量研究提出了两种流量压缩技术:抽样技术和数据流技术,通过牺牲一部分测量精度,压缩采样流量的规模,在高速网络测量研究领域取得了一定的成果.尽管如此,传统网络测量技术在实际应用时受到了很大程度的制约<sup>[2]</sup>.

- 首先是网络设备的制约,设备制造商为了提高转发性能,将设备的控制逻辑和转发逻辑紧密耦合在硬件 ASIC 交换电路和网络处理 CPU 中,导致设备的可扩展性很低<sup>[3]</sup>.由于设备制造商之间市场竞争的原因,设备的内部实现是完全封闭的,对用户只开放有限的接口用于配置和管理,这使得研究人员难以利用现有设备验证新的测量架构和测量算法<sup>[4]</sup>;
- 其次是网络架构的制约,传统网络是一个完全的分布式系统,每个网络节点只感知相邻节点的状态信息,然后通过分布式协议的计算完成数据的转发,测量人员难以掌握准确的网络全局信息,无法有效地进行网络的配置和管理工作.

为突破传统网络封闭的发展模式,斯坦福大学 Mckeown 教授于 2008 年发表了基于 OpenFlow 协议的新型网络原型设计的论文,并在 2009 年正式提出软件定义网络(software defined network,简称 SDN)的概念,旨在打破传统网络控制平面和数据平面之间的紧耦合,构建一个控制逻辑和转发逻辑相分离的网络架构<sup>[5]</sup>.2011 年, Mckeown 教授联合相关研究者成立了 ONF(open networking foundation,开放式网络基金会),负责 OpenFlow 协议的标准化制定和推广(最新版本为 1.6),极大地推进了 SDN 的标准化工作<sup>[6]</sup>.随着 SDN 成为未来互联网架构和网络功能虚拟化(network function virtualization,简称 NFV)领域的研究热点,美国、欧洲、日本相继在实验网络 GENI,Internet2,OFELIA,JGN2plus 中开展 SDN 的部署和研究,IETF,ITU,ETSI 等标准化组织开始研究 SDN 在各自领域的发展前景和应用,以 Nicira,Big Switch 为代表的新兴厂商和 Juniper,NEC,HP,IBM 等老牌厂商先后发布了支持 OpenFlow 协议的交换设备<sup>[7,8]</sup>.2013 年,Cisco 联合 BROCADE,CITRIX,DELL,ERICSSON, HP,INTEL, REDHAT,NEC 等公司,与 LINUX 基金会合作,成立了 OpenDayLight 开源组织,旨在厂商之间通过技术融合的方式共同打造一个开放、标准的 SDN 平台,推动 SDN 产业化的进程.目前,ODL 控制器已经发布到第 8 个版本(Oxygen),组织会员超过 50 个,涵盖了全球知名的设备制造商、运营商、服务商.到目前位置为止,以 Onix<sup>[9]</sup>,HyperFlow<sup>[10]</sup>,Floodlight,Ryu,ONOS<sup>[11]</sup>,OpenDayLight<sup>[12]</sup>为代表的控制器和以 Open vSwitch,Microsoft Hyper-V Virtual Switch,Cisco Nexus 1000V Virtual Switch 为代表的虚拟 SDN 交换机已经在云平台数据中心<sup>[13]</sup>、5G 无线移动网络<sup>[14]</sup>、传感器网络<sup>[15]</sup>、物联网<sup>[16]</sup>中得到了相当程度的应用<sup>[17,18]</sup>.

软件定义网络在开放性、透明性、标准性等方面的巨大优势,给网络测量研究工作注入了新活力<sup>[19]</sup>.因此,近年来学术界在软件定义网络测量领域进行了大量的研究工作,提出了很多有价值的测量架构与测量方法,测量对象涉及各个领域.一些文献<sup>[20-28]</sup>综述了软件定义网络中流量工程和安全防御相关研究的进展,但是其中涉及软件定义网络测量的相关篇幅较少,因此,本文分别从测量架构和测量对象的角度对软件定义网络测量目前的研究成果进行了完整和系统的归纳总结,并分析阐述了主要的研究问题和挑战,最后展望了未来的研究方向.

## 1 研究背景

正常运行的网络由网络设备、网络链路和网络流量组成,从广义上来说,网络测量是同时在物理和逻辑的层面对网络每个组成部分的运行状态和行为模式进行监测和统计,测量结果用于指导网络管理工作,网络测量与网络管理的关系如图 1 所示。

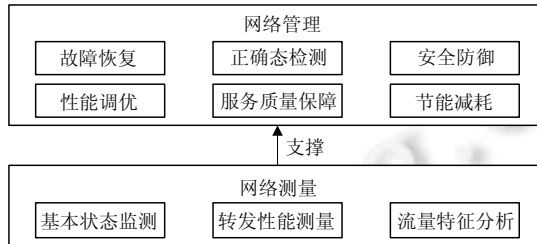


Fig.1 Relationship between network measurement and network management

图 1 网络测量与网络管理的关系

近年来,随着硬件监测技术的不断成熟,因此当前的网络测量研究工作主要集中于对网络实际流量的分析和检测,以此来估计网络的运行参数.基于流量的测量方法主要包含两个要素:测量架构和测量对象,因此,本文就从测量架构和测量对象这两个角度对当前的软件定义网络测量研究工作进行归纳和分析。

传统网络测量技术由于受到网络设备和网络架构的制约,测量架构的实现方式非常有限,因此研究的重点在于流量的采集方法和压缩手段。

- 采集方法从类型上可分为主动采集和被动采集两种<sup>[29]</sup>.
  - 主动采集是指在网络链路中注入特定的探测报文,然后通过实时、连续地收集探测报文在网络中的行为数据并处理分析来得到所需的测量信息.这种方法可以测量网络实时的行为,但是探测流量在一定程度上会增加网络负载;
  - 被动测量方法通过在网络中若干个预先设定的节点采集数据,一般由交换机根据采集规则主动地向测量服务器转发测量流量,被动测量不引入额外的流量,可以用于较大规模流量的采集和分析;
- 流量压缩手段主要包括抽样技术和数据流技术.
  - 抽样技术是从总体流量数据中筛选出代表性的子集,通过对该子集的测量结果推断原始流量的特征信息.NetFlow 和 SFlow 是抽样技术中具有代表性的应用实例;
  - 数据流技术是使用特定的哈希函数对报文进行计算,将原始报文数据映射存储到 sketch 中.sketch 是一种专门设计的数据结构,它利用哈希函数的性质,可以用很少的存储空间保存大量数据的统计特征信息.基于 sketch 数据结构的数据流算法在流量分析领域得到了广泛应用,一般来说,一种 sketch 结构针对一种或者几种流量统计特征而设计,常用的 sketch 结构有 Bitmap,Bloom Filter,Count-min sketch,Count sketch 等,结合若干个 sketch 可以实现更加复杂的流量特征的测量.基于数据流技术的主要流量分析算法包括流量熵估计、流量矩阵(traffic matrix,简称 TM)估计、网络节点连接度估计、流长分布估计等。

软件定义网络在不改变 IP 网络基础的前提下重构了网络控制平面和数据平面,一方面使得传统网络测量技术中的流量采集方法和压缩手段可以应用在软件定义网络测量中;另一方面,使得网络测量架构的设计灵活性大大增加.软件定义网络测量的优势主要体现在以下几个方面。

- 1) 测量控制平面和测量数据平面相分离.在网络测量架构中,测量算法的部署离不开控制平面和数据平面的支持,SDN 在分离控制平面和数据平面的同时,也意味着将测量控制平面和测量数据平面相分离,这大大增强了测量架构的设计灵活性,使得待测流量的获取方式、测量节点的部署位置、测量算

法的处理流程、测量结果的反馈机制变得更加自由;

- 2) 基于 OpenFlow 协议的交互机制.软件定义网络提供了标准的南向协议 OpenFlow 用于控制器与交换机之间的通信,控制器通过 OpenFlow 消息下发测量流表规则,查询报文统计信息,交换机通过 OpenFlow 异步消息向控制器发送通知.软件定义网络测量方法可以充分地利用 OpenFlow 协议消息所携带的信息进行测量;
- 3) 基于流表规则的报文处理机制.SDN 交换机基于控制器下发的流表项对报文进行转发,流表头域的元组包含了网络层和传输层的协议字段,这使得网络处理过程扁平化,支持对流量更加细粒度的测量;
- 4) 标准的编程接口.SDN 控制器支持控制器内部扩展模块的开发,同时提供标准的北向 API 给上层应用使用,这使得测量应用可以快速地开发和部署;
- 5) 与传统网络测量技术的兼容性.软件定义网络虽然重构了网络控制平面和数据平面,但是并没有改变底层的 IP 协议,这使得传统网络测量研究的成果可以应用在软件定义网络测量中,并且结合 SDN 的优势以更加灵活的方式实现.

## 2 软件定义网络测量架构

测量架构是网络测量的总体闭环过程,结合文献[2],该过程可以分为 3 个主要步骤——流量的采集和预处理、流量的传输和存储以及流量的分析与反馈.

- 1) 流量的采集和预处理.该步骤是流量测量过程中最重要的一环,主要包含 4 个问题——采集方法、采集位置、采集频率和处理格式:采集方法是指流量通过何种方式采集,从类型上可分为主动采集和被动采集;采集位置代表在网络中的哪些节点进行采集;采集频率代表两次采集之间的时间间隔;采集格式代表原始流量经过预处理后的传输和存储格式.由于原始报文体积较大,一般在处理时只提取针对测量对象所需的信息,通常是报文头部的若干元组或者报文的固定字节区域等;
- 2) 流量的传输和存储.该步骤包含两个方面的问题:一方面是经过采集和预处理后的测量流量通过何种方式转发至何处进行临时存储,另一方面是如何将临时存储的测量流量发送至分析服务.不同的传输和存储机制对测量架构所占用的带宽、存储资源和测量实时性有着较大影响;
- 3) 流量的分析和反馈.该步骤根据测量对象选取算法对流量数据进行分析处理,获取所需的流量统计特征信息,例如流量类型、报文数量统计、流长统计、流量熵估计等.同时,将测量结果以实时、半实时或者离线的方式反馈给网络管理者,用于网络的优化配置.

结合传统高速网络测量方法的评价指标,本文将软件定义网络测量架构的评价指标定义如下.

- 1) 实时性.反映测量架构线速地处理分析高速流量数据的能力;
- 2) 并发性.反映测量架构同时执行多个网络测量任务的能力;
- 3) 可扩展性.反映测量架构支持多个种类的测量任务的能力;
- 4) 准确性.反映测量架构保障测量任务测量精度的能力;
- 5) 复杂性.可分为计算复杂性、存储复杂性和通信复杂性:计算复杂性反映测量架构占用控制器中计算资源的程度,存储复杂性反映测量架构占用交换机存储资源的程度,通信复杂性反映测量架构引入的控制器和交换机之间链路开销的程度.

这些指标在一定程度上存在互相矛盾的关系,即测量架构针对某一指标性能的优化可能导致另一指标性能的下降.因此,软件定义网络测量架构研究工作主要围绕如何平衡上述指标之间的关系,寻求最优化的解决方案.本节基于测量架构所包含的 3 个方面对当前的研究工作进行归纳总结.

### 2.1 流量的采集和预处理

软件定义网络测量中的流量采集方法类似于传统网络的主动采集和被动采集,可以分为基于查询的采集方法和基于推送的采集方法.由于 SDN 交换机中部署的流表本身记录流量的基本统计信息,并且支持 SDN 控制器通过 OpenFlow 消息进行查询,所以基于查询的采集方法就是利用控制器周期性地轮询交换机上流级别的统

计信息.这种方法会造成一定的网络开销,因此需要对查询频率和测量精度进行平衡.基于推送的采集方法是通过控制器向 SDN 交换机下发测量流表规则,将匹配规则的待测流量转发至测量服务器.这种方法缓解了控制器的性能压力,不引入额外流量,但是需要占用交换机中有限的流表空间,对交换机造成一定的额外性能开销.传统网络测量一般通过在交换机上开启端口镜像或者部署测量代理的方式采集流量,成本较高且灵活性较差.软件定义测量中可以通过控制器动态地调整采集位置和采集频率.在流量的预处理阶段,相较于传统网络测量一般使用的五元组或七元组过滤规则,软件定义网络测量支持流表头域所包含元组的任意组合的规则,可以实现更加特定的流量采集.

