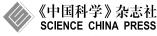
SCIENTIA SINICA Informationis

评述





SDN 网络测量技术综述

张恒、蔡志平*、李阳

国防科技大学计算机学院, 长沙 410073 * 通信作者. E-mail: zpcai@nudt.edu.cn

收稿日期: 2017-10-25; 接受日期: 2018-02-28; 网络出版日期: 2018-03-16

国家自然科学基金 (批准号: 61379145, 61702539) 资助项目

摘要 近年来,随着网络规模的不断扩大,结构和功能日趋复杂,网络流量快速增长,各种新型网络应用不断出现,对传统网络的配置、运行和管理提出了严峻挑战.软件定义网络 (software defined networking, SDN) 作为新兴的网络架构,较好地适应了各种网络应用的需求.在 SDN 网络中进行测量任务能及时了解、监控和掌握网络状态,是优化网络结构、改善网络服务质量、实现网络故障诊断和恢复的重要手段.本文对 SDN 网络测量技术进行综述,研究网络性能测量中的延迟测量、丢包测量和带宽测量等方面以及网络流量测量等相关技术,从设计理念、测量方式、测量对象、技术特点,以及优缺点等方面做了详尽的描述.文章最后进行了总结并对未来工作进行了展望.

关键词 软件定义网络, 延迟测量, 丢包测量, 带宽测量, 流量测量

1 引言

网络测量是获取网络性能和开展网络管理的基础, 其测量结果不仅能展现当前网络状态, 更能应用于网络管理和网络性能优化, 满足行为监控、流量工程、异常检测、故障分析等方面的需求, 为各类用户提供 QoS 保障. 然而, 传统网络的固有缺陷和不断发展使得网络测量的难度日益增大, 各种新的网络应用不断出现, 如云计算、云存储、大规模数据中心, 以及各种新的流媒体服务等, 更是对传统网络的配置、运行和管理提出了更高的要求. 传统网络测量技术的准确性、通用性和便捷性等方面都无法适应当前网络的要求.

网络测量的方法主要可分成两大类,即主动测量和被动测量方式.在传统网络中,主动测量模式下,主机主动向网络内发送特定的探测包,从而使重要的网络属性处在持续的监控状态下以保证服务性能.这种处理方式可以实现测量单向时延、循环时间或调整转发策略等目的.尽管主动测量方法能实现不同层次、不同粒度的端到端服务质量保证,但主动测量方法会产生特定的流量开销,这有可能对重要的网络流量产生影响.传统网络中的被动测量方法则是利用预先设定的模式触发,捕捉并分析

引用格式: 张恒, 蔡志平, 李阳. SDN 网络测量技术综述. 中国科学: 信息科学, 2018, 48: 293-314, doi: 10.1360/N112017-00203 Zhang H, Cai Z P, Li Y. An overview of software-defined network measurement technologies (in Chinese). Sci Sin Inform, 2018, 48: 293-314, doi: 10.1360/N112017-00203 实时流量.由于被动测量不产生任何探测报文,因而它不产生额外的开销,它适合收集并处理当前流量状态以及通过某个特定网络位置的整体流量状态.但被动测量的主要问题在于较小的一些参数可能被漏掉,采样方法设计不合理也会导致采样疏漏或采样重复,造成最终测量结果不准确.此外,在处理高速网络流量时,被动测量方法要求更为复杂的分析机制.而在 SDN (software defined networking)网络中,主动测量方法要求计划周密,处理并满足集中的控制结构要求.主动测量设备的部署增加了数据的获取量,使 SDN 的集中控制机制得到迅速满足.被动测量采用非介入方式,也不产生额外的开销,依赖数据包采样,并采用统计学方法推断当前测量状态.

因此,当前涌现出的一些尝试克服传统网络中出现的现实问题的方法是通过利用 SDN 网络控制与转发解耦的灵活性和可编程接口来满足细粒度的网络测量. 作为一种新的网络架构, SDN 基于网络抽象的思想, 其控制和转发功能分离的设计以及可编程网络的实现方式, 将 IP 网络设备上的控制权分离, 通过控制器 (SDN controller) 统一管理网络行为, 减少对底层设备的依赖, 屏蔽底层网络差异性,使得网络测量的逻辑实现可简单地通过控制器完成, 具体网络测量指标的采集交由交换机统一处理,测量方式更加高效, 测量结果更加准确可靠[1]. 总结而言, 基于 SDN 开展网络测量的优势可概括为以下几个方面:

- (1) 控制和转发功能解耦,各网络元素分工明确,控制器主要负责网络测量的逻辑实现,交换机主要负责各项网络信息的采集.
- (2) SDN 可编程的功能, 使得网络测量任务的实现者只需要综合考虑网络状态和网络应用的需求, 通过软件的方式完成测量任务.
- (3) OpenFlow 协议支持更细粒度的网络测量, 更灵活的网络资源分配, 这使得网络测量技术能基于上述特性得到更好的提升.
- (4) SDN 网络中实现网络测量能更好地保证检测结果的实时性, 可更好地应对网络拥塞、网络入侵、网络故障等情况, 及时掌握网络运行状态, 建立安全防护机制, 加强网络运维和管控能力.

近两年来,国内外研究机构相继开展了 SDN 网络测量技术的研究,包括美国 Princeton University, University of Southern California 等国外大学,清华大学、东南大学、国防科技大学、北京邮电大学等国内大学,以及 Google 等互联网企业,华为等网络设备厂商. SIGCOMM, INFOCOM 和 IMC 等国际一流学术会议上有关 SDN 网络测量的论文也逐渐增多.本文针对现阶段 SDN 中网络测量相关技术进行介绍和分析.第 2 节介绍了 SDN 网络测量的定义和具体的技术研究分类,第 3,4 节分别对网络性能测量和网络流量测量进行详细阐述,第 5 节对全文进行总结,第 6 节对未来工作提出展望.

2 SDN 网络测量概述

2.1 网络测量的定义

网络测量是为了实现了解、掌握和管理网络等目的,对网络行为进行特征化、具像化,对网络各项指标进行量化的最基础的技术手段.通过网络测量技术,网络管理者能更有效地理解网络中发生的行为,及时监控并纠正网络故障情况,有效检测和预防网络攻击行为,将网络事态维持在可控范围之内,网络运营商亦可通过网络测量技术及时获得网络状态信息反馈,提高网络服务质量 (QoS) 以满足用户对各种类型的网络应用的需求. 网络测量包含 3 个要素: 测量对象、测量环境以及测量方法 [2]. 测量对象是指被测量的节点和链路,以及带测量节点、链路或网络的某种特性.测量环境涵盖测量点选取、测量时间确定、测量设备和通信链路的类型等. 测量方法即针对某个具体的测量目标或某个特定网络

SDN network measurement				
Performance measurement	Traffic measurement			
 (1) Link latency (2) Packet loss (3) Available bandwidth 	(1) Comparison of traffic measurement technologies (2) Analyses of traffic measurement framework			

图 1 SDN 网络测量技术分类

Figure 1 Research on SDN measurement classification

选取契合的测量方法.

1995 年, 美国自然科学基金会 (National Science Foundation, NSF) 开始着手对互联网进行系统的测量工作. 1996 年初, 美国应用网络研究国家实验室 (NLANR) 在 NSF 的支持下召开了互联网统计与指标分析 (Internet statistics and metrics analysis, ISMA) 研讨会, 这标志着大规模、系统化的网络测量的开始. 随着网络不断发展, 网络测量技术不断创新优化以满足不同的需求. 同时, 随着新型网络应用不断涌现, 需要测量的网络性能和指标也不断增多, 对网络测量提出了更多更高的要求. 但现实情况是, 传统网络在发展过程中产生的架构复杂僵化, 各层网络协议臃肿繁复, 网络规模庞大, 给网络测量任务带来严峻挑战 [3]. SDN 的提出, 解决了网络测量在 IP 架构中面临的亟待解决的问题, 引起了学术界和产业界的高度重视.