### 2.1.1 基于流表规则

软件定义网络中,控制器通过安全通道来配置、管理交换机,下发、更新交换机中部署的流表规则,交换机按照流表规则对流量进行处理(转发、丢弃、缓存).流表由若干条流表项组成,标准 OpenFlow 协议中规定了流表项包含 3 个部分:头域(header fields)、统计域(counters)和动作域(actions).其中,头域定义了报文的匹配规则,动作域指定了对匹配报文的动作,而统计域则保存了该条流表项所匹配报文的基本统计信息.统计域提供了基于 4 个维度(流表、流、接口、队列)的统计信息,该域内所有统计项的值都在 SDN 交换机运行时自动更新.一些研究工作通过控制器直接获取交换机流表中的统计数据,并利用这些信息进行测量.OpenTM<sup>[30]</sup>使用基于查询的采集方法,利用控制器路由模块中的路由信息探测并跟踪全网的活动流量,然后周期性地从流量经过的交换机的流表中逐项查询并获取统计域的信息.Payless<sup>[31]</sup>通过在控制器中部署测量模块,将上层测量应用发起的测量任务解析为对交换机流表统计域的查询请求.除了直接获取交换机提供的标准统计信息之外,还可以通过控制器主动下发测量流表规则来针对特定的流量进行测量.OpenFlow 协议 1.0 版本规定流表项的匹配域包含十二元组,长度为 250 比特.在后续的更新中,为了扩展功能又陆续加入了元数据、MPLS 标签、MPLS 类型、IPv6、可选字段等,因此,灵活地组合流表项中的匹配域字段可以构造更细粒度的测量规则,满足更加精细化的测量需求.SLAM<sup>[32]</sup>通过主动构造并下发测量流表规则实现了端到端的时延测量,该方法事先在源交换机和目的交换机之间指定报文的转发路径,然后在该路径上的每个交换机中部署测量流表规则,控制器下发符合测量规则的探测报文,该报文按照指定路径转发时,在沿途的每个交换机触发带有时间戳的异步 OpenFlow 消息返回至控制器,控制器根据这些时间戳来估算链路时延.与 SLAM<sup>[32]</sup>类似,OpenNetMon<sup>[33]</sup>预先在数据平面下发好测量流表规则,然后从终端主机向网络中主动地发送探测报文,通过在设定好的网络节点上依次捕获探测报文并解析其中所含的网络状态信息,判断端到端的 QoS 参数是否对每条流正确的生效.OpenMeasure<sup>[34]</sup>使用基于学习的预测方法动态地通过控制器更新交换机测量流表规则.典型的基于流表规则的测量架构如图 2 所示.

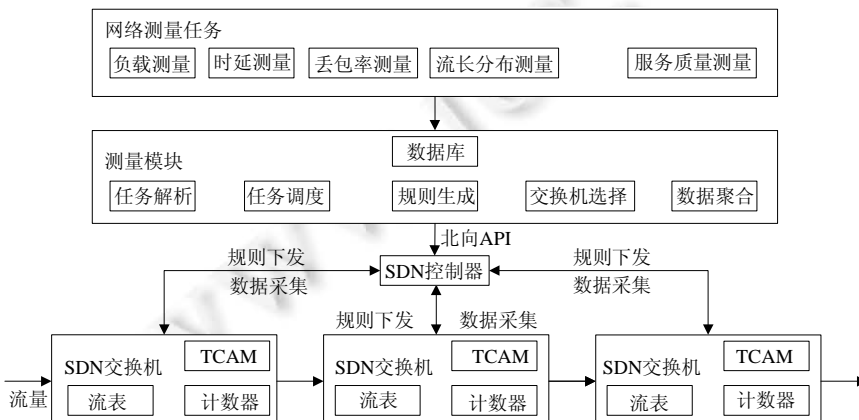


Fig.2 Measurement architecture based on flow rules

图 2 基于流表规则的测量架构

### 2.1.2 基于端口镜像和代理

基于流表规则的流量采集和预处理方法的优点是实现方便,使用商用 SDN 交换机即可部署,不需要额外的软硬件支持.但是频繁地操作控制器对交换机进行查询会给控制器和交换机造成不小的性能压力,在控制器带内(in band)部署的情况下还会占用一定的带宽资源,可能对网络正常的转发性能产生影响.因此在采集较大规模的流量时,一些研究工作使用基于端口镜像或者代理的方法.

端口镜像是指交换机将从一个端口接收的流量镜像复制到另一个端口中发送,商用交换机一般都具备此功能,该方法在不超过端口容量限制的前提下可以实现线速地转发.PLANCK<sup>[35]</sup>在商用 SDN 交换机中利用端口镜像技术将测量流量转发至独立的测量服务器进行离线的分析处理,测量报文间隔在 1Gbps 的线速下为 280 $\mu$ s~7ms,在 10Gbps 的线速下为 275 $\mu$ s~4ms.

测量代理是实现了特定测量功能的软件或者硬件模块,一般部署在网络设备之上或者作为中间件(middle box)部署在网络设备之间.测量代理对经过的流量进行拦截、修改并重定向,将流量重新引导至测量服务器.代理分散了控制器的性能压力,并且不受控制器本身错误的影响,但是实现成本较高,灵活性较低.MicroTE<sup>[36]</sup>以数据中心中的机架为单位统计流量信息,用于网络全局的路由决策.该方法在数据中心的每台服务器上部署代理测量模块,将服务器的流量信息发送至该机架的 ToR(top of rack)服务器,ToR 汇总流量后再发送至控制器.Hone<sup>[37]</sup>同样通过部署在主机上的软件代理采集流量数据.OpenSample<sup>[38]</sup>在数据平面使用部署了 sFlow 代理的 OpenFlow 交换机,网络流量根据流表中部署的测量规则进行过滤后,再经过 sFlow 代理抽样汇总至 sFlow 流量采集器,提供给上层的流分析器和端口分析器使用.OFRewind<sup>[39]</sup>以软件代理的方式串联部署在控制平面与数据平面之间的控制链路中,拦截并修改控制器发送的控制消息,指导数据平面交换机将所需要的流量转发至专门的流量存储服务器中.Veriflow<sup>[40]</sup>在控制器和交换机之间部署代理服务器,对控制器和交换机之间双向通信的每条 OpenFlow 控制消息进行检测,在每条 OpenFlow 转发规则下发时检测网络的一致性冲突,判断该次更新是否会产生网络异常.NetSight<sup>[41]</sup>在控制器和交换机之间部署代理服务器,通过修改控制消息在网络中的每台交换上增加处理探测报文的规则.典型的基于代理的测量架构如图 3 所示.

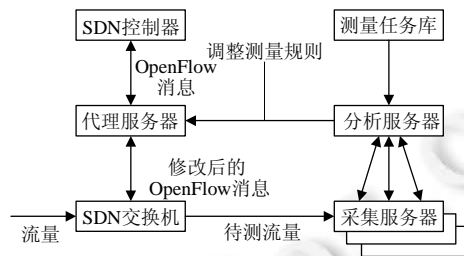


Fig.3 Measurement architecture based on proxy

图 3 基于代理的测量架构

### 2.1.3 基于可编程 SDN 交换机

在基于商用 SDN 交换机实现的软件定义网络测量架构研究之外,一些文献开始研究对测量数据平面进行改造,结合抽样技术和数据流技术在数据平面部署测量机制和算法,以实现更加复杂的测量架构.为了对 SDN 交换机的数据平面进行改造,这些研究工作使用可编程网络设备进行组网验证.可编程网络设备分为软件和硬件两种实现:软件实现目前以虚拟交换机 Open vSwitch 为代表,硬件实现主要以高性能可重构硬件 FPGA 作为核心进行编程改造.基于虚拟 SDN 交换机实现的测量架构开发和部署较为方便,可以在较大规模的网络环境中进行验证.基于硬件 SDN 交换机实现的测量架构由于可以充分利用高速 TCAM 存储器和 SRAM 存储器,可以达到较高的性能.

基于可编程 SDN 交换机实现的流量采集和预处理方法的核心思想是直接在交换机数据平面报文处理流程中部署过滤、抽样和哈希算法,直接对原始流量进行处理.vPROM<sup>[42]</sup>基于改造的 Open vSwitch(UMON<sup>[43]</sup>)设计了测量系统,它对 OpenFlow 协议进行了扩展,使得 RYU 控制器可以直接控制 UMON 交换机,部署在控制器

中的 Pyretic 模块将上层测量应用解析为流表规则下发至交换机进行流量采集.iSTAMP<sup>[44]</sup>使用硬件高速 TCAM 存储器对流量进行采集和预处理,由于交换机中的 TCAM 容量相当有限,无法对每条流进行统计,因此 iSTAMP 将 TCAM 分为两部分:一部分用于流量的聚合,测量聚合后流量的统计信息,同时标记被认为带有重要信息的流量;另一部分用于流量的解聚合,将被标记的特定流从流量聚合中还原出来单独进行测量.OpenSketch<sup>[45]</sup>在基于 NetFPGA 实现的可编程 SDN 交换机中部署了哈希-分类-计数的三级数据处理管道:第 1 级管道提供多种哈希算法用于压缩存储流量数据,第 2 级管道使用 TCAM 中的通配符规则对流量进行分类,第 3 级管道使用 SRAM 作为计数器统计流量信息.Yu 等人<sup>[46]</sup>在数据平面部署了一个两级的过滤算法,并将处理后的报文存放在 Bloom Filter 中.Wellem 等人<sup>[47-51]</sup>基于 NetFPGA 实现了可编程 SDN 交换机的原型系统并进行了大量的相关工作,在数据平面实现了解析-哈希-缓存-计数的报文处理流程,支持 10Gbps 线速下的实时测量.Dai<sup>[52]</sup>基于 FPGA 实现的可编程 SDN 交换机 FAST 设计了网络拓扑和报文路径的检测方法,该方法在数据平面部署哈希函数对探测报文的头部进行计算,并将报文的路径信息存放在 SRAM 中.典型的基于可编程 SDN 交换机的测量架构如图 4 所示.

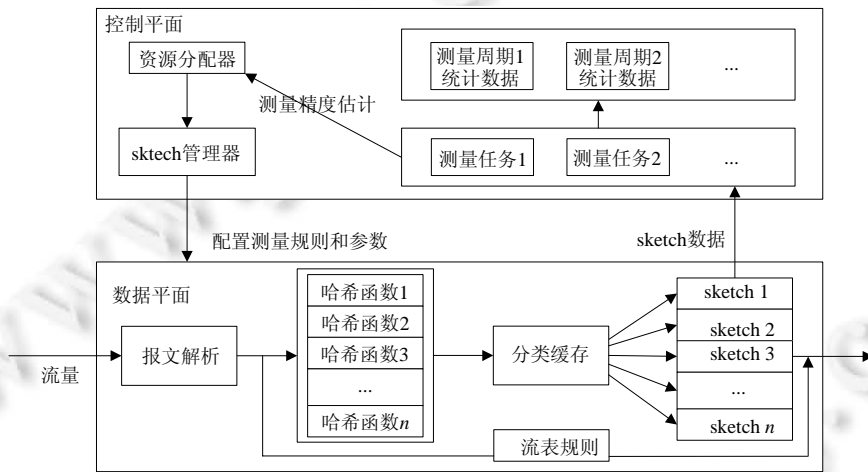


Fig.4 Measurement architecture based on programmable SDN switch  
图 4 基于可编程 SDN 交换机的测量架构

## 2.2 流量的传输和存储

传统网络测量由于设备的封闭性,待测流量一般通过代理转发或镜像转发至独立的测量服务器进行存储.软件定义网络测量方法中一方面可以通过标准的 OpenFlow 同步和异步消息将测量流量传输至控制器;另一方面,一些研究工作对可编程 SDN 软硬件交换机的数据平面进行改造,将少量测量结果的统计数据保存在数据平面,通过牺牲一定的可扩展性提高测量速率.

### 2.2.1 基于 OpenFlow 协议

软件定义网络中控制器和交换机之间通过 OpenFlow 协议进行交互,控制器向交换机发送的消息类型包括 Features, Configuration, Modify-State, Read-State, Send-Packet 和 Barrier,交换机向控制器主动发起的异步消息类型包括 Packet-in, Flow-Removed, Port-State 和 Error.其中,与网络测量相关的消息类型主要是 Read-State, Send-Packet, Packet-in 和 Flow-Removed.控制器通过 Read-State 消息获取交换机中的流量统计信息,发送 Stat-Request 请求通知交换机,交换机将流表统计域中存储的计数值通过 Stats\_Reply 消息返回至控制器,OpenFlow1.0 协议中原生的 SSR(single stat-request)请求,可以获取一条特定流的统计信息.Send-Packet 消息用于控制器向指定交换机的指定端口发送报文,在主动发送探测报文的测量架构中,不但可以从主机向网络中注入探测流量,还可以利用这一消息操作控制器向网络中注入探测流量.Packet-in 消息是当交换机接收到的报文无法在流表中查询到匹配项时向控制器发送的事件消息,消息中包含报文的头部信息.Flow-Removed 消息是当交换机流表中某一

流表项因长期无报文匹配而触发超时时向控制器发送的事件消息。Openflow 1.5 版本中加入了 FEST(flow entry statistics trigger)功能,可以在流表统计域中设置阈值,当统计值超过阈值时,触发交换机向控制器发送异步事件消息。

通过 OpenFlow 协议消息,可以方便地在控制器和交换机之间传输测量数据,并且具有较好的实时性。文献[30,31,33,53]利用 Read-State 消息从交换机获取流量统计信息,CeMon<sup>[53]</sup>对在控制器分布式部署的环境下,Stat-Request 查询请求聚合的方法进行了研究。文献[32,54-56]利用交换机主动发送的异步消息来进行测量,Wang 等人<sup>[57]</sup>结合了监听 FEST 消息、监听 Flow-Removed 消息、主动查询这三种方式,使用可编程 SDN 交换机 NetMagic 获取流量信息.Flowsense<sup>[54]</sup>持续地监听交换机和控制器之间交互的 PacketIn 消息和 FlowRemoved 消息,根据这两条消息的时间戳得到一条流在网络中的持续时间.SLAM<sup>[32]</sup>和 SDNtraceroute<sup>[55]</sup>主动向网络中发送探测报文,利用探测报文经过每一跳交换机返回的 PacketIn 消息进行测量,SLAM 记录报文经过每一跳的时间戳用于链路时延的估计,SDNtraceroute 记录报文的真实转发路径.Suárez-Varela 等人<sup>[56]</sup>根据交换机发送的 FlowRemoved 事件消息来触发控制器对该条流的统计值的查询。

### 2.2.2 基于数据平面 sketch

基于可编程 SDN 交换机实现的测量架构研究一般将预处理后的流量数据以 sketch 结构的格式缓存在数据平面内存或 SRAM 存储器中,在每个测量周期结束后将缓存数据发送至控制器进行分析。一些研究工作还在可编程 SDN 交换机中部署了协议适配层,将自定义的控制协议与 OpenFlow 协议进行适配和扩展,使得改造后的交换机可以通过 OpenFlow 协议进行管理。

Wellem 等<sup>[47,48,51]</sup>分别针对超点(Superspreader)检测、变化(Heavy Change)检测和大流(Heavy Hitter)检测在数据平面设计了基于 sketch 数据结构的报文处理结构,该结构组合了缓存队列和多种 sketch,实现了在较高线速下流量的实时处理。JOTA<sup>[58]</sup>和 SCREAM<sup>[59]</sup>讨论了如何通过动态资源调度的方式利用数据平面有限的 TCAM 和 SRAM 资源实现不同类型测量任务的并发执行,并保证一定的精度,测量架构使用了包括 PCSA sketch、Count-min sketch、HyperLogLog sketch 在内的多种 sketch 数据结构进行了验证。SketchVisor<sup>[60]</sup>基于虚拟交换机 Open vSwitch 设计了基于 sketch 的数据平面,并对目前主流的 sketch 数据结构(FlowRadar<sup>[61]</sup>,RevSketch<sup>[62]</sup>,UnivMon<sup>[63]</sup>,Deltoid<sup>[64]</sup>,OpenSketch<sup>[45]</sup>,FM<sup>[65]</sup>,kMin<sup>[66]</sup>,Linear Counting<sup>[67]</sup>,MRAC<sup>[68]</sup>)进行了验证。OpenSketch<sup>[45]</sup>在数据平面部署了一个测量库,其中包含了多种哈希算法和 sketch 数据结构,提供了 API 来支持不同测量任务的开发和部署,使用这个测量库可以自动的为不同的 sketch 配置数据平面管道并且分配相应的交换机资源使得任务达到最大的精度。UnivMon<sup>[63]</sup>在数据平面使用多个 Count sketch 数据结构对测量数据进行并行抽样计算,支持多种测量算法。

## 2.3 流量的分析和反馈

传统网络测量中需要部署独立的测量服务器对待测流量进行离线地分析,网络管理者根据测量结果人为地调整网络配置。软件定义网络测量中,由于软件定义网络采用中心管控的架构,网络中的所有交换机都可以通过控制器进行集中地配置和管理。在待测流量较小、实时性要求较高的测量任务中,可以通过在控制器中部署扩展模块,利用监听程序直接获取测量数据并分析,测量结果实时反馈至控制器对网络配置进行调整。在对实时性要求相对较低的半实时测量任务中,可以在控制器北向部署测量应用通过北向 API 与控制器交互进行更大规模流量的测量分析。测量模块主要基于控制器部署,根据应用场景的不同可分为两种部署方式。

- 一种方式是作为扩展模块部署在控制器内部,此方式需要修改和扩展控制器程序,开发难度较大。优点在于部署在控制器内部的测量模块可以监听控制器发送和接收的全部流量,并接管控制器内部的处理流程,调用控制器发送和接收控制 OpenFlow 消息和测量数据,适用于实时性较高的测量任务。文献[30,31,33,54]通过在控制器中部署测量模块的方式实现,其中,Payless<sup>[31]</sup>在 FloodLight 控制器中以 RESTAPI 的形式编写了上层服务接口,支持用户自定义的测量构件,测量任务使用 JSON 格式的文件来定义,交换机反馈的统计信息汇总到控制器进行处理,为上层应用提供一个抽象的网络信息视图;
- 在控制器内部部署扩展模块的方式会对控制器性能造成一定影响,同时受到控制器有限的计算资源



和存储资源的显示,因此涉及较大规模流量、较为复杂算法的测量任务通常使用另一种部署方式,这种方式中,测量模块部署在控制器之上的应用层,通过北向 API 与控制器进行交互,这种方式一般用于半实时的测量结果反馈(通常以一个测量周期为单位).文献[56,58,59,69,70]将多种较为复杂的流量检测算法部署在控制器之上的应用层,在每个测量周期结束后通过控制器获取网络中所有交换机数据平面中的 sketch 数据进行分析处理.测量结果反馈到控制器,指导下一个测量周期内交换机测量参数的配置.

### 3 软件定义网络测量对象

测量对象一般可分为网络状态参数、网络性能参数和网络流量参数:网络状态参数包含网络节点的配置信息、网络链路的状态信息以及网络的拓扑结构,是网络的基本参数;网络性能参数包括链路吞吐率、链路丢包率和链路时延等,该参数对测量的实时性要求较高,反映网络的瞬时运行状态;网络流量参数一般通过对一定周期内的网络流量进行收集分析获得,是一个统计参数,例如报文数量统计、流长统计等,对流量参数进行深入挖掘可以获得网络潜在的信息.

#### 3.1 网络运行状态

网络运行状态包含拓扑结构、链路状态、控制逻辑等,对其进行测量的主要目的是监测网络运行时的相关指标和行为,及时地发现网络异常和故障,确保网络运行在正确状态.

##### 3.1.1 故障检测

SDN 控制器通过 LLDP 协议来探测网络拓扑结构,控制器将 LLDP 报文作为 Packet-out 消息发送至全网的交换机,SDN 交换机收到报文后将其转发至所有相邻的交换机,相邻交换机收到报文后,会以 Packet-in 的形式传回控制器.这样,控制器就得到了所有交换机的相邻信息,并以此构建出整个网络拓扑.当网络因为故障导致拓扑结构和链路状态发生改变时,需要通过测量手段及时的定位故障.OFRewind<sup>[39]</sup>在控制器和交换机之间部署了流量记录与流量重放模块:流量记录模块控制交换机将测量流量转发至流量存储服务器中,流量重放模块控制流量存储服务器将流量数据重新注入到网络中.通过这种方式,该方法可以在全网络范围内复现网络流量,方便故障诊断工作的定位.Zeng 等人<sup>[71]</sup>提出了一种自动化测试报文生成工具——ATPG(automatic test packet generation),ATPG 通过读取路由配置文件生成设备无关的模型.该模型被用来生成最小的测试报文集,生成的测试报文最少测试网络中的每条链路,最多测试网络中的每条规则.ATPG 的目标是检测静态状态的故障,但是该方法不能检测动态变化网络中的故障.Kempf 等人<sup>[72]</sup>实现了一个可扩展的故障管理系统,通过控制器向逻辑上的组端口发送探测报文来检测网络故障.

##### 3.1.2 逻辑一致性检测

网络控制逻辑和转发逻辑一致性是指数据平面在对报文进行转发的过程中,完全按照控制器下发的规则执行正确的行为.在软件定义网络中,尽管数据平面得到了简化,但是控制平面存在很高的软件复杂度.尽管控制平面掌握网络的全局信息,将高层的控制平面逻辑描述解析成为低层的数据平面转发行为,对数据平面进行统一地调度部署,理论上可以保证网络配置策略的一致性,但是也存在一定的隐患.控制平面部署的多个 SDN 应用并发执行时,下发的流表规则可能发生潜在的逻辑冲突,或者当控制器和交换机内部发生一些错误时,可能会导致控制平面的转发逻辑与数据平面的实际行为不一致,而控制器本身并不能对这一情况进行感知.为了对这样的故障进行检测,就需要对数据平面的真实行为进行测量.Veriflow<sup>[40]</sup>设计了面向 SDN 网络的实时一致性检测工具,它对控制器和交换机之间双向通信的每条 OpenFlow 控制消息进行检测,通过图搜索技术结合网络拓扑来验证网络中的不变量,只有当不变量成立时才将流表规则下发至网络设备,从而保证下发规则的正确性.Veriflow 根据最长前缀匹配规则将全网报文基于目的 IP 地址进行分类,无论是转发规则的插入、更新或是删除,一次规则检测只会影响到以一个前缀为目的地址的报文.通过这种方式,在平均每条流表规则的下发间隔小于 10ms 的 SDN 网络中,Veriflow 可以在数百毫秒的时间内完成验证.

对报文在网络中的真实转发路径进行测量,也是检测控制平面与数据平面逻辑一致性的有效手段.

NetSight<sup>[41]</sup>通过主动发送探测报文的方式检测报文在网络中的真实路径,它在网络中的每台交换机上部署处理探测报文的流表规则.当探测报文经过网络中每一跳交换机时,除了被正常转发之外,还将事件记录作为一个 postcard 发送至专门的测量服务器,每个 postcard 包含报文头、交换机 ID、出端口、交换机状态.测量服务器通过将所有 postcard 与网络的拓扑信息相结合,就可以重建出报文的实际路径.CherryPick<sup>[73]</sup>通过将路径信息存放在报文头部中来测量报文路径,它利用数据中心网络中 Fat-Tree 拓扑结构的性质,通过边缘着色算法对网络中非必要的链路进行删减,有效地减少了交换机中所需部署的测量规则的数量以及报文头部的空间占用,同时保证报文路径的正确恢复.SDNtracerooute<sup>[55]</sup>充分利用了软件定义网络中控制平面和数据平面交互的实时性,通过在网络中所有交换机上部署测量规则,使得探测流量每经过一个交换机都通过 PacketIn 消息发送回控制器.控制器解析探测报文中的路径信息,对其作一定修改再重新下发回网络中.这样,控制器根据探测报文每一跳的信息可以容易地重建其真实的转发路径.Dai<sup>[52]</sup>基于 FPGA 实现的 SDN 交换机 FAST 设计了检测网络拓扑和报文路径的测量方法,该方法通过在 FAST 中添加 3 条流表规则,对探测报文进行匹配处理.FAST 中部署的哈希函数对探测报文的头部信息进行计算,并将计算结果作为 SRAM 索引地址,然后将相关的路径信息存入相应的内存空间.控制器中部署相同的哈希函数,计算得到探测报文在 SRAM 中的存放地址,根据该地址收集报文的全部路径信息.该测量机制支持一定规模的并发路径探测任务.

### 3.2 网络转发性能

网络转发性能一般通过 3 个指标反映——吞吐量、丢包率和时延.