2.2 SDN 网络测量技术研究内容

网络测量技术从不同角度出发,可以获得不同的分类结果. 在研究基于 SDN 网络中的测量技术时,本文按网络测量对象为分类标准,将其划分为网络性能测量和网络流量测量 2 个部分,如图 1 所示.

- (1) 网络性能测量. 网络性能测量是网络测量的核心分支, 通过某种方法或技术, 结合软、硬件工具来测量和具像化网络性能指标的一系列行为, 是理解、认识和量化网络行为最基本、最有效的手段, 对网络管理和优化、网络故障检测和排除, 以及网络安全防护机制建立等诸多领域均有广泛应用, 一直是网络测量研究的热点问题之一.
- (2) 网络流量测量. 流量测量是对一个特定网络中流量的规模、特征进行测量的过程, 是有效实现 网络管理的基础, 有助于网络运营商更深刻地了解网络特性. 传统测量方案 (如 NetFlow, sFlow, JFlow 等) 消耗太多网络资源 (带宽、CPU) 以及相当数量的硬件部署来满足不断提高的流量测量要求, 且多数情况下, 测量结果准确性和测量效率不尽如人意 [4~6]. 因此, 研究在 SDN 网络中的流量测量技术, 获取流量特征, 捕获网络异常, 更好地满足网络规划、流量计费、异常监测、负载均衡、流量工程、网络安全等网络应用的需求 [7], 具有十分重要的意义.

3 SDN 网络性能测量

根据 IETF 的 IP 网络性能测量工作组 (IPPM) 指定的相关草案和性能指标必须遵循的标准, 结合大多数研究对网络性能基本参数指标的定义, 本节将从 SDN 延迟测量、SDN 丢包测量和 SDN 带宽测量 3 个方面具体阐释 SDN 网络性能测量的相关技术.

3.1 SDN 延迟测量

随着虚拟云计算、大规模数据中心,以及各式各样的流媒体服务等延迟敏感型网络应用的不断涌现,对传统网络的运行效率、管理方式、QoS等方面提出了更高的要求^[8]. 在网络规模日益扩大,网络拓扑动态变化,运维难度日渐提高的现状下,如何持续监测网络状态,动态高效地测量网络延迟,在最短时间内找出低延迟网络路径,成为网络运营商亟待解决的难题. 目前 SDN 网络研究进展和 OpenFlow协议的标准化,促进了许多行之有效地测量 SDN 网络延迟的方案. 本节首先介绍延迟测量的基本概念,并从单向和往返延迟作为切入点分析 SDN 延迟测量方法.

3.1.1 定义

网络延迟通常分为单向延迟和往返延迟, 并由此衍生出抖动等性能参数, 根据 OpenFlow 协议特性, 通常还需要对控制器到交换机的控制链路延迟进行测量 ^[9]. 影响网络延迟的因素有很多, 主要分为以下 3 个方面: (1) 硬件相关, 包括物理层的传输介质、网络设备的转发处理性能等; (2) 数据结构相关, 包括传递的数据报文协议、报文长度、发送频率和方式等; (3) 网络负载相关, 即延迟与当前网络流量负载呈正相关. 下面将针对延迟相关概念进行阐释.

定义1 (单向延迟 (one way delay, OWD)) 数据包从发送端发出到接收端接收所花费的时间.

定义2 (往返延迟 (round trip time, RTT)) 数据包从发送端发出, 到发送端接收到来自接收端的确认信息所花费的时间.

定义3 (抖动 (jitter)) 是延迟的变化程度, 表明相同数据包通过同样路径而花费不同时间的情况. 抖动越小, 表明网络状况越好, 是延迟变化的具体情况.

定义4 (控制链路延迟 (control link delay, CLD)) 数据包从控制器出发到交换机接收所需要的时间, 反向传输亦然.

定义5 (时钟频差 (clock offset)) 某个设备的时钟频率与基准时钟频率之间的差值.

3.1.2 单向延迟测量

针对多数单向延迟测量, 其思路主要是从控制器发送探测报文, 经过待测路径再回到控制器, 通过减去首尾控制链路延迟得到待测路径的延迟. 这里需要说明的是, 基于以前的研究表明, 交换设备的时钟频差范围在百万分之五到百万分之百之间, 且在相当长一段时间内在很小的范围内波动, 因此这里不予考虑交换设备的时钟频差.

SLAM [10] 是典型的采用上述思想测量预设路径延迟的方案. 通过 OpenFlow 协议中 PACKET_IN 和 PACKET_OUT 控制消息测量网络中任意两个交换机之间的延迟, 且不需要额外的硬件支持. 同时, 为减去控制链路延迟, 利用 STATISTICS_REQUEST 和 STATISTICS_REPLY 消息获取控制链路 RTT. SLAM 主要分为 3 个步骤:

- (1) 预先选定需要测量延迟的路径, 向该路径上所有交换机预先安装转发规则, 确保探测报文沿测量路径传递.
 - (2) 在探测报文分别抵达路径首尾交换机时, 触发交换机向控制器发送 Packet In 消息.
- (3) 控制器通过两个 Packet In 消息到达控制器的时间计算出路径延迟. OpenNetMon [11] 令控制器向每条待测路径上的第一个交换机上发送探测报文,并让路径上的最后一个交换机将探测报文传回控制器. 控制器通过探测报文离开和到达控制器的时间,以及控制器到交换机的估计延迟,计算得出

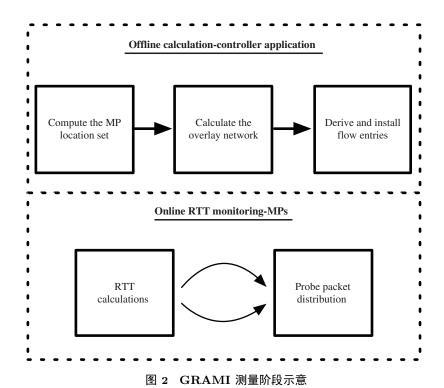


Figure 2 Measurement phases on GRAMI technology

路径上的延迟. 控制链路延迟根据 EchoRequest 消息测出的 RTT 值计算得出. 为使测量准确性以及探测报文开销能匹配每个流的大小, 注入探测报文的速率与路径的吞吐率需匹配. 具体如下:

$$t_{\text{delay}} = \left(t_{\text{arrival}} - t_{\text{sent}} - \frac{1}{2}(\text{RTT}_{s1} + \text{RTT}_{s2})\right). \tag{1}$$

3.1.3 往返延迟测量

文献 [12] 提出一种基于 TTL (time to live) 的测量往返延迟的改进方案,即 TTL-based looping method. 该方案的设计理念是将探测报文在测量链路上循环转发多次以求得该链路在某时间段内的平均延迟.循环次数受报文头部的 TTL 字段控制. TTL-based looping method 的改进之处在于,它不用在每个交换机上安装大量匹配规则以实现探测报文循环转发,只需 3 个流表项即可达到预期目标.同时它的实现方式比较简单,引入的探测流量较小. 但时效性和准确性不足是该测量方案明显的软肋. 文献 [12] 指出,由于测量结果是循环 TTL 次后的平均延迟,不适宜对延迟敏感应用的测量.而且,由于交换机队列延迟存在较大的波动性,在循环的过程中会被成倍放大,从而影响延迟测量结果的准确性.

GRAMI^[13] 在需要测量延迟的路径上预先设定监控点 MP (monitoring point), 控制器设定路由路径及转发规则, MP 发送探测报文并通过各探测报文返回的时间戳计算得出路径中所有链路的 RTT值. 图 2 描述了 GRAMI 的两个工作阶段: 离线阶段和在线测量阶段.