- 1) 吞吐量分为端口吞吐量和链路吞吐量,是指端口、链路在单位时间内成功传送数据的数量.其中,端口吞吐量测量的基本方法是:由控制器向交换机发送 ofp\_port\_stats\_request 消息,查询交换机中的端口统计信息,交换机返回 ofp\_port\_stats 消息,通过对其中的 rx\_bytes 和 tx\_bytes 数值进行计算得出.链路吞吐量由构成链路的两个端口的吞吐量决定,其值等于二者中的较小者;
- 2) 丢包率是通过链路两端的报文数量之差与源端报文总数之比.测量丢包率的基本方法是:分别在  $t_1$  时刻和  $t_2$  时刻通过控制器发送 ofp\_port\_stats\_request 消息,获取链路两端交换机端口的发送报文数目和接收报文数目.假设  $t_1$  时刻和  $t_2$  时刻  $sw_1$  交换机  $port_2$  端口发送报文数量差值为  $(\Delta t_x \text{ packets})$ ,  $sw_2$  交换机  $port_1$  端口接收报文数量差值为  $(\Delta r_x \text{ packets})$ , 则链路  $((sw_1, port_2), (sw_2, port_1))$  在  $(sw_1, sw_2)$  方向的丢包率为:  $(\Delta t_x \text{ packets} - \Delta r_x \text{ packets}) / \Delta t_x \text{ packets}$ ;
- 3) 数据报文在网络中的时延由两部分组成:链路传播时延和交换机处理时延.链路传播时延是指数据报文在交换机连接介质上进行传输所花费的时间,由传输介质的物理性质决定;交换机处理时延是数据报文经过交换机时所花费的时间,在 SDN 交换机中处理时延由 3 部分组成:查找时延、转发时延和控制时延.查找时延是新报文到达时,交换机查找流表匹配项所花费的时间;转发时延是交换机将报文从入端口迁移到出端口所花费的时间,报文在出端口发送队列停留的时间受队列中的报文数量和优先级影响;控制时延是当新报文在交换机流表中查询不到匹配项时,交换机向控制器请求下发新流表项所花费的时间.

Flowsense<sup>[54]</sup>是一个基于推送的被动测量方法,它利用交换机主动向控制器发送的 PacketIn 消息和 FlowRemoved 消息中携带的信息计算链路的吞吐率.Flowsense 的优势在于其较低的测量开销,但是由于被动方式采集到的信息时间点较为离散,对于持续时间较长的流进行测量时精度较低.SLAM<sup>[32]</sup>在网络中任意交换机之间测量时延,测量架构包含 4 个模块:规则生成器、流量生成器、流量监听器和实验估计器.测量分为 3 个步骤:首先指定端到端的转发路径,在路径上的每个交换机上部署测量流表规则;然后,控制器下发探测报文,探测报文每经过一个交换机会触发一个 packet\_in 消息发送至控制器;最后,控制器根据每个 packet\_in 消息的时间戳来估算时延.SLAM 通过 Statistics\_Request 消息和 Statistics\_Reply 消息消除了控制时延带来的误差,但是没有考虑查找时延和转发时延,因此测量精度会受到交换机本身性能和负载的影响.PLANCK<sup>[35]</sup>通过跟踪 TCP 报文的序列号和接收时间来计算 TCP 流的吞吐量,然后将一条链路上所有流的吞吐量相叠加来计算链路利用率.OpenNetMon<sup>[33]</sup>设计了一个 SDN 网络性能测量系统,根据在预设网络节点上查询到的探测报文信息持续地估

算网络的性能参数,具体的计算方式为:对流量转发路径上的最后一个交换机进行查询,获取交换机端口计数器中测量周期  $T$  内通过的报文数目  $S$ ,通过计算  $S/T$  可以得到吞吐率;分别查询转发路径上的第 1 个和最后一个交换机中测量周期  $T$  内的报文增量,两者的差值除以第 1 个交换机的报文增量值可以得到这两个节点之间的丢包率;主动发送探测报文穿过所有预定节点后返回控制器,根据控制器记录的时间戳之差再减去控制器到数据平面之间的往返时延可以得到所探测路径的链路传输时延。

### 3.3 网络流量特征

网络中传输的流量是网络测量研究工作中最重要的测量对象,对流量进行深入地挖掘和分析可以发现网络中潜在的问题,这是指导网络管理和安全工作的重要依据。网络管理者根据流量的统计特征可以制定更好的管理策略和安全策略。流量统计特征的主要表征参数包括:

- 1) **Cardinality**: 一个测量周期内测量的不同流的数量;
- 2) **Flow size distribution**: 一个测量周期内所有流的大小的分布比例;
- 3) **Entropy**: 流长分布的熵值;
- 4) **Heavy Hitter**: 测量周期内一条流的大小超过一定阈值;
- 5) **Hierarchical Heavy Hitter**: Heavy Hitter 的扩展,基于报文头部的某个或者某些字段最长前缀聚合的 Heavy Hitter,且它的大小减去所有子 Heavy Hitter 节点的大小之和后仍然超过一定阈值;
- 6) **Heavy Change**: 一条流在相邻周期之间的大小之差超过一定的阈值;
- 7) **DDoS(distributed denial of service)**: 一个目的主机在测量周期内接收的来自不同源主机请求的数量超过一定阈值;
- 8) **Superspreader**: 一个源主机在测量周期内向不同目的主机发送请求的数量超过一定阈值。

#### 3.3.1 流量工程

网络流量特征可以有效地指导流量工程相关的研究工作通过调度网络流量、优化网络资源来实现优化网络性能、保证服务质量、降低网络能耗的需求。

数据中心局域网具有网络节点密集、网络流量集中的特点,如何高效地利用带宽资源是主要的研究问题。同时,大型互联网服务提供商一般拥有多个地理位置跨度较大的数据中心,如何保证广域网数据中心之间高效互联并对外提供高质量、高可靠的服务,也是重要的工程问题。由于软件定义网络使用中心管控的架构,通过对链路中传输流量的状态信息进行测量,可以有效地指导控制器调整路由配置,均衡网络负载,提高网络链路利用率,非常适合数据中心的需求,因此得到了一定程度的应用。目前,大规模的软件定义网络测量实例包括 Microsoft 的 SWAN 系统和 Google 的 B4 系统。SWAN<sup>[74]</sup>(software-driven WAN)是 Microsoft 提出的跨数据中心 WAN 解决方案,通过为不同等级的网络服务设置相应的发送速率和网络路径来改善网络整体的资源利用率。SWAN 将网络流量按照优先级从高到低的顺序分为 3 类:交互流量、弹性流量和背景流量,中央控制器严格按照流量优先级顺序为每个服务分配相适应的链路资源。为了降低网络拥塞造成的影响,SWAN 会保留一部分最短路径的链路资源和交换机资源,在紧急情况时供最高优先级的服务使用。相较于单纯使用 MPLS 技术的跨数据中心 WAN,SWAN 可以将链路利用率从 60% 提高到 98%。Google 在其广域网数据中心中部署了基于 OpenFlow 技术的 B4<sup>[75]</sup>系统,该系统使用 Google 自行研发的专用 SDN 设备,用于 WAN 级的流量管控。WAN 网络在数以千计的独立链路上以总和 TB 每秒的数量级传输数据时会面临严重的性能和可靠性问题,在不考虑链路故障的情况下,一般链路利用率在 30%~40% 之间。B4 的核心是中央流量管控模块,该模块实时地监测全网的拓扑结构和可用资源信息,它将网站服务抽象为网络节点,网站与网站之间的连通性抽象为网络链路,网络流量抽象为以三元组(源网站,目的网站,QoS)表示的 Flow Group,然后通过基于哈希的 ECMP 算法动态地平衡流量。Google 表明,B4 系统可以使广域网链路的长期平均利用率保持在 70% 以上,部分链路利用率可达 99% 以上。

通过对流量大小进行感知和对流量类型进行识别,可以根据网络实时的负载情况调整路由策略,保证网络应用的服务质量。流量矩阵由网络中全部节点两两之间的流量大小构成,是反映网络状态的重要参数,文献[30,76,77]通过将相同源地址和目的地址的流统计信息合并来构造并维护全网的流量矩阵。MicroTE<sup>[36]</sup>周期性

地将瞬时流速与历史平均流速相比较,当流量大小显著变化时触发一次流量信息汇总,检测流量中是否包含大流:如果包含,则启动启发式的 ECMP 算法对网络全局路由进行优化.Suárez-Varela 等人<sup>[56,78,79]</sup>基于控制器 OpenDaylight 和软件交换机 openvswitch 实现了一个测量系统,对网络流量进行过滤、抽样,并使用深度包检测算法(deep packet inspection,简称 DPI)和机器学习(machine learning,简称 ML)算法对其中的 DNS 流量、HTTP 流量和加密流量进行识别分类.该系统在交换机中部署了三级流表:第 1 级为过滤规则,第 2 级为抽样规则,第 3 级为转发规则,然后统计待测流量的报文计数、流长计数和流持续时间信息.系统实现了两种抽样算法:一种是基于 IP 后缀的抽样,保证抽样报文来自不同的子网,抽样速率为  $1/(2m \cdot 2n)$ , $m$  是源 IP 后缀比特数, $n$  是目的 IP 后缀比特数;另一种是基于哈希的抽样,利用 OpenFlow 协议 1.3 中组表(action buckets)的功能对报文头五元组进行哈希,如果结果落在指定范围内则抽样.测量模块部署在控制器中,处理优先级设置为最高,当监听到控制器收到 Flow-Removed 消息时,触发控制器发送 Multipart-Request 消息获取交换机上的统计信息.

近年来,随着全人类节约能源和保护环境的意识不断提高,节约发展、清洁发展成为未来的主流发展模式.有研究表明:在发达国家中,数据和通信网络消耗的能源占社会总能源消耗的 5%,并且以每年 10%的速率增加.因此,研究如何降低网络的运行能耗是非常有意义的.软件定义网络由于可以从全局的视角根据网络中流量的分布状态动态地优化网络资源的分配,在一定程度上降低网络的能耗,引发了一些研究工作的关注.

Kaup 等人<sup>[80]</sup>提出了一个测量软件定义网络能耗的模型,它同时考虑硬件 OpenFlow 交换机和部署在服务器上的软件 OpenFlow 交换机的能耗,根据网络状态信息的测量结果,动态地将部分 OpenFlow 交换机调整至睡眠模式.实验结果表明:在 10Mbps 的组网环境下,该方法在硬件 OpenFlow 交换机和软件 OpenFlow 交换机上的能耗测量误差分别小于 1%和 8%,可以降低 6.6%网络总体能耗.文献<sup>[81,82]</sup>通过将多条链路的流量合并至一条链路,然后关闭其余链路相关的交换机端口来节约能耗.

### 3.3.2 安全防御

基于网络层面的安全防御研究工作通过对流量的统计特征进行分析,发现隐藏在正常流量中的攻击流量,然后通过定位攻击模式、追溯攻击源头、转移攻击目标、屏蔽攻击流量的方式来防御网络攻击,保障网络设备和主机设备的正常运行以及网络服务的正常访问.常用的安全相关的流量检测算法包括大流(heavy hitter)检测、分级大流(hierarchical heavy hitter)检测、变化(heavy change)检测、超点(superspreader)检测、DDoS 检测等,涵盖了主流的网络攻击手段.

OpenSample<sup>[38]</sup>提出,数据中心流量中超过 99%都是 TCP 流,通过计算同一个流中相邻两个采样报文中 TCP 数据包头部序号的差值,再除以报文各自对应的 sFlow 采样时间戳的差值,就可以计算出该 TCP 流在测量周期内的估计速率值.根据这一机制,OpenSample 实现了对大流的快速检测,并且可以实时地生成网络状态快照提供给上层应用使用.Openwatch<sup>[69]</sup>提出了一个自适应的流量计数方法,使用线性预测算法动态地调整异常流量检测的间隔,预测到攻击可能发生时进行细粒度测量,否则进行粗粒度的测量.Mehdi 等人<sup>[83]</sup>在软件定义测量架构中验证了多种异常检测算法,实现了端口速率限制、可疑报文过滤、异常分类等功能.iSTAMP<sup>[44]</sup>实现了网络中 Top- $k$  的大流检测,并且使用智能的 MAB(multi-armed bandit)算法估算全网的流长分布.vPROM-GUARD<sup>[42]</sup>在基于改造的 Open vSwitch 实现的测量架构中,对 DDoS 检测算法和端口扫描检测算法进行了验证.Moshref 等人<sup>[84]</sup>设计了 Max-Cover HHH 算法用于检测网络中的重要节点,并且在交换机数据平面直接部署了用于分级大流检测(hierarchical heavy hitter,简称 HHH)的哈希算法和 sketch 数据结构.HashPipe<sup>[85]</sup>在数据平面使用哈希算法对原始报文进行压缩处理,存储报文的标识符和计数值,用于跟踪网络中的大流(heavy hitter,简称 HH).Dream<sup>[70]</sup>和 Scream<sup>[59]</sup>设计了可以并发运行大流检测、分级大流检测、超点检测算法的测量架构,并且可以保证用户指定的精度.Univmon<sup>[63,86]</sup>基于一种测量结构实现了所有主流的流量检测算法,并达到较好的精度.