离线阶段, 需要在控制器安装监测程序并根据当前网络状态创建覆盖网络, 同时为所有交换机安装对应的转发流表项. 控制器根据 MP 的个数以及所有测量链路最短原则生成探测报文路由路径. 在线阶段, MP 向预设路径中的交换机发送探测报文, 交换机接收到探测报文后执行两个动作, 首先记录

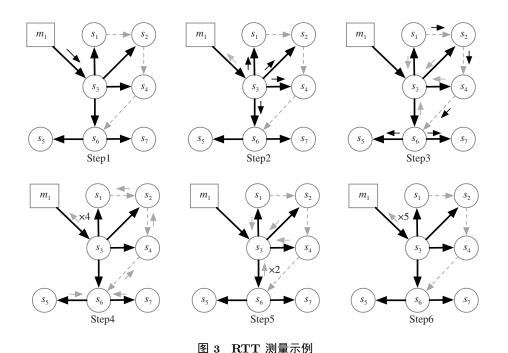


Figure 3 Example of RTT measurement technology

报文到达时间并复制报文; 其次将复制报文原路返回, 并将原探测报文转发至下游交换机. 最终控制器根据各报文返回的时间戳计算得出各链路的 RTT 值. 以图 3 为例, 为测量图中 s_2 , s_4 和 s_6 路径的 RTT, 控制器需要将 m_1 , s_3 , s_2 , s_4 , s_6 路径的总 RTT 减去 m_1 , s_3 , s_2 路径的 RTT.

文献 [14] 给出一种基于背景流突发情况下, 特殊的端到端延迟的测量方法. 一般延迟测量在无法测出准确时间的情况下, 都会采用平均值的方法来近似真实值, 即概率论与数理统计学中的期望. 而该测量方法利用期望与方差的数学关系, 通过严格的数学公式推导, 以方差和协方差的值来估计真实的端到端延迟, 测量结果更为准确.

文献 [14] 将总延迟 D 分解为每段链路 d_k 的和, 其中 d_k 由 3 个部分组成: 等待时间 w_k 、传输时间 s_k 、传播时间 r_k . 值得注意的是, 由于 r_k 是固定的, 因此方差为 0; 处理时间的量级较小, 在此忽略不计. 因此, 目标流量的端到端延迟方差 V[D] 的计算方法如下:

$$V[D] = V\left[\sum_{k=1}^{N-1} d_k\right] = V\left[\sum_{k=1}^{N-1} (w_k + s_k)\right] = V\left[\sum_{k=1}^{N-1} w_k\right] + V\left[\sum_{k=1}^{N-1} s_k\right] + \sum_{k=1, l=1}^{N-1} \text{Cov}(w_k, s_k).$$
(2)

其中等待时间 w_k 的方差计算方法如下:

$$V\left[\sum_{k=1}^{N-1} w_k\right] = \sum_{k=1}^{N-1} V[w_k] + 2\sum_{l=1,k<1}^{N-1} \text{Cov}(w_k, w_l).$$
(3)

传播时间 s_k 的方差计算方法如下:

$$V\left[\sum_{k=1}^{N-1} s_k\right] = \left(\sum_{k=1}^{N-1} \frac{1}{C_i}\right)^2 V[p_N^{f_1}]. \tag{4}$$

等待时间 w_k 与传播时间 s_k 的协方差如下:

$$Cov(w_k, s_l) \approx \frac{1}{C_i} (\mathbb{E}[w_k P_k^{f_1}] - \mathbb{E}[w_k] \mathbb{E}[P_k^{f_1}]). \tag{5}$$

通过计算上述公式,可完成目标流量的端到端延迟测量.实验测试了在不同节点数、不同负载、不同设备缓存容量等情况下的延迟,相比而言能获得较小测量误差和较高的准确性.但该方法的缺点也十分明显,繁琐的公式造成计算任务呈指数级增长.此外,为支持等待时间的计算,需要各节点支持时钟同步功能,对网络设备的性能要求比较高.

3.1.4 方法对比分析

SLAM 较 OpenNetMon 的优势体现在硬件设备方面, 无需花费额外的更新升级成本. 同时, SLAM 的测量思想更契合 SDN 网络控制与转发功能分离的设计初衷. 值得注意的是, 在文献 [10] 描述延迟测量过程中, 预装转发规则需严格匹配转发报文. SLAM 还存在交换机与控制器来回延迟不均等问题, 利用 OpenFlow 协议中 EchoRequest 消息测出的控制链路 RTT 的一半时间估计出的控制链路延迟与真实值之间会造成误差. 同时, 由于各交换机处理性能以及当前负载不同引起的处理延迟也对最后的结果造成影响.

GRAMI 可适用于不同网络拓扑结构, 控制器介入过程较少, 网络资源占用相对较少, 进而可降低控制层与数据层的控制开销. 而且 GRAMI 延迟测量结果的准确性和测量效率能随 MP 数量的增加而提升. 但鉴于 MP 在测量过程中起到至关重要的作用, MP 位置的选取直接影响 GRAMI 测量结果的好坏. GRAMI 需要预先设定好延迟监测路径, 凸显该测量方案灵活性不足. GRAMI 对动态网络的变化响应速度也不够, 主要体现在, 当网络发生较大变化时, 监控程序需要重新生成覆盖网络并为每个交换机安装新的流匹配规则. 较之 GRAMI, TTL-based looping 简单直接, 可移植性强, 但受限于测量精度, 只适合要求不高的应用.

3.2 SDN 丢包测量

3.2.1 定义

丢包是常见的网络现象, 其发生的原因也非常多. 以目前较为常见的数据中心网络为例, 数据信息在设备网络间传递、计算、展示与存储. 在整个数据的流转过程中, 丢包可能出现在网络拥塞路段, 设备端口缓冲区溢出会导致流入或流出的数据包丢弃. 丢包可能出现在网络黑洞处, 由于逻辑规则匹配错误而造成某些特定行为的流的大范围丢弃, 最终影响网络性能. 网络设备的硬件故障也是导致丢包的重要原因之一 [15]. 因此, 网络管理员需要耗费巨大的时间成本和网络资源来判定网络丢包的原因和地点, 采取一系列措施恢复网络性能. 由此可见, 丢包率是十分重要的用于检测网络状态、改善网络性能、提高网络服务能力的重要性能指标, 是网络管理员用于判别网络状态好坏, 制定并执行网络管理策略的重要依据. 下面首先将给出丢包检测相关定义.

定义6 (丢包 (packet loss)) 若发送端发送的数据包在指定时间内未能成功到达目的端, 称上述现象为丢包.

定义7 (丢包率 (packet loss rate)) 在单位时间内, 数据包丢失数与总发送数的比值.

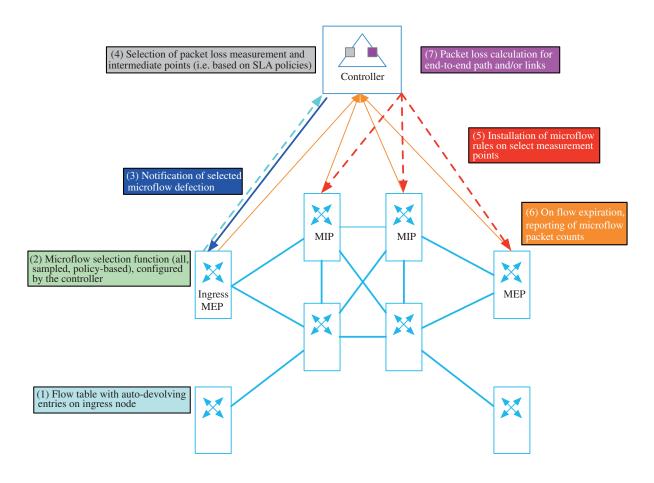


图 4 (网络版彩图) EPLE 丢包测量示意

Figure 4 (Color online) Packet loss measurement of EPLE. MEP is the abbreviation of measurement end point, MIP is the abbreviation of measurement installation point

3.2.2 SDN 网络中的丢包测量方案

通常, 网络丢包发生时间短, 发生次数也相对较少, 而且限于网络中粗粒度的流量处理方式, 准确检测丢包率也具有较大的挑战性. 文献 [16,17] 基于 OpenFlow 协议特性, 并囿于现实网络管理中避免过度测量开销的需求, 提出一种轻量的被动丢包测量方案 EPLE (如图 4).

EPLE 为丢包率检测提出了 3 个要求: 第一, 准确估计丢包率; 第二, 在考虑到数据层和控制层开销的情况下, 提供轻量级和扩展性能强的测量方案; 第三, 提供细粒度的丢包率检测. 具体测量思想可分解为以下几个步骤:

- (1) 对于待测路径上的入口交换机, 在流入流量中识别待测流后, 将该流的特征 (例如采取的协议、IP、TCP 端口号等) 通过 Packet_In 消息发送给控制器.
 - (2) 控制器将待测路径的转发规则下发至路径上所有测量节点并安装.
- (3) 当流传输结束时, 路径上所有测量节点向控制器发送 Flow_Removed 消息, 其中包括计数器中记录的数据包数等流的信息.
 - (4) 控制器通过接收的消息计算出端到端丢包率或相邻节点的丢包率.

通过 Mininet 平台实验验证, EPLE 的测量结果与实验之初的设定值相匹配. 同时, EPLE 还验证

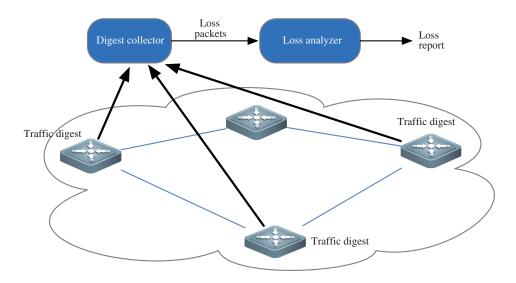


图 5 (网络版彩图) LossRadar 测量架构设计图

Figure 5 (Color online) Structure design of LossRadar measurement

了控制器和交换机的处理开销, 经实验, 仅在流表项安装和修改的过程中, 处理开销增长 4%~9%, 内存开销增加 1.5%~3%.

在实现上述测量过程, 其关键步骤在于在 OpenFlow 协议中支持 microflow 筛选功能. 当新的流量 到达节点时触发 auto-devolving 功能, 匹配新流是否符合预设的流筛选规则, 如果匹配成功, 该流对应的 idle_timeout 被设定, 同时流表中安装对应流表项.

现有的网络监控工具通常会覆盖多个监控目标,对某个特定网络性能的测量效果不能令人满意.为优化丢包率的测量,提高测量结果的准确性并减小测量开销,文献 [18] 设计出专门针对丢包率的测量系统 LossRadar. 该测量系统定位于小范围、轻量级的网络规模,主要实现细粒度的网络丢包检测和丢包类型判别,并捕捉具体的丢包信息,其测量结果可精确至毫秒级. LossRadar 的架构设计如图 5 所示,大体分为 3 个步骤. 首先, LossRadar 在网络中选取部分或全部节点安装计量模块,计量模块会对流入和流出设备的数据流进行监控,记录每个流的 flow ID 以及对应的 flow size. 当一个测量周期结束后,计量模块将监控到的数据上传给信息收集器 (digest collector). collector 对收集到的信息进行汇总处理,将捕获到的丢包信息传给丢包解析模块,提取具体的丢包信息 (5-tuple, Packet_Identifier等),生成最终的丢包情况汇报.

LossRadar 在具体实现过程中, 对丢包测量的细节考虑全面. 首先采取 invertible Bloom filter (IBF) 的编码方式以降低每个流占用的内存资源. 其次, LossRadar 是通过比对流入和流出的流量大小差值来判断丢包情况. 因此对路由路径的动态掌握以及在网络过程中的时钟同步问题需要着重考虑. 文中对每个流添加身份标签以及当前丢包检测的周期, 若因路由路径变化导致流量入口和出口大小不一致的情况, 可通过身份标签进行全局比对. 同时, 由于始终存在时钟误差的情况, 因此可根据流信息中附带的周期属性计算当前的流量计数, 确保丢包率的测量误差控制在最小. 对于最终丢包信息的汇总和报告, LossRadar 还提供了额外的信息添加功能, 可根据需要灵活地添加流的相关信息, 以便网络管理员获取必要的信息, 从而判断网络的状态, 制定相应的策略.

CP [19] 也是基于数据中心应用而设计的主动的丢包检测机制. 其特点是在遇到丢包情况时, 主动选择丢弃报文中的数据部分, 保留数据包头信息, 并根据报头信息通知发送端丢包情况, 以便后续重

传等情况. 在支持 CP 的交换机中,每个端口处设有一定大小的缓冲区. 缓冲区被分为两个部分,其中一部分用于存放待转发的数据包,另一部分用于存放砍去负载后的报文头信息. 当缓冲区将要填满时,CP 交换机对后续流入的报文进行丢弃数据内容处理,同时记录报文大小,重新计算并修改校验和,然后将报头存放在缓冲区等待转发. CP 还会将丢包信息回传给报文发送方,告知报文丢弃信息. 该功能在一定程度上能起到保护网络性能,确保 QoS 的作用.

3.3 SDN 带宽测量

带宽是重要的网络资源,测量网络带宽能够有力保障网络传输能力,实现流量负载均衡,诊断网络故障. 在传统网络规模激增的背景下,现有的带宽测量技术和可用的测量工具完全无法满足当前管理网络带宽资源的需求. 通常情况下,对非对称路径的上/下行带宽的测量一般采取两个单向测量的方法,且需要额外的控制模块. 而其他测量方案又受限于探测包的大小. 因而始终无法取得一个较好的平衡 [20]. 而新的 SDN 架构下,带宽依旧是评判网络性能的重要指标,对网络维护和管理,改善网络服务质量,加强网络故障诊断能力起到十分重要的作用. 对带宽的测量方法,也将基于 SDN 特性,设计更通用、更准确、更均衡的测量方案. 由于本小节将概述带宽相关概念,并以主、被动的测量方式为切入点,对 SDN 可用带宽测量进行综述.

3.3.1 定义

定义8 (链路/路径带宽 (link/path bandwidth)) 是指在无背景流量条件下, 链路/路径所能提供的最大传输速率.

定义9 (可用带宽 (available bandwidth)) 在给定链路上, 除去背景流量条件下可用的最大传输速率.

3.3.2 主动测量方案

OpenNetMon 提出一种主动测量吞吐率的方法, 可在此基础上测量可用带宽. 控制器在网络初始 化阶段会获取整个网络的拓扑结构, 对于链路的可用带宽测量, 控制器会针对根据拓扑结构规划路由 并生成相应的流表规则下发给对应链路上的交换机. 在实际测量部分, OpenNetMon 周期性地轮询链路上的交换机, 获取流量大小 S, 流持续时间 T 等相关信息, 以计算流经交换机的每个流的有效吞吐率 S/T. 根据已知的每条链路的带宽, 可计算得出链路的实际可用带宽. 对于端到端路径的可用带宽, 亦可通过服务器、主机等设备进行协同测量. 路径一端服务器向另一端发送测量报文, 控制器根据网络状况, 以自适应频率轮询首尾服务器, 以完成可用带宽测量. 这种方式能更好地在资源开销和结果准确度之间取得平衡.

3.3.3 被动测量方案

文献 [21,22] 提出一种在 SDN 网络中采用被动方式测量可用带宽的方法. 主要基于下列 3 种测量背景: (1) 网络路径失控, 运营商想获取两点之间的路由以及相关的可用带宽; (2) 网络路径可控, 在网络中任意两点之间找到最高可用带宽及其对应的路径; (3) 基于第 2 种情况, 在多个路径选择中, 获取最佳的路径方案.