## 4 主要研究问题和挑战

表 1 总结了软件定义网络测量方法的相关研究成果.

Table 1 Summary of software defined measurement technology

表 1 软件定义网络测量方法总结

文献	测量架构	测量对象	研究特点
OpenTM <sup>[30]</sup>	基于 OpenFlow 协议查询	网络矩阵	选取部分交换机进行查询降低网络开销
MicroTE <sup>[36]</sup>	基于主机代理汇总流量	大流检测	网络全局路由优化
Flowsence <sup>[54]</sup>	监听 OpenFlow 异步消息	链路吞吐量	测量开销低,但是对于持续时间较长的流测量误差较大
SLAM <sup>[32]</sup>	主动下发测量流表规则	链路时延	支持网络任意路径的测量,但是测量误差受交换机本身性能和负载影响
PLANK <sup>[35]</sup>	基于端口镜像	链路吞吐量	适用于高速网络环境
OpenSample <sup>[38]</sup>	基于部署 sFlow 代理的 SDN 交换机	大流检测	低延迟,高处理速度
OpenNetMon <sup>[33]</sup>	主动下发测量流表规则	吞吐量,丢包率,时延	根据链路负载动态地调整查询频率
Payless <sup>[31]</sup>	基于 OpenFlow 协议查询	链路利用率	根据流量特征动态地调整查询频率
OFRewind <sup>[39]</sup>	基于控制器和交换机之间的代理	故障检测	通过记录与重放网络流量复现网络故障
Veriflow <sup>[40]</sup>	基于控制器和交换机之间的代理	逻辑一致性检测	通过图搜索技术结合网络拓扑来验证网络中的不变量
NetSight <sup>[41]</sup>	基于控制器和交换机之间的代理	报文路径检测	报文经过每一跳交换机时路径信息被封装为 Postcard 发送至测量服务器
CherryPick <sup>[73]</sup>	基于 OpenFlow 控制消息	报文路径检测	报文路径信息携带在报文头部,使用边缘着色算法对网络中非必要链路进行删减
SDNTraceroute <sup>[55]</sup>	基于 OpenFlow 控制消息	报文路径检测	报文经过每一跳交换机时路径信息被封装在 Packetin 消息中发送至控制器
Dai <sup>[52]</sup>	基于硬件可编程 SDN 交换机	报文路径检测	报文头部的哈希值作为 SRAM 索引,存放报文的路径信息
OpenWatch <sup>[69]</sup>	基于 OpenFlow 协议查询	异常检测	对流量变化进行预测,动态地调整测量粒度
Kaup 等人 <sup>[80]</sup>	基于 OpenFlow 协议查询	能耗测量	同时考虑硬件 SDN 交换机和软件 SDN 交换机的能耗,动态地休眠部分交换机降低能耗
iSTAMP <sup>[44]</sup>	基于硬件 TCAM 存储器	大流检测	将 TCAM 分为两部分,一部分用于流量的聚合,另一部分用于流量的解聚合
Wellem 等人 <sup>[47-51]</sup>	基于硬件可编程 SDN 交换机	超点检测,异常检测,大流检测	支持 10Gbps 线速下实时测量
Su 等人 <sup>[58,87,88]</sup>	基于硬件可编程 SDN 交换机	独立 IP 统计,流长统计	对测量任务建立数学模型,使用两级启发式算法进行优化资源调度
OpenSketch <sup>[45]</sup>	基于硬件可编程 SDN 交换机	大流检测,超点检测,DDoS 检测,异常检测,流数统计,流长分布统计	通用测量架构,数据平面抽象三级报文处理管道,部署多种哈希算法和 sketch 数据结构,测量库根据任务需求动态地分配和组合测量模块
DREAM <sup>[70]</sup> SCREAM <sup>[59]</sup>	基于商用 SDN 交换机和可编程 SDN 交换机	大流检测,分级大流检测,超点检测	支持并发测量,在时间和空间层面复用测量资源,动态地调整测量资源在不同测量任务之间的分配,保证全局测量精度的最优
UnivMon <sup>[63,86]</sup>	基于软件可编程 SDN 交换机	大流检测,DDoS 检测,异常检测,熵估计	基于一种 sketch 数据结构 Count sketch 实现了通用测量架构
Tahaei 等人 <sup>[89,90]</sup>	基于 OpenFlow 协议	通信开销,控制开销,同步开销	基于控制器完全分布式部署模式的网络流量协同测量架构
SketchVisor <sup>[60]</sup>	基于软件可编程 SDN 交换机	大流检测,异常检测,DDoS 检测,超点检测,流数统计,流长分布统计,熵估计	数据平面分离为普通路径和快速路径,快速路径通过牺牲一定的测量精度可以达到较高的处理速度,在数据中心网络中实现线速测量
Suárez-Varela 等人 <sup>[56,78,79]</sup>	基于软件可编程 SDN 交换机	流量识别,流量分类	基于 IP 后缀的抽样,基于哈希的抽样,使用 DPI 算法和 ML 算法对报文进行识别分类
Naveen 等人 <sup>[91]</sup>	基于软件可编程 SDN 交换机	XCP 协议,RCP 协议,CONGA 协议	通过近似技术对常用的资源优化调度算法进行改造,提高了报文处理速率并降低了硬件资源消耗
Gioacchino 等人 <sup>[92]</sup>	基于 OpenFlow 协议	链路负载	基于多控制器组网环境下的去中心化网络测量架构,实时地对网络负载进行监测和平衡

本节基于这些研究将软件定义网络测量的主要研究问题和挑战归纳为 4 个方面:测量资源与测量精度的

平衡、测量实时性与测量精度的平衡、高效通用的测量架构、多控制器下分布式协同测量。

#### 4.1 测量资源与测量精度的平衡

软件定义网络中的流量可以分为两种类型:控制流量和数据流量.控制流量是 SDN 控制器与 SDN 交换机之间传输的流量,数据流量是 SDN 交换机之间传输的流量.网络的正常运行离不开控制流量的正常传输,但是 SDN 测量任务往往需要在控制器和交换机之间进行额外的通信,或者在控制流量中携带额外的测量信息,这给有限的控制器和交换机之间的带宽资源带来了一定的压力,尤其是在控制器的带内(in band)部署的情况下. Ahmed 等人<sup>[93]</sup>讨论了控制器带外部署的组网设计,在这种情况下,协议控制信令通过专门的控制器到交换机链路传输,传输开销与查询的交换机数量和流的条数相关,与控制器部署的位置无关.而控制器带内部署的情况下,控制流量与数据流量共享带宽,控制通信开销的计算变得更加复杂,传输跳数也需要加入考虑.在软件定义网络中,为了保证网络的正常运行,控制平面和数据平面之间需要不断地交换 OpenFlow 消息.然而控制平面和数据平面之间的链路带宽是有限的,链路负载会随着网络规模的增加不断升高.如果测量方法过于频繁地利用控制器读取交换机中的计数器信息,会造成控制器与交换机之间的链路负载饱和,影响网络的稳定运行.此外,主动测量方法中使用较大规模的探测报文也有可能对网络中的生产流量造成一定影响.

同时,由于软件定义网络开放了控制平面的编程接口,控制器中部署的测量扩展模块会占用一定的控制器计算资源,可能对控制器性能造成影响.另外,SDN 交换机的流表空间是有限的,在流表中部署过多的测量规则可能导致对交换机硬件资源的过度消耗.OpenFlow 协议 1.0 版本中流表项匹配域的长度为 250 比特,在后续的更新中新字段不断加入,使得匹配域长度不断增加,远远超出了传统网络中的匹配域长度.过长的匹配域极大地增加了硬件的查表成本,尽管 OpenFlow 协议 1.1 版本发布了多级流表和流水线处理结构来压缩单张流表的大小,降低查表时的硬件资源消耗,但是也导致了 SDN 交换机硬件设计复杂度和报文转发时延的增加.

一些研究工作利用 SDN 交换机中的硬件存储器资源进行测量,使用 TCAM 存储器对报文进行高速地匹配和查找,使用 SRAM 存储器缓存预处理后的测量数据.在这些测量架构中,分配给测量任务的 TCAM 存储器大小和 SRAM 存储器大小与测量精度直接相关,然而交换机上的 TCAM 资源和 SRAM 资源是非常有限的,并发执行多个测量任务时所需的资源可能超出硬件的容量.

综上所述,在软件定义网络中部署额外的测量机制需要消耗额外的网络资源,包括链路的带宽资源、SDN 控制器和 SDN 交换机的计算和存储资源,可能对网络的运行性能造成一定影响.因此,如何在保证合理的测量精度的前提下尽可能地降低对网络资源的消耗,是软件定义网络测量的重要研究问题.基于查询的测量方法一般通过动态地调整查询频率或减少查询的交换机数量来减少带宽资源的开销.OpenTM<sup>[30]</sup>通过在流量转发路径中选取部分交换机进行查询的方式来降低网络通信的开销,并使用多种选择算法对测量精度进行了验证. Payless<sup>[31]</sup>为降低网络开销采用了自适应调度算法,根据流量特征,动态地调整查询交换机的频率.与普通轮询算法每秒 13.5 个查询请求相比较,PayLess 每秒只需发送 6.6 个查询请求. Narayana 等人<sup>[94]</sup>设计了一个自定义查询语言,实现了自适应的数据查询,降低了查询造成的开销. SDN-Mon<sup>[95]</sup>在控制器中部署 GMT(global monitoring table,全局测量表)对全网交换机的测量统计信息和实时负载情况进行汇总,检测多个交换机对同一条流进行冗余测量的情况,选择一个负载最低的交换机对该流进行测量,删除其他交换机中的测量规则.一些工作研究如何在不影响测量精度的前提下对流表中的测量流表项数量进行压缩. Yu 等人<sup>[96]</sup>对流表项的压缩机制进行了深入的研究,他们通过设置超时时间、限制匹配域长度、合并相似规则等方式对流表进行压缩,压缩导致的错误流表项数量小于 1%. DevoFlow<sup>[97]</sup>通过移除不重要的流表项来减少对流表资源的占用. Hone<sup>[37]</sup>使用被称为 Lazy Materialization 的方法将流量的统计特征数据压缩为类似数据库表的格式.一些工作研究在网络中并发执行多类型多数量的测量任务时,如何在时间层面和空间层面对测量资源进行复用,并动态地在测量任务之间调度和分配,保证全局测量精度的最优化. Dream<sup>[70]</sup>和 Scream<sup>[59]</sup>发现:测量架构可以在空间层面将大流分解,然后通过地址前缀和端口号聚集小的流量;在时间层面,可以在不同的测量周期内采集数据.两篇文献分别从商用 SDN 交换机和可编程 SDN 交换机的角度对该问题进行了研究,设计了一个高效复用网络资源的测量架构.该架构可以动态地调整测量资源在不同测量任务之间的分配,保证测量结果的精度在用户指定的范围之内.该研究表明:



分配给测量任务的资源大小与测量精度之间存在一个收益递减点,当精度超过一定阈值后,继续增加测量资源得到的收益极低.此外,测量任务仅需要在有与规则匹配的流经过的交换机上分配资源,并且所需资源的多少由流量的大小和分布所决定.因此,可以通过全局的测量资源的调度实现最大化的资源利用率和最高的全局精度.测量系统由任务实体、精度估计器和资源分配器组成,工作流程如下:当新的测量任务被下发时,系统首先根据目前可用的资源决定是否接受该任务,如果接受该任务会导致正在运行的测量任务的精度低于期望值,该任务会被丢弃.一旦任务被接受,系统会为新任务创建一个任务实体,并在交换机上分配默认数量的资源.系统周期性地从交换机中获取测量结果并传递给任务实体,每个任务实体包含一个精度估计器来估算当前的任务精度.系统为每种测量算法(HH,HHH,SS)都设计了精度估计算法,结果传递给资源分配器,资源分配器根据结果决定分配给该任务的资源数量.任务实体获取测量资源后,决定如何分配这些资源.如果一个测量任务被终止,系统会删除其相关任务实体并且释放所有相关的资源.MOZART<sup>[98]</sup>在 Dream<sup>[70]</sup>和 Scream<sup>[59]</sup>的基础之上对 SDN 交换机之间时间层面上的资源复用进行了更深入的研究.Su 等人<sup>[58,87,88]</sup>对动态分配测量任务、优化调度测量资源的测量架构进行了研究,测量架构包含 4 个层面,自上而下分别是应用层、资源调度层、任务分配层和数据平面层,其中,数据平面层使用和 OpenSketch<sup>[45]</sup>相同的设计.该研究对测量任务建立数学模型,将交换机、链路、每个交换机的 TCAM 资源、每个交换机的 SRAM 资源、测量任务集、运行测量任务的交换机集、分配给任务的 TCAM 资源、分配给任务的 SRAM 资源、测量任务的理论精度、实际精度、权重完全参数化,然后使用两级启发式算法进行计算.资源调度层估算测量任务所需资源,定位资源位置,压缩资源占用;任务分配层定位交换机位置,重新分配交换机测量资源.

## 4.2 测量实时性与测量精度的平衡

软件定义网络测量支持流量的实时分析和反馈,控制器根据测量结果迅速调整和优化网络配置,满足网络服务的动态需求.但是由于测量机制和算法具有一定复杂度,对流量处理需要消耗一定的时间,因此在对较大规模流量进行线速测量时,需要在测量实时性和测量精度之间进行平衡.Omid 等人<sup>[99]</sup>对常用测量算法消耗的计算资源、存储资源与测量精度之间的关系进行了对比分析,提出:在内存资源充足的情况下(如使用软件 SDN 交换机),测量任务面临的主要挑战是测量速率与测量精度的矛盾.该研究表明:在进行高速测量时,使用线性哈希表、数组等简单数据结构的性能甚至优于 Cukoo 哈希、Count-Min sketch 等复杂的测量算法.SketchVisor<sup>[60]</sup>通过改造虚拟 SDN 交换机 Open vSwitch 的数据平面,并结合数据流技术,实现了数据中心网络中流量的高速测量.该研究指出:硬件 SDN 交换机由于改造 ASIC 成本较高,难以在生产环境大量应用,因此,目前数据中心主要使用基于服务器部署实现的软件 SDN 交换机.但是虚拟交换机在进行基于 sketch 数据结构的哈希计算时会消耗大量的 CPU 资源,难以在数据中心生产环境中进行线速地测量.因此,SketchVisor 提出了一种解决方法,将数据平面进行再分离,在普通路径之外独立出一条快速流量处理路径,快速路径通过牺牲一定的测量精度可以达到较高的处理速度.正常情况下,所有报文被转发至普通路径,通过一个 FIFO 缓存,当缓存溢出时,额外的流量被重定向至快速路径.快速路径中部署了一种新型的 top-k 算法对大流进行捕捉,同时,使用一个全局计数器对小流的统计特征进行计数.该算法将 Misra-Gries's top-k 算法和 Probabilistic Lossy Counting 算法相结合,可以一次性剔除多条小流,降低了算法的复杂度.普通路径具有通用的结构,可以部署用户指定的任意 sketch.控制器通过 ZeroMQ 采集数据平面的测量数据,并使用矩阵叠加的方式合并普通路径和快速路径的 sketch 数据.实验中,单 CPU 的吞吐量达到了 17Gbps,并且快速路径只占用若干 kb 大小的内存.Naveen 等人<sup>[91]</sup>认为:随着交换机技术的进步,在硬件可编程 SDN 交换机的软件定义数据平面中可以灵活地部署对每个单独报文的 MA(match and action)处理结构,但是报文处理速度受到有限的硬件状态和硬件操作的限制.该研究通过近似技术对常用的资源优化调度算法进行改造,提高了报文处理速率并降低了硬件资源消耗,并在基于 FlexSwitch 实现的原型系统中对 XCP,RCP,CONGA 等拥塞控制协议和负载均衡协议进行了验证.

## 4.3 高效通用的测量架构

当前,网络中部署的应用服务数量繁多、类型丰富,需要测量架构支持多数量、多种类测量任务的并发执

行.矛盾的是:用于流量分析的数据流算法所使用的 sketch 数据结构虽然具备较好的内存与精度的平衡性,但是与测量对象的耦合性很强,可扩展性非常有限,一种 sketch 数据结构难以支持不同类型的测度.因此,如何通过部署额外的测量机制来加强测量架构的通用性,是软件定义网络测量中重要的研究问题.

P4<sup>[100]</sup>和 ClickNP<sup>[101]</sup>分别从软件抽象和硬件抽象的角度研究了通用可编程 SDN 交换机的设计与实现.OpenSketch<sup>[45]</sup>是基于可编程 SDN 交换机设计的通用测量架构,它充分利用了 SDN 测量控制平面和测量数据平面分离的优势,支持用户自定义测量任务的并发执行.为了使用一种硬件结构支持多类型的测量任务,OpenSketch 将测量数据平面的报文处理流程抽象为哈希-分类-计数这 3 个步骤,这种抽象降低了数据平面功能之间的耦合性.OpenSketch 部署了多种哈希算法对报文进行处理,使用 TCAM 存储器对报文进行分类,并将统计结果存储在 SRAM 存储器中.交换机 SRAM 被划分为若干个逻辑区域,每个区域分别被设计为不同类型和大小的 sketch 数据结构,以支持不同类型的测量任务.在测量控制平面,OpenSketch 提供了一个测量库,用户可以调用编程接口来下发测量任务,OpenSketch 根据测量任务的测量类型和精度需求自动地选择哈希函数和 sketch 数据结构配置数据平面管道,并且分配合适的交换机资源.OpenSketch 三级数据管道的设计很巧妙,把成本高但速度快的 TCAM 存储器用作流量分类,速度慢但成本低的 SRAM 存储器用于流量存储,在保证处理速度的同时控制了硬件成本.实验表明:其支持多种常用类型测量任务的并发执行,并且保证了较高的精度.但是 OpenSketch 对数据平面修改的幅度较大,其通用性是以牺牲硬件资源为代价,在执行测量时未参与计算的 sketch 数据结构浪费了其占有的存储空间.Univmon<sup>[63,86]</sup>通过理论研究,基于 Count sketch 一种数据结构设计了通用的测量架构,支持多种测度的测量并具有较好的精度.该方法将原始流量通过哈希抽样分解为  $N$  条子流,并行地通过  $N$  个 Count sketch 进行计算,计算结果以递归的方式合并.Univmon 虽然在数据平面只使用了一种 sketch 数据机构,但是需要  $N$  个独立的存储空间,同样是以牺牲存储资源为代价换取通用性.

#### 4.4 多控制器分布式协同测量

在较大规模的软件定义网络中,单一控制器的部署方式由于处理能力有限会造成网络的性能瓶颈,因此,多控制器分布式部署是软件定义网络未来的主流部署方式.目前,关于多控制器组网的研究内容主要包括控制器的部署数量、控制器的部署位置以及控制器之间的交互方式,已发表的分布式控制器部署模式可以分为 3 种:第 1 种是主流控制器包括 Onix,Hyperflow,ONOS,OpenDayLight 均采用的物理分布、逻辑集中的部署模式,在这种模式中,控制器之间需要不断地同步信息,以保证网络逻辑视图的一致性,在网络状态变化频繁的情况下,会造成大量的网络开销;第 2 种是基于两层控制逻辑的部署模式<sup>[102]</sup>,该模式中,控制器分为根控制器和从控制器,从控制器负责各自区域内的网络管理并将区域网络视图与主控制器进行同步,主控制器掌管全局的网络视图;第 3 种是集群主备模式,该模式中,网络的控制逻辑完全分布,根据 OpenFlow 协议 1.3 版本规定,一个交换机同时只能被一个控制器控制,因此,该模式下控制器需要双机互备,在运控制器故障时,由备份控制器进行替补.

多控制器部署可以解决软件定义网络控制平面的性能瓶颈问题,但是在这种组网模式中,设计测量架构需要考虑更复杂的测量机制,包括测量任务的调度、测量配置的同步、测量资源的分配、测量结果的合并等.MDCP<sup>[103]</sup>对分布式控制器组网环境下控制器放置位置对测量造成的影响进行了研究.Tahaei 等人<sup>[89,90]</sup>设计了在控制器完全分布式部署的模式下进行网络流量协同测量的架构,该架构分为 3 层:网络层、控制器层和协作层.网络层包含所有的软件和硬件交换设备;协作层向控制器层中的本地控制器下发测量流表规则;每个控制器被指定对若干条流量进行测量.控制器包含流跟踪器、组表生成器、查询生成器和采集器,以 1s 为测量周期查询数据平面的测量数据,并聚集后通过 UDP 报文的方式发送至协作层.协作层通过 UDP 监听器收集测量数据并整合形成流量矩阵,还提供东西向接口用于控制器之间的交互.该研究分析了分布式控制器组网情况下的控制器开销和通信开销,并讨论了相应的优化手段.Gioacchino 等人<sup>[92]</sup>设计了大规模多控制器组网环境下的去中心化网络测量架构,实时地对网络负载进行监测和平衡.该架构在网络中的每个控制器内部署了一个本地管理器(local manager),本地管理器包含 4 个模块:持久化数据存储器负责保存网络拓扑和链路状态信息;需求解析器负责将上层应用的需求解析为流表规则;任务调度器负责建立和管理每个测量任务线程;结果处理器负责对原始流量进行抽样哈希,以哈希表的形式保存统计信息.本地管理器周期性地测量负责网络区域内的平均链路利用



率和平均流速,并和相邻本地管理器进行同步,当发生链路拥塞时,负责该链路的本地管理器首先将多余流量调度至空闲链路,再由相邻的本地管理器进行迭代的调度.实验结果表明,该分布式控制器测量架构造成的控制通信开销是中心化控制器测量架构的4倍.

## 5 结束语和未来展望

软件定义网络给网络测量技术研究注入了新的活力,借助于软件定义网络多方面的优势,很多在传统网络中难以部署的测量架构和测量算法在软件定义网络中得到了实现和验证.本文从测量架构和测量对象两方面对近年来主流的研究工作进行了归纳分析,其中,测量架构分为流量的采集和预处理、流量的传输和存储、流量的分析和反馈这3个步骤进行阐述,测量对象分为运行状态、转发性能、流量特征这3种类型进行说明.最后总结了主要的研究问题和挑战,包括测量资源与测量精度的平衡、测量实时性与测量精度的平衡、高效通用的测量架构、多控制器分布式协同测量这4个方面.在现有成果的基础之上,软件定义网络测量研究中有些领域值得深入地探索,未来的研究方向主要基于以下几个方面.

- 1) 更强大的测量算法.软件定义网络测量技术离不开流量压缩、流量分析等测量算法的支持,优秀的测量算法可以在保证测量精度的前提下,显著地降低对网络中带宽资源、计算资源、存储资源的消耗,从根本上提高测量架构的性能.因此在未来的工作中,需要对抽样技术和数据流技术中的基础算法开展进一步的研究,降低算法的复杂度,提高算法的通用性.Univmon<sup>[63,86]</sup>虽然牺牲了一定的存储资源,但是大大提高了 sketch 数据结构的通用性,并且达到了较好的执行效率,未来需要在目前的研究基础上探索更加高效的哈希算法和更加通用的 sketch 数据结构;
- 2) 更智能的测量机制.未来的软件定义网络测量技术应该充分利用控制平面掌握网络全局信息的优势,根据网络运行状态的变化,在全网范围内动态地调度和复用测量资源,提高测量资源的利用率.在数据平面需要部署更加智能的流表管理机制,根据流量特征对测量流表项数量进行压缩,节约宝贵的流表资源.在控制平面需要部署更加智能的流量查询机制,根据网络负载动态地调整查询的对象和频率,减少控制平面和数据平面之间的通信流量,降低对网络性能的影响.Su 等人<sup>[58,87,88]</sup>所提出的网络参数化模型和两级启发式算法提供了一个测量机制的建模思路,值得进一步的探索;
- 3) 云服务环境中的应用感知测量.云服务环境中部署的网络应用类型丰富,对网络性能保障的需求各不相同.实时性需求较高的应用对延迟非常敏感,例如:无线移动服务需要实时跟踪用户信道的切换;远程会议服务需要全程的 QoS 保障;而网络直播服务则需要灵活的带宽保障,在流行节目播出时,面对短时间内大量用户的涌入,需要及时地增加带宽容量保证视频流量的高效分发,直播结束后,需要及时地释放带宽资源.软件定义网络测量由于支持测量数据的实时分析和反馈,并且根据测量需求的变化动态地调整测量配置,因此在云服务环境中可以针对不同类型的应用服务提供定制化的感知测量.这要求测量算法能够快速预测网络资源和应用需求的未来变化,并整合为具体的网络配置,指导控制器进行调整;
- 4) 网络空间安全领域的支撑.近年来,随着大规模网络安全事件的不断爆发,网络空间安全研究工作的重要性不断提高.现有的基于网络攻击层面的安全防御手段主要通过动态地调整主机地址、网络路由、应用端口的方式转移受攻击的网络对象,同时将攻击流量引导至虚假的目标,切断后续的攻击流量.由于软件定义网络测量架构具备流量采集、流量分析、异常检测、配置调整的闭环测量过程,能够做到威胁检测、配置调整、攻击面转移的自动化,有效地支撑网络空间安全的主动防御技术;
- 5) 多自治域的域间测量.目前,软件定义网络测量的相关研究工作主要集中于单一自治域的域内测量,没有涉及到多个自治域的域间测量,虽然有文献研究了多控制器下的分布式协同测量,但仍然在同个自治域内.在多个自治域间进行测量时,需要消除传输抖动对测量过程造成的影响,这对测量机制和测量架构提出了更高的要求.