文中使用被动测量方式,利用控制器协议测量链路可用带宽.首先该方法基于一个假设,即网络中每条链路的带宽 C_i 已知,其主要构想分为 4 个方面:

- (1) 通过控制器提供的网络拓扑结构掌握网络中的链路状态并创建网络拓扑图, 且每条链路的权值由该链路的可用带宽来表示, 据此为后续判断网络中任意两点的路径作准备.
- (2) 检测网络链路的已用带宽 B. 端口计数器中可以统计此端口发送和接收数据包的个数以及发送和接收的字节数, 控制器通过 OpenFlow 协议中的 FlowStatsReqest 消息以时间周期 T 询问交换机端口计数器中和 t 时刻的字节数. 然后, 控制器根据一个周期内端口计数器字节数的变化情况计算出目前此链路的已用带宽 B. 如式 (6) 所示, 其中 N_t 代表计数器的数值, T 代表统计时间长度.

$$B = \frac{N_t - N_{t-T}}{T}. (6)$$

- (3) 控制器利用链路的带宽容量和此链路的已用带宽计算出对应链路的可用带宽,即 ABW = C B,其含义为此链路还可以用来进行数据传输的能力.
- (4) 为实现第 2 个测量背景中提出的要求,某条路径的可用带宽为这条路径上所有链路可用带宽的最小值,即 $Min(ABW_i)$,对应路径可根据创建的 G(V,E) 图,利用 Dijkstra 算法获得最佳路径. 在实现第 3 个测量背景提出的要求时,需要比对每条路径对应的最小可用带宽中的最大值,即 $Max[Min(ABW_i)]$.

3.3.4 方法对比分析

被动测量方法操作简单,容易实现,不会对网络造成附加影响. 但被动测量不够灵活,测量的样本难以预知. 同时,文献 [21,22] 中测量结果的准确性受时间估计的影响较大. 主要原因在于: (1) 由于控制器需要读取交换机计数器上的测量信息,而控制器广播控制消息时各控制消息发送的先后顺序带来的时间差及其控制信道的传输延迟会引起时间上的混乱,导致可用带宽测量的准确性下降; (2) 由于 OpenFlow 协议中没有 TimeStamp 字段,交换机读取流计数器的精确时间只能通过控制器估计得到,其不确定性影响最后测量结果的准确性. 在当前实现方法中,根据 FlowStatsReq 消息到达控制器的时间差来近似真实的时间 T.

4 流量测量

当前私有云、公有云、混合云、移动云等云类型层出不穷, VMware vCenter Server、思科统一计算系统 (UCS)、阿里云 OS 等云产品相继问世, 这种规模大、资源集约的应用在高效便捷地满足现实需求的同时, 又增加了网络管理的复杂程度, 对网络流量测量技术提出了更高的要求. 一方面是因为和资源池网络资源虚拟化支撑起多租户模式的发展, 使得流量测量任务从源头处变得更加困难, 另一方面是基于云、数据中心等应用规模的爆发式增长, 使得流量测量任务越发繁重 [23]. 鉴于 SDN 在网络管理方面引入新的测量策略和框架以及基于流的细粒度流量测量技术, 本节将从两个方面深入解析 SDN 网络流信息采集, 并通过采集结果获得更好的流量控制和管理以及在网络性能指标上更好的表现: (1) 从流量测量方式出发, 分析主动和被动的网络流信息采集技术; (2) 从强化流量管控出发, 研究流量测量框架的效率和可扩展性.

4.1 网络流信息采集

4.1.1 主动测量技术

OpenTM ^[24] 是针对 SDN 网络中 traffic matrix 的估计方法, TM 表示的是网络中 OD 对 (源节点和目的节点对) 之间的流量大小, 对 TM 的准确评估是负载均衡、异常检测、路由配置等网络操作和任务的基础. OpenTM 追踪网络中所有的活跃流量, 通过控制器提供的路由信息和流转发信息并基于不同的交换机询问方法, 获取准确的流量大小以及数据包计数信息. 由于每个 OD 对间流经许多流量,对流经某个 OD 对的所有流量累加以完成 TM 更新.

文献^[24] 指出,在具体测量流量时会权衡资源使用,测量开销以及准确度.主要影响因素是测量路径上交换机的询问策略以及询问频率.询问策略需根据流量测量需求选定,这体现出 OpenTM 的灵活性.同时,为获取更准确的测量结果,询问交换机的频率将越高,测量开销也相应增加.同时, OpenTM 的缺点还体现在主观选择交换机方面,导致测量结果损失一部分准确性.文献^[24] 提到,存在下述可能情况,即 OpenTM 可能花费较大的代价换取较小的收益.

与 OpenTM 测量原理类似, PayLess [25] 是一个高效的、低开销的流量测量统计框架. 不同点在于, 在 PayLess 设计方案中, 控制器采用动态自适应频率的询问方式对交换机进行数据请求. 同时, 对并发的超时流量提供批处理功能, 令其合并优化, 进一步降低流量测量开销. PayLess 框架存在于控制器北向接口与上层应用之间, 主要由需求解释、轮询策略调度、交换机选择器, 以及数据整合存储 4 个部分构成. 在测量流量时, PayLess 首先解析上层需求, 选取合适的交换机轮训策略以及需要询问的交换机. 在获取流量信息后, PayLess 存储获取到的流量信息, 按应用需求整合数据并提交.

PayLess 最大的优势在于自适应轮询频率, 在保证准确性的情况下减少了网络开销. 实验结果表明, 相对于固定频率方案中 13.3 个/s 的询问开销, PayLess 引入的询问开销为 6.6 个/s. 总体说来, PayLess 通过较少的测量开销获取较高的测量准确率.

Planck ^[26] 采取了更为新颖的测量方式,利用商用交换机支持端口镜像功能的特性,牺牲交换机的一个端口作为镜像端口,将所有流经交换机的数据包在镜像端口复制后再转发. 该方法的好处在于突破传统采样方式的准确性受限于采样速率的瓶颈,即采样速率无法满足应用对流量测量的准确性要求. 通过将流量完整复制,在镜像端口处执行流量统计,可极大程度上确保测量结果的准确性. 实际测量中,交换带宽为 1 Gbps 的商用交换机上,从开始测量到获取测量数据的时间范围在 280 μs 到 7 ms之间, 10 Gbps 的商用交换机则是 275 μs 到 4 ms 之间,比传统测量速度快 11~18 倍,在硬件支持的理想情况下,可达到现行速度的 291 倍.

由于镜像端口的处理性能有限, 当流经交换机的流量超出镜像端口处理能力时, 会发生严重的丢包行为. 即使 Planck 为端口设置了缓冲区间, 但仍会对测量的准确度产生很大影响. 而且, Planck 只能在流量突发等事件发生后, 通过收集到的信息检测到, 不能实时地处理当前网络中发生的问题.

OpenNetMon 可通过主动询问路径上首尾交换机的端口计数器来实现丢包率检测. 根据源端口计数器和目的端口计数器之间关于数据包个数的差值可精确计算出某个采样频率下, 测量路径的丢包率. 较上述的测量方案, OpenNetMon 的测量方式较为简单, 准确性也得以保证.

CSAMP [27] 是一个统一的、集中式的流量监控系统,它通过调度全网范围内路由器、交换机等网络设备实现对单个流的采样,避免传统包采样带来的信息冗余、准确性低、小流丢失等情况.为提高采样效率,减少采样过程中的资源消耗, CSAMP 对数据包 5 元组进行 Hash 运算以减少采样数据的冗余度,并将该 Hash 值用为流表匹配项,当流表项存在该 Hash 值时,该流对应的字节计数器和包计数器更新相应数值,反之则创建新流表项以监控新的流入流量.

4.1.2 被动测量技术

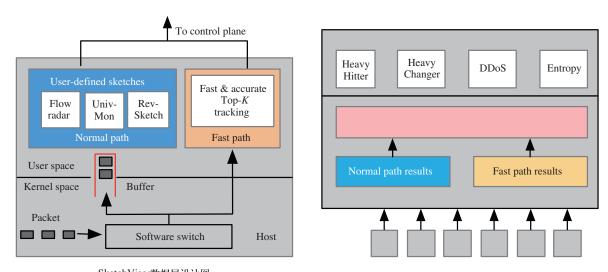
FlowSense ^[28] 利用 OpenFlow 协议中 PacketIn 和 FlowRemoved 控制消息实现对流量变化的实时监测. 当新的流到达交换机时,交换机向控制器发送 PacketIn 消息以询问新流的转发规则,控制器根据该消息获知当前活跃流的情况并下发对应流的转发规则. 当流结束时,交换机向控制器发送 FlowRemoved 消息,告知该流的持续时间、流量大小等具体信息,控制器借此可计算出交换机之间的链路利用率. 实验结果显示, FlowSense 有较高的准确性,能在 3 s 内完成 90% 的链路评估, 10 s 内完成全网链路利用率评估,并不引入任何额外测量流量. 但 FlowSense 只能在特定的离散时间点进行速率估计,限制了获取流信息的时效性,在遇到大流测量时,测量结果耗时较长,难以满足需求,因此 FlowSense 只适用于小流测量.

MicroTE [29] 是被动式的流信息统计方案. 该方法基于一个观测事实, 即 80% 的数据中心流量生存周期在 10 s 以内, 不超过 0.1% 的流量持续时间会超过 200 s, 以及当网络中发生流量突变时, 大多数流量可在 1~2 s 内被提前感知. 在具体的测量架构中, 文献 [29] 指出, MicroTE 首先将平稳可预测的流量分离开来, 并为其采取最优的路由策略. 剩余流量将触发一次流信息统计行为, 判断流量是否趋于平稳可测量范围, 并通过控制器为其制定最优的路由转发规则. MicroTE 能主动响应网络流量变化, 并及时提供最优的流量路由方案. 其次, MicroTE 的可扩展性强, 能应对不同规模的网络需求. 但 MicroTE 需要对网络中现有的交换机进行硬件升级, 不利于推广.

OpenSample [30] 是基于 OpenFlow 协议改进的一种被动流量测量方案. 该方案基于一个统计事实,即数据中心网络中 99% 的流量都是 TCP 流. OpenSample 的采样方法通过改进 sFlow 方法而来,提取相邻两个采样数据包头部的 TCP 序列号可得出流过的流量大小,将其除以采样时间窗口大小,可获得当前流量的准确速率. 实验结果显示, OpenSample 会引入 100 ms 控制回路延迟,但与传统方式的 1~5 s的延迟相比,有了极大提升. 而且 OpenSample 测量方案适用于任意 TCP 流,将其用于 Elephant 流检测时,不需要对交换机进行任何修改,这使得 OpenSample 具有较好的市场价值. 同时 Elephant 流也是流量工程的重点关注方向,也对网络安全相关领域的研究起到促进作用,使得 OpenSample 在研究领域也有巨大的潜力.

4.2 流量测量框架

OpenSketch [31] 是一个通用的、抽象的测量框架,与 SDN 网络架构类似,OpenSketch 提出将测量控制层和数据层解耦.数据层运行设为可动态配置的 3 阶段流水线,首先对数据流进行 Hash 运算以减少需要测量的数据量;其次在分类阶段,通过定制并匹配通配符规则实现对流量的分类;最后在流量计数阶段,根据不同精度需求,每个流对应一个或多个计数器以实现流信息统计、整合和回溯.文章中还提到,通过灵活地组合 Hash 运算、分类和计数功能实现对不同 sketch 方法的支持. 控制层最主要的功能在于根据任务需求动态选取合适的 sketch 方法,同时根据准确性要求和当前可用资源获取最优的测量结果. OpenSketch 的分层设计与 SDN 架构契合,可实现细粒度的流量测量,亦可从流识别 5 元组中提取 IP, MAC 地址等实现对终端的流量统计,整个过程耗费较小的开销,获取较高的准确性. 其流水线设计能将测量算法抽象细化为若干步骤,便于网络管理员实现不同的测量算法. 同时,OpenSketch 目前已被广泛用于数据中心网络的标准化制定,并向商用交换机拓展,商用价值潜力巨大.但 OpenSketch 需要网络交换机硬件支持,这对网络运营商来说耗资巨大,成为推广 OpenSketch 的巨大阻碍.



SketchVisor数据层设计图

图 6 (网络版彩图) SketchVisor 系统架构设计图

Figure 6 (Color online) Structure of SketchVisor measurement

由于 Sketch 快速的处理性能和较小的资源开销,在多个特定的应用场景下(如 Heavy Hitter, Heavy Changer, DDOS, Entropy 等),优化后的基于 Sketch 的测量方案 $^{[32\sim36]}$ 也获得了很好的效果. 在数据中心网络中,存在大量软件交换机,作为架顶交换机或主机内虚拟交换机部署在网络中. 因此,近期一些研究关注如何基于软件交换机利用 Sketch 进行测量的方案. 由此带来另一个问题,在软件交换条件下,测量的鲁棒性、可维护性、故障排查复杂性,以及计算资源和内存资源的较高需求都限制了 Sketch 的进一步提升. 针对上述问题,华为未来网络理论研究实验室的 Huang 与他的团队在 2017年 SIGCOMM 上设计出一个折中的方案 —— SketchVisor $^{[37]}$. 该方案最大的贡献在于开辟一条快速数据通路 FastPath. 在高速网络情况下,当负载增加以致正常的 Sketch 方案无法处理当前的过量数据时,快速数据通路将自动启用,缓存区溢出的数据将无差别分流至 FastPath 实现高速处理(图 6). SketchVisor 为快速路径设计 Top-K 算法,同时,提供两个额外的全局计数器以备后续流量情况估计时捕获小流量的聚合特征. SketchVisor 准确性的关键在于 Top-K 算法的设计. 该方案为每个计入 Hash 表的流量配备 3 个计数器 e_f , r_f 和 d_f , e_f 代表该流被计入前 K 个表项前可能丢失的最大字节数, r_f 代表持续统计的字节数, d_f 代表算法在每次估计前 K 个大流时计算得出的期望值. 可以得出 $r_f + d_f \leqslant v_f \leqslant r_f + d_f + e_f$,其中 v_f 代表 SketchVisor 方案测量得出的字节数. 通过分析和实验的比对,该 Top-K 算法能在略微降低统计结果准确性的情况下显著提高流量处理的效率.

上述两种基于 Sketch 的测量方案采用的是相对紧凑的数据结构,在减少数据量和内存占用等方面有自身优势. 而树、Hash map 等数据结构在快速索引方面的特性也能改善流量测量的性能. RMAS (re-evaluating measurement algorithms in software) [38] 通过对比 Sketch, Hash map 以及堆栈等 3 种不同数据结构获取的不同测量结果,得出采用 Hash map 的数据结构在各方面能获得更均衡的测量数据.同时,文中还进一步优化测量功能,提高测量性能和测量结果的准确度. RMAS 首先给出对比前提,基于研究得出测量过程中,包处理的性能与内存空间的占用大小无关. 接着以吞吐率、延迟和准确度为指标,对比 3 种数据结构的优劣. 然后在测量功能优化方面, RMAS 在准确度和测量性能方面取得折中,并通过分析当前硬件状况. 适时地将测量任务分散至多核,缓解单点过热带来的性能断崖.

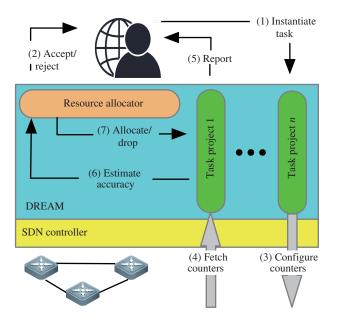


图 7 (网络版彩图) DREAM 系统设计图

Figure 7 (Color online) Framework of DREAM traffic measurement system

PayLess 是一个低开销的、基于主动询问的测量框架. PayLess 位于控制器北向接口与应用层之间,并为应用层提供良定义的可编程接口—— RESTful. PayLess 框架主要作用在于解析上层应用需求,将其转换成对交换机的具体轮询规划并完成交换机选取,最后该框架将反馈到控制器的数据进行整合,提交给上层应用. 而且, RESTful API 接口支持多语言编程,调用灵活,能极大程度降低网络应用开发难度,也能促进底层网络管理的高效和便捷,使得 PayLess 极具推广价值.