**References:**

- [1] Zhou AP, Cheng G, Guo XJ. High-speed network traffic measurement method. *Ruan Jian Xue Bao/Journal of Software*, 2014,25(1): 135–153 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4445.htm> [doi: 10.13328/j.cnki.jos.004445]
- [2] Lee S, Levanti K, Kim HS. Network monitoring: Present and future. *Computer Networks*, 2014,65:84798.
- [3] Zuo QY, Chen M, Zhao GS, *et al.* Research on OpenFlow-based SDN technologies. *Ruan Jian Xue Bao/ Journal of Software*, 2013,24(5):1078–1097 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4390.htm> [doi: 10.3724/SP.J.1001.2013.04390]
- [4] Zhang CK, Cui Y, Tang HY, *et al.* State-of-the-art survey on software-defined networking (SDN). *Ruan Jian Xue Bao/Journal of Software*, 2015,26(1):62–81 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4701.htm> [doi: 10.13328/j.cnki.jos.004701]
- [5] Nunes BAA, Mendonca M, Nguyen XN, *et al.* A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 2014,16(3):1617–1634.
- [6] Hu F, Hao Q, Bao K. A survey on software-defined network and openflow: From concept to implementation. *IEEE Communications Surveys & Tutorials*, 2014,16(4):2181–2206.
- [7] Sezer S, Scott-Hayward S, Chouhan PK, *et al.* Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine*, 2013,51(7):36–43.
- [8] Nunes BAA, Mendonca M, Nguyen XN, *et al.* A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 2014,16(3):1617–1634.
- [9] Koponen T, Casado M, Gude N, *et al.* Onix: A distributed control platform for large-scale production networks. In: *Proc. of the OSDI*. 2010. 1–6.
- [10] Tootoonchian A, Ganjali Y. Hyperflow: A distributed control plane for openflow. In: *Proc. of the 2010 Internet Network Management Conf. on Research on Enterprise Networking*. 2010.
- [11] Berde P, Gerola M, Hart J, *et al.* ONOS: Towards an open, distributed SDN OS. In: *Proc. of the 3rd Workshop on Hot Topics in Software Defined Networking*. ACM Press, 2014. 1–6.
- [12] Medved J, Varga R, Tkacik A, *et al.* OpenDaylight: Towards a model-driven sdn controller architecture. In: *Proc. of the 2014 IEEE 15th Int'l Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2014. 1–6.
- [13] Jammal M, Singh T, Shami A, *et al.* Software defined networking: State of the art and research challenges. *Computer Networks*, 2014,72:74–98.
- [14] Yang M, Li Y, Jin D, *et al.* Software-defined and virtualized future mobile and wireless networks: A survey. *Mobile Networks and Applications*, 2015,20(1):4–18.
- [15] Luo T, Tan HP, Quek TQS. Sensor OpenFlow: Enabling software-defined wireless sensor networks. *IEEE Communications Letters*, 2012,16(11):1896–1899.
- [16] Qin Z, Denker G, Giannelli C, *et al.* A software defined networking architecture for the internet-of-things. In: *Proc. of the 2014 IEEE Network Operations and Management Symp. (NOMS)*. IEEE, 2014. 1–9.
- [17] Kreutz D, Ramos FMV, Verissimo PE, *et al.* Software-defined networking: A comprehensive survey. *Proc. of the IEEE*, 2015, 103(1):14–76.
- [18] Farhady H, Lee HY, Nakao A. Software-defined networking: A survey. *Computer Networks*, 2015,81:79–95.
- [19] Yassine A, Rahimi H, Shirmohammadi S. Software defined network traffic measurement: Current trends and challenges. *IEEE Instrumentation & Measurement Magazine*, 2015,18(2):42–50.
- [20] Zhou TQ, Cai ZP, Xia J, *et al.* Traffic engineering for software defined networks. *Ruan Jian Xue Bao/Journal of Software*, 2016,27(2):394–417 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5020.htm> [doi: 10.13328/j.cnki.jos.005020]
- [21] Wang MM, Liu JW, Chen J, *et al.* Software defined networking: Security model, threats and mechanism. *Ruan Jian Xue Bao/Journal of Software*, 2016,27(4):969–992 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/27/969.htm> [doi: 10.13328/j.cnki.jos.005020]
- [22] Agarwal S, Kodialam M, Lakshman TV. Traffic engineering in software defined networks. In: *Proc. of the 2013 IEEE INFOCOM*. IEEE, 2013. 2211–2219.
- [23] Akyildiz IF, Lee A, Wang P, *et al.* A roadmap for traffic engineering in SDN-OpenFlow networks. *Computer Networks*, 2014,71: 1–30.

- [24] Shu Z, Wan J, Lin J, *et al.* Traffic engineering in software-defined networking: Measurement and management. *IEEE Access*, 2016, 4:3246–3256.
- [25] Akyildiz IF, Lee A, Wang P, *et al.* Research challenges for traffic engineering in software defined networks. *IEEE Network*, 2016, 30(3):52–58.
- [26] Mendiola A, Astorga J, Jacob E, *et al.* A survey on the contributions of software-defined networking to traffic engineering. *IEEE Communications Surveys & Tutorials*, 2016,19(2):918–953.
- [27] Rawat DB, Reddy SR. Software defined networking architecture, security and energy efficiency: A survey. *IEEE Communications Surveys & Tutorials*, 2017,19(1):325–346.
- [28] Sridharan V, Gurusamy M, Truong-Huu T. Multi-controller traffic engineering in software defined networks. In: *Proc. of the 2017 IEEE 42nd Conf. on Local Computer Networks (LCN)*. IEEE, 2017. 137–145.
- [29] Mohan V, Reddy YRJ, Kalpana K. Active and passive network measurements: A survey. *Int'l Journal of Computer Science and Information Technologies*, 2011,2(4):1372–1385.
- [30] Tootoonchian A, Ghobadi M, Ganjali Y. OpenTM: Traffic matrix estimator for OpenFlow networks. In: *Proc. of the Int'l Conf. on Passive and Active Network Measurement*. Berlin, Heidelberg: Springer-Verlag, 2010. 201–210.
- [31] Chowdhury SR, Bari MF, Ahmed R, *et al.* Payless: A low cost network monitoring framework for software defined networks. In: *Proc. of the 2014 IEEE Network Operations and Management Symp. (NOMS)*. IEEE, 2014. 1–9.
- [32] Yu C, Lumezanu C, Sharma A, *et al.* Software-defined latency monitoring in data center networks. In: *Proc. of the Int'l Conf. on Passive and Active Network Measurement*. Cham: Springer-Verlag, 2015. 360–372.
- [33] Van Adrichem NLM, Doerr C, Kuipers FA. Opennetmon: Network monitoring in openflow software-defined networks. In: *Proc. of the 2014 IEEE Network Operations and Management Symp. (NOMS)*. IEEE, 2014. 1–8.
- [34] Liu C, Malboubi AM, Chuah CN. OpenMeasure: Adaptive flow measurement & inference with online learning in SDN. In: *Proc. of the 2016 IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2016. 47–52.
- [35] Rasley J, Stephens B, Dixon C, *et al.* Planck: Millisecond-scale monitoring and control for commodity networks. *ACM SIGCOMM Computer Communication Review*, 2014,44(4):407–418.
- [36] Benson T, Anand A, Akella A, *et al.* MicroTE: Fine grained traffic engineering for data centers. In: *Proc. of the 7th Conf. on Emerging Networking EXperiments and Technologies*. ACM Press, 2011.
- [37] Sun P, Yu M, Freedman MJ, *et al.* Hone: Joint host-network traffic management in software-defined networks. *Journal of Network and Systems Management*, 2015,23(2):374–399.
- [38] Suh J, Kwon TT, Dixon C, *et al.* Opensample: A low-latency, sampling-based measurement platform for commodity SDN. In: *Proc. of the 2014 IEEE 34th Int'l Conf. on Distributed Computing Systems (ICDCS)*. IEEE, 2014. 228–237.
- [39] Wundsam A, Levin D, Seetharaman S, *et al.* OFRewind: Enabling record and replay troubleshooting for networks. In: *Proc. of the USENIX Annual Technical Conf.* 2011. 15–17.
- [40] Khurshid A, Zhou W, Caesar M, *et al.* Veriflow: Verifying network-wide invariants in real time. In: *Proc. of the 1st Workshop on Hot Topics in Software Defined Networks*. ACM Press, 2012. 49–54.
- [41] Handigol N, Heller B, Jeyakumar V, *et al.* I know what your packet did last hop: Using packet histories to troubleshoot networks. In: *Proc. of the NSDI, Vol.14*. 2014. 71–85.
- [42] Wang A, Guo Y, Chen S, *et al.* vPROM: vSwitch enhanced programmable measurement in SDN. In: *Proc. of the 2017 IEEE 25th Int'l Conf. on Network Protocols (ICNP)*. IEEE, 2017. 1–10.
- [43] Wang A, Guo Y, Hao F, *et al.* Umon: Flexible and fine grained traffic monitoring in open vswitch. In: *Proc. of the 11th ACM Conf. on Emerging Networking Experiments and Technologies*. ACM Press, 2015.
- [44] Malboubi M, Wang L, Chuah CN, *et al.* Intelligent sdn based traffic (de) aggregation and measurement paradigm (istamp). In: *Proc. of the 2014 IEEE INFOCOM*. IEEE, 2014. 934–942.
- [45] Yu M, Jose L, Miao R. Software defined traffic measurement with OpenSketch. In: *Proc. of the NSDI, Vol.13*. 2013. 29–42.
- [46] Yu Y, Qian C, Li X. Distributed and collaborative traffic monitoring in software defined networks. In: *Proc. of the 3rd Workshop on Hot Topics in Software Defined Networking*. ACM Press, 2014. 85–90.
- [47] Wellem T, Li GW, Lai YK. Superspreader detection system on NetFPGA platform. In: *Proc. of the 10th ACM/IEEE Symp. on Architectures for Networking and Communications Systems*. ACM Press, 2014. 247–248.
- [48] Lai YK, Lee CC, Huang BH, *et al.* Real-time detection of changes in network with OpenFlow based on NetFPGA implementation. *Microprocessors and Microsystems*, 2014,38(5):431–442.