文献 [39] 针对流量测量多任务并行的现实情况, 提出 DREAM 测量框架, 用于动态分配硬件资源, 实现测量任务高效并行的目的. DREAM 框架主要分 3 层, 详见图 7.

- (1) 最上层为应用层,用于响应用户发出的测量任务,包括具体的任务类型、要求、准确性等要求,并接收提交的实时检测结果.
- (2) 中间层次运行 DREAM 算法. 主要负责创建对应的任务对象, 并通过 SDN 控制接口将抽象任务部署到多个交换机中. 部署方式即分配当前交换机可用 TCAM 资源. 为达到应用规定的准确性要求, DREAM 对测量的准确性进行评估, 并相应调整资源分配. 同时, 分析当前系统承载任务情况以及可用资源, 决定是否满足下一个测量任务的需求, 以实现资源的高效利用和测量任务并行处理.
 - (3) 底层是 SDN 网络. 为 DREAM 测量框架提供网络集中控制、资源存储、流量计数等功能.

HONE [40] 是有效管控网络流量的测量平台. 主要特点是设计标准接口使所有数据结构统一化、标准化 (如图 8 所示). 经 Hone 系统整合后的数据被 HONE 代理统一存储并按需读取. 同时代理程序亦可通过 HONE 提供的标准接口与网络设备进行交互, 执行细粒度的流量测量, 减少测量开销.

由于设备存储资源的有限性以及持续统计流量信息带来的高测量开销, HONE 提出了两个新颖的解决方案:

(1) 惰性测量. 由于 HONE 系统将所有统计数据抽象化为特定的数据结构, HONE 代理程序可与控制器协商频率并按需同步数据结构表中的内容. 既减轻了数据处理的难度, 又达到最小化测量开销的目的.

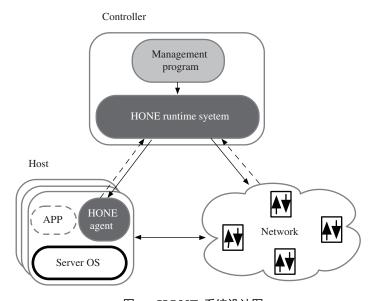


图 8 HONE 系统设计图

Figure 8 Framework of HONE traffic measurement system

(2) 增加并行数据流处理器. 由于 HONE 平台强大的可编程性, 程序员可在本地增加流处理器模块, 当终端应用提出流量测量需求并生成并行数据流时, 流处理器快速执行逻辑分析功能, 将测量需求与控制端系统同步以执行测量任务.

5 总结

SDN 网络测量作为了解、监控和掌握网络状态, 进而优化网络结构, 改善网络服务质量的重要手段, 是 SDN 网络发展的重要方向. 本文从网络性能测量和网络流量测量两个方面对当前 SDN 测量技术进行综述.

SDN 网络性能测量是网络测量的核心部分,对延迟、丢包、带宽的测量是量化分析网络性能,掌握当前网络状态的基础性工作.同时,网络性能测量结果的准确性、测量算法的稳健性对网络测量的其他方面起到了十分重要的辅助功能,对网络应用的支撑、网络 QoS 的保障,以及网络故障的诊断有着积极意义^[41].因此,在网络性能测量这节分 3 个部分,分别对 SDN 网络的延迟、丢包、带宽等性能指标的测量技术进行总结.文献 [10~16] 从延迟测量的方向性入手,分析单向和往返延迟在准确性和实操方面的不同,可根据测量要求判断选择不同的延迟测量方式.文献 [17~19] 通过丢包的概念入手,分析了具有代表性的丢包测量方法,并提出许多值得深思的细节问题.文献 [13,21,22] 基于可用带宽对网络故障诊断和恢复的重要意义,研究了主动测量和被动测量的相关技术.研究表明,在背景流非恒定的现实情况下,完全准确的测量可能难以做到,并以此指明高精度带宽测量对日后研究的现实意义.这些研究工作的比较分析如表 1 所示.

SDN 流量测量主要收集网络流信息,并分析得出流量特征,为管控网络流量,获取网络状态提供数据支持(见表 2). 文献 [35,36] 从流信息采集和流量测量框架研究两方面介绍了当前 SDN 流量测量的技术和测量框架的效率和扩展性能.通过分析对比总结在流量测量方面,综合网络负载、资源开销、结果准确性等因素设计可行的流量测量方案对于未来 SDN 网络的发展能起到重要意义.

表 1 SDN 网络性能测量技术汇总

 ${\bf Table~1} \quad {\bf Comparison~of~SDN~network~performance~measurement~technologies}$

名称	测量对象	测量方式	技术特点	优缺点分析
SLAM [10]	延迟	主动	充分利用 OpenFlow 协议功能, 不需要对硬件进行额外设计与支 持. 测量准确性与开销成正相关.	测量思路简单新颖, 在可能影响延迟测量结果准确性方面考虑不足, 例如控制链路延迟的具体测法.
TTL-based looping $^{[12]}$		主动	利用 TTL 作为控制参数控制数据包的循环次数, 从而获取链路的平均延迟.	思路清晰,实现难度较小. 但测量结果的准确性和时效性不高,只能满足一般网络应用.
GRAMI ^[13]		主动	预设测量点 MP, 计算覆盖网络 (overlay network), 实现网络中任意节点间延迟测量.	需要设计高效的 MP 选择算法, 优化 overlay 计算效率, 测量结果准确性高, 但其网络故障恢复能力较弱, 耗时较长.
End-to-End Multiflow delay [14]		主动	计算路径上每段链路延迟并求和,获取端到端延迟.	通过严格的数学公式推导,得出期望与方差的数学关系,测量结果准确性高.对网络设备的计算性能要求很高,适用范围较小.
EPLE [16, 17]	丢包	被动	基于 OpenFlow 协议实现丢包检测. 引入零探测流量, 适用与轻量级的丢包测量.	测量开销为零, 计算处理开销仅增加了 4%~9%, 内存开销仅增加了 1.5%~3%.
LossRadar ^[18]		主动	采用 invertible Bloom filter 编码方式减少内存资源占用,同时能根据需要灵活地提取丢包相关的流的信息.	测量方式简单,测量结果误差小.测量策略灵活,可按需获取出现丢包的流的相关信息.
CP [19]		主动	改变传统丢包方式, 只丢弃数据部分, 报文头保留, 以获得更好的测量结果.	测量方式新颖,缓冲区分块,以存放待转发数据包以及丢弃数据内容后的数据包头.对丢包信息的回传也可在一定程度上保护网络性能.
ABWM [21, 22]	带宽	被动	测量思想基于已有可用带宽测量的改进, 适用于 SDN 网络架构.	测量前提是网络中每条链路带宽己知,且背景流波动在可控范围之内,测量条件较为理想,据实际应用还有差距.
OpenNetMon [11]	延迟	主动	OpenNetMon 的测量内容涵盖了	测量方法简单有效, 测量结果的准
	丢包		延迟、丢包、带宽等主要网络性能指标,测量方案的设计思路具	确性和实时性相对较高.
	带宽		有创新性、灵活度高、扩展性好.	

6 研究方向与挑战

当前提出的一些 SDN 网络测量方法尽管解决了一些困难, 但许多设计思想中考虑欠缺的部分以及在面向真实网络部署过程中性能优化、准确性、可扩展性、安全性、分布式控制等方面都需要进一步讨论和发展. 最后提出以下几个未来需要改进和发展的方面:

(1) OpenFlow 对多控制器的支持. OpenFlow 协议是现在学术界和产业界公认的标准协议, 该协议的版本迭代是各大厂商关注的焦点, 决定着 SDN 未来的发展方向. 从 OpenFlow V1.0 开始, 该协议已经经历了几次大大小小的版本更新, 每次更新都以实际功能需求和演进趋势为导向 [42,43]. 但从目

表 2 SDN 网络流量测量技术汇总

Table 2 Comparison of SDN Traffic measurement technologies

名称	测量对象	测量方式	技术特点	优缺点分析
OpenTM [24] 字节数 持续时间 流量速率	主动	固定频率向交换机请求流的统 计数据.	采样节点选取方式较主观, 对测量结果有影响. 结果准确性与开销成正相关.	
SketchVisor [37]		主动	提供快速通路实现高速流量处 理性能.	克服 Sketch 设计本身缺陷, 提高高速网络下流量处理的效率. 测量结果的准确性相对下降, 同时在流量分流的处理上, 缺少灵活性, 主动选择能力不足.
PayLess [25]		主动	自适应频率对流信息进行采样.	根据应用具体需求和当前网络流量 状态, 动态调整对流的采样频率, 减 少了网络开销.
Planck ^[26]		主动	利用商用交换机端口镜像功能, 对流入流量进行备份并分析.	可对网络流进行全方位分析,结果可靠准确.但对端口处理性能要求高.需要考虑突发流情况下,端口的缓冲和应对机制.
FlowSense [28]		被动	基于 OpenFlow 协议改进的流量测量技术.	测量结果准确,测量开销较低,适用与小流检测,在大流测量方面实时性不高.
MicroTE ^[29]		被动	适用于小流检测, 可提前感知流量突发情况, 并提供重路由方案.	测量开销低,扩展性能强,需要对硬件升级,普及成本较高,不利于市场推广.
${\rm OpenSample}^{[30]}$		被动	适用于任意 TCP 流的检测, 对Elephant 流检测结果较理想.	改进 sFlow 采样技术, 测量效率和延迟均得到有效降低, 同时 Elephant 流检测具有较高市场价值.

前实际开发和测量过程来看, OpenFlow 协议中仍有许多需要增加和优化的方面来更好地支持网络应用的开发和使用. 以多控制器的扩展与协调为例, 单一控制器无法满足大规模的网络需求, 因此将网络划分多域, 域间控制器协调机制势必成为以后 OpenFlow 协议的发展方向之一. 控制器的组网方式、认证机制、通信方式、协同效率、安全防护等都需要 OpenFlow 协议扩展支持 [44].

- (2) 提高测量技术的普适性、扩展性, 以及测量对象的多样化. 目前许多测量技术的研究背景都是基于某些特定的应用场景或针对某些微小具体的现实问题, 这些测量技术的设计大多以前人的研究成果为基础, 进行性能优化、功能添加等改进, 因此, 网络测量的应用场景受限、测量对象单一、测量效果不够理想等情况成为亟待解决的问题 [45,46]. 目前缺少具有较强普适性和扩展性的测量框架来满足上述测量需求. 同时, 测量对象的多样化也是值得学者后续推进的相关内容.
- (3) SDN 网络测量在安全领域的深度应用. 网络流量和带宽的测量作为网络异常检测的重要依据, 应该与网络安全问题充分结合. SDN 网络分层思想在方便控制器监控和调度网络流量的同时, 也带来了许多安全问题. 由于控制器逻辑上集中的特性以及接口的开放性会产生针对控制器的新一轮攻击 [47], 例如对控制器进行 DDoS 攻击消耗控制器资源, 恶意占用整个 SDN 网络带宽等行为都将导致 SDN 网络瘫痪. 而现有的 SDN 被动测量方法灵活性不足, 无法适用流量波动较大的网络状态, 限制了发现及分析网络异常流量的能力 [48]. 因此, 设计自适应 SDN 网络测量方式, 结合当前网络拓扑和网络性能参数动态地配置测量任务, 有效应对复杂网络中出现的异常, 正确甄别异常流量, 制定相应安

全策略是下一步值得研究的课题.

(4) 增强真实网络环境的部署能力. 当前, SDN 网络环境部署主要面向校园网、企业网, 以及一定规模的数据中心, 大多数 SDN 网络测量应用仅限于理论阶段, 在理想实验环境下取得了一定成果 [49]. 但真实网络规模宏大, 环境复杂多变, 各种突发情况层出不穷, 如何将各类测量技术从理论应用于实际, 能够克服真实网络环境带来的现实困难, 在测量开销可接受的程度下获取可靠的、准确的测量结果. 是未来 SDN 网络测量技术值得深入研究和发展的方向 [50].

致谢 感谢审稿专家的中肯建议及编委的辛勤工作!

参考文献

- 1 Kreutz D, Ramos F M V, Esteves Verissimo P, et al. Software-defined networking: a comprehensive survey. Proc IEEE, 2014, 103: 10–13
- 2 Cai Z P, Yin J P, Liu F, et al. Efficiently monitoring link bandwidth in IP networks. In: Proceedings of IEEE Global Telecommunications Conference, St. Louis, 2005
- 3 Yan X C, Yin J P, Cai Z P. Survey on network topology discovery algorithm. Comput Eng Appl, 2007, 43: 131–135 [闫兴篡, 殷建平, 蔡志平. 网络拓扑发现算法综述. 计算机工程与应用, 2007, 43: 131–135]
- 4 Tsai P W, Hsu C Y, Luo M Y, et al. On the implementation of adaptive flow measurement in the SDN-enabled network: a prototype. Proc Asia-Pacific Adv Netw, 2015, 40: 7
- 5 Akyildiz I F, Lee A, Wang P, et al. A roadmap for traffic engineering in SDN-OpenFlow networks. Int J Comput Telecommun Netw, 2014, 71: 1–30
- 6 Zhou T Q, Cai Z P, Xia J, et al. Traffic engineering for software defined networks. J Softw, 2016, 27: 394–417 [周桐庆, 蔡志平, 夏竟, 等. 基于软件定义网络的流量工程. 软件学报, 2016, 27: 394–417]
- 7 Nunes A, Mendonca M, Nguyen X N, et al. A survey of software-defined networking: past, present, and future of programmable networks. Commun Surv Tut IEEE, 2014, 16: 1617–1634
- 8 Karakus M, Durresi A. Quality of service (QoS) in software defined networking (SDN): a survey. J Netw Comput Appl, 2017, 80: 200–218
- 9 Cai Z P. Network measurement technologies, models and algorithms based on active and passive measurement. Dissertation for Ph.D. Degree. Changsha: National University of Defense Technology, 2005 [蔡志平. 基于主动和被动测量的网络测量技术,模型和算法研究. 博士学位论文. 长沙: 国防科学技术大学, 2005]
- 10 Yu C, Lumezanu C, Sharma A, et al. Software-defined latency monitoring in data center networks. In: Proceedings of Software-defined Latency Monitoring in Data Center Networks, New York, 2015. 360–372
- Adrichem N L M V, Doerr C, Kuipers F A. OpenNetMon: network monitoring in OpenFlow software-defined networks.
 In: Proceedings of IEEE Network Operations and Management Symposium, Krakow, 2014
- 12 Sinha D, Haribabu K, Balasubramaniam S. Real-time monitoring of network latency in software defined networks. In: Proceedings of IEEE International Conference on Advanced Networks and Telecommunications Systems, Kolkata, 2015
- 13 Atary A, Bremler-Barr A. Efficient round-trip time monitoring in OpenFlow networks. In: Proceedings of the 35th Annual IEEE International Conference on Computer Communications, San Francisco, 2016
- 14 Nguyen H N, Begin T, Busson A, et al. Evaluation of an end-to-end delay estimation in the case of multiple flows in SDN networks. In: Proceeding of the 12th International Conference on Network and Service Management, Montreal, 2017. 336–341
- 