- [49] Wellem T, Lai YK, Chung WY. A software defined sketch system for traffic monitoring. In: Proc. of the 2015 ACM/IEEE Symp. on Architectures for Networking and Communications Systems (ANCS). IEEE, 2015. 197–198.
- [50] Wellem T, Lai YK, Huang CY, *et al.* A hardware-accelerated infrastructure for flexible sketch-based network traffic monitoring. In: Proc. of the High Performance Switching and Routing (HPSR). 2016. 162–167.
- [51] Wellem T, Lai YK, Cheng CH, *et al.* Implementing a heavy hitter detection on the NetFPGA OpenFlow switch. In: Proc. of the 2017 IEEE Int'l Symp. on Local and Metropolitan Area Networks (LANMAN). IEEE, 2017. 1–2.
- [52] Dai M, Cheng G, Wang Y. Detecting network topology and packet trajectory with SDN-enabled FPGA platform. In: Proc. of the 11th Int'l Conf. on Future Internet Technologies. ACM Press, 2016. 7–13.
- [53] Su Z, Wang T, Xia Y, *et al.* CeMon: A cost-effective flow monitoring system in software defined networks. *Computer Networks*, 2015,92:101–115.
- [54] Yu C, Lumezanu C, Zhang Y, *et al.* Flowsense: Monitoring network utilization with zero measurement cost. In: Proc. of the Int'l Conf. on Passive and Active Network Measurement. Berlin, Heidelberg: Springer-Verlag, 2013. 31–41.
- [55] Agarwal K, Rozner E, Dixon C, *et al.* SDN traceroute: Tracing SDN forwarding without changing network behavior. In: Proc. of the 3rd Workshop on Hot Topics in Software Defined Networking. ACM Press, 2014. 145–150.
- [56] Suárez-Varela J, Barlet-Ros P. Flow monitoring in software-defined networks: Finding the accuracy/performance tradeoffs. *Computer Networks*, 2018,135:289–301.
- [57] Wang B, Su J, Li J, *et al.* EffiView: Trigger-based monitoring approach with low cost in SDN. In: Proc. of the 2017 IEEE 19th Int'l Conf. on High Performance Computing and Communications, IEEE 15th Int'l Conf. on Smart City, IEEE 3rd Int'l Conf. on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2017. 309–315.
- [58] Su Z, Wang T, Hamdi M. JOTA: Joint optimization for the task assignment of sketch-based measurement. *Computer Communications*, 2017,102:17–27.
- [59] Moshref M, Yu M, Govindan R, *et al.* Scream: Sketch resource allocation for software-defined measurement. In: Proc. of the 11th ACM Conf. on Emerging Networking Experiments and Technologies. ACM Press, 2015.
- [60] Huang Q, Jin X, Lee PPC, *et al.* SketchVisor: Robust network measurement for software packet processing. In: Proc. of the Conf. of the ACM Special Interest Group on Data Communication. ACM Press, 2017. 113–126.
- [61] Li Y, Miao R, Kim C, *et al.* FlowRadar: A better NetFlow for data centers. In: Proc. of the NSDI. 2016. 311–324.
- [62] Schweller R, Li Z, Chen Y, *et al.* Reversible sketches: Enabling monitoring and analysis over high-speed data streams. *IEEE/ACM Trans. on Networking (ToN)*, 2007,15(5):1059–1072.
- [63] Liu Z, Manousis A, Vorsanger G, *et al.* One sketch to rule them all: Rethinking network flow monitoring with univmon. In: Proc. of the 2016 ACM SIGCOMM Conf. ACM Press, 2016. 101–114.
- [64] [Cormode G, Muthukrishnan S. What's new: Finding significant differences in network data streams. *IEEE/ACM Trans. on Networking (TON)*, 2005,13(6):1219–1232.
- [65] Flajolet P, Martin GN. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 1985,31(2):182–209.
- [66] Bar-Yossef Z, Jayram TS, Kumar R, *et al.* Counting distinct elements in a data stream. In: Proc. of the Int'l Workshop on Randomization and Approximation Techniques in Computer Science. Berlin, Heidelberg: Springer-Verlag, 2002. 1–10.
- [67] Whang KY, Vander-Zanden BT, Taylor HM. A linear-time probabilistic counting algorithm for database applications. *ACM Trans. on Database Systems (TODS)*, 1990,15(2):208–229.
- [68] Kumar A, Sung M, Xu JJ, *et al.* Data streaming algorithms for efficient and accurate estimation of flow size distribution. *ACM SIGMETRICS Performance Evaluation Review*, 2004,32(1):177–188.
- [69] Zhang Y. An adaptive flow counting method for anomaly detection in SDN. In: Proc. of the 9th ACM Conf. on Emerging Networking Experiments and Technologies. ACM Press, 2013. 25–30.
- [70] Moshref M, Yu M, Govindan R, *et al.* DREAM: Dynamic resource allocation for software-defined measurement. *ACM SIGCOMM Computer Communication Review*, 2015,44(4):419–430.
- [71] Zeng H, Kazemian P, Varghese G, *et al.* Automatic test packet generation. In: Proc. of the 8th Int'l Conf. on Emerging Networking Experiments and Technologies. ACM Press, 2012. 241–252.
- [72] Kempf J, Bellagamba E, Kern A, *et al.* Scalable fault management for OpenFlow. In: Proc. of the 2012 IEEE Int'l Conf. on Communications (ICC). IEEE, 2012. 6606–6610.
- [73] Tammana P, Agarwal R, Lee M. Cherrypick: Tracing packet trajectory in software-defined datacenter networks. In: Proc. of the 1st ACM SIGCOMM Symp. on Software Defined Networking Research. ACM Press, 2015.

- [74] Hong CY, Kandula S, Mahajan R, *et al.* Achieving high utilization with software-driven WAN. *ACM SIGCOMM Computer Communication Review*, 2013,43(4):15–26.
- [75] Jain S, Kumar A, Mandal S, *et al.* B4: Experience with a globally-deployed software defined WAN. *ACM SIGCOMM Computer Communication Review*, 2013,43(4):3–14.
- [76] Gong Y, Wang X, Malboubi M, *et al.* Towards accurate online traffic matrix estimation in software-defined networks. In: *Proc. of the 1st ACM SIGCOMM Symp. on Software Defined Networking Research*. ACM Press, 2015.
- [77] Malboubi M, Gong Y, Yang Z, *et al.* Software defined network inference with evolutionary optimal observation matrices. *Computer Networks*, 2017,129:93–104.
- [78] Suárez-Varela J, Barlet-Ros P. Towards a NetFlow implementation for OpenFlow software-defined networks. In: *Proc. of the 2017 29th Int'l Teletraffic Congress (ITC 29)*. IEEE, 2017. 187–195.
- [79] Suárez-Varela J, Barlet-Ros P. SBAR: SDN flow-based monitoring and application recognition. In: *Proc. of the Symp. on SDN Research*. ACM Press, 2018..
- [80] Kaup F, Melnikowitsch S, Hausheer D. Measuring and modeling the power consumption of openflow switches. In: *Proc. of the 2014 10th Int'l Conf. on Network and Service Management (CNSM)*. IEEE, 2014. 181–186.
- [81] Chanda A, Westphal C, Raychaudhuri D. Content based traffic engineering in software defined information centric networks. In: *Proc. of the 2013 IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2013. 357–362.
- [82] Ruiz-Rivera A, Chin KW, Soh S. GreCo: An energy aware controller association algorithm for software defined networks. *IEEE Communications Letters*, 2015,19(4):541–544.
- [83] Mehdi SA, Khalid J, Khayam SA. Revisiting traffic anomaly detection using software defined networking. In: *Proc. of the Int'l Workshop on Recent Advances in Intrusion Detection*. Berlin, Heidelberg: Springer-Verlag, 2011. 161–180.
- [84] Moshref M, Yu M, Govindan R. Resource/Accuracy tradeoffs in software-defined measurement. In: *Proc. of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. ACM Press, 2013. 73–78.
- [85] Sivaraman V, Narayana S, Rottenstreich O, *et al.* Heavy-hitter detection entirely in the data plane. In: *Proc. of the Symp. on SDN Research*. ACM Press, 2017. 164–176.
- [86] Liu Z, Vorsanger G, Braverman V, *et al.* Enabling a RISC approach for software-defined monitoring using universal streaming. In: *Proc. of the 14th ACM Workshop on Hot Topics in Networks*. ACM Press, 2015.
- [87] Su Z, Wang T, Xia Y, *et al.* FlowCover: Low-cost flow monitoring scheme in software defined networks. In: *Proc. of the 2014 IEEE Global Communications Conf. (GLOBECOM)*. IEEE, 2014. 1956–1961.
- [88] Su Z, Wang T, Hamd M. Costa: Cross-layer optimization for sketch-based software defined measurement task assignment. In: *Proc. of the 2015 IEEE 23rd Int'l Symp. on Quality of Service (IWQoS)*. IEEE, 2015. 183–188.
- [89] Tahaei H, Salleh R, Khan S, *et al.* A multi-objective software defined network traffic measurement. *Measurement*, 2017,95: 317–327.
- [90] Tahaei H, Salleh RB, Ab Razak MF, *et al.* Cost effective network flow measurement for software defined networks: A distributed controller scenario. *IEEE Access*, 2018,6:5182–5198.
- [91] Sharma NK, Kaufmann A, Anderson TE, *et al.* Evaluating the power of flexible packet processing for network resource allocation. In: *Proc. of the NSDI*. 2017. 67–82.
- [92] Tangari G, Tuncer D, Charalambides M, *et al.* Decentralized monitoring for large-scale software-defined networks. In: *Proc. of the 2017 IFIP/IEEE Symp. on Integrated Network and Service Management (IM)*. IEEE, 2017. 289–297.
- [93] Ahmed R, Boutaba R. Design considerations for managing wide area software defined networks. *IEEE Communications Magazine*, 2014,52(7):116–123.
- [94] Narayana S, Rexford J, Walker D. Compiling path queries in software-defined networks. In: *Proc. of the 3rd Workshop on Hot Topics in Software Defined Networking*. ACM Press, 2014. 181–186.
- [95] Phan XT, Fukuda K. Toward a flexible and scalable monitoring framework in software-defined networks. In: *Proc. of the 2017 31st Int'l Conf. on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, 2017. 403–408.
- [96] Yu C, Lumezanu C, Madhyastha HV, *et al.* Characterizing rule compression mechanisms in software-defined networks. In: *Proc. of the Int'l Conf. on Passive and Active Network Measurement*. Cham: Springer-Verlag, 2016. 302–315.
- [97] Curtis AR, Mogul JC, Tourrilhes J, *et al.* DevoFlow: Scaling flow management for high-performance networks. *ACM SIGCOMM Computer Communication Review*, 2011,41(4):254–265.
- [98] Liu X, Shirazipour M, Yu M, *et al.* Mozart: Temporal coordination of measurement. In: *Proc. of the Symp. on SDN Research*. ACM Press, 2016.

- [99] Alipourfard O, Moshref M, Zhou Y, *et al.* A comparison of performance and accuracy of measurement algorithms in software. In: Proc. of the Symp. on SDN Research. ACM Press, 2018.
- [100] Bosshart P, Daly D, Gibb G, *et al.* P4: Programming protocol-independent packet processors. ACM SIGCOMM Computer Communication Review, 2014,44(3):87–95.
- [101] Li B, Tan K, Luo LL, *et al.* Clicknp: Highly flexible and high performance network processing with reconfigurable hardware. In: Proc. of the 2016 ACM SIGCOMM Conf. ACM Press, 2016. 1–14.
- [102] Hassas Yeganeh S, Ganjali Y. Kandoo: A framework for efficient and scalable offloading of control applications. In: Proc. of the 1st Workshop on Hot Topics in Software Defined Networks. ACM Press, 2012. 19–24.
- [103] Su Z, Hamdi M. MDCP: Measurement-aware distributed controller placement for software defined networks. In: Proc. of the 2015 IEEE 21st Int'l Conf. on Parallel and Distributed Systems (ICPADS). IEEE, 2015. 380–387.

#### 附中文参考文献:

- [1] 周爱平,程光,郭晓军.高速网络流量测量方法.软件学报,2014,25(1):135–153. <http://www.jos.org.cn/1000-9825/4445.htm> [doi: 10.13328/j.cnki.jos.004445]
- [3] 左青云,陈鸣,赵广松,邢长友,张国敏,蒋培成.基于 OpenFlow 的 SDN 技术研究.软件学报,2013,24(5):1078–1097. <http://www.jos.org.cn/1000-9825/4390.htm> [doi: 10.3724/SP.J.1001.2013.04390]
- [4] 张朝昆,崔勇,唐嵩嵩,吴建平.软件定义网络(SDN)研究进展.软件学报,2015,26(1):62–81. <http://www.jos.org.cn/1000-9825/4701.htm> [doi: 10.13328/j.cnki.jos.004701]
- [20] 周桐庆,蔡志平,夏竟,徐明.基于软件定义网络的流量工程.软件学报,2016,27(2):394–417. <http://www.jos.org.cn/1000-9825/4935.htm> [doi: 10.13328/j.cnki.jos.004935]
- [21] 王蒙蒙,刘建伟,陈杰,毛剑,毛可飞.软件定义网络:安全模型、机制及研究进展.软件学报,2016,27(4):969–992. <http://www.jos.org.cn/1000-9825/5020.htm> [doi: 10.13328/j.cnki.jos.005020]



戴冕(1988—),男,江苏南京人,博士生,主要研究领域为软件定义网络,数据中心网络,网络测量技术.



周余阳(1994—),男,博士生,主要研究领域为网络安全,移动目标防御.



程光(1973—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络空间安全监测和防护,网络大数据分析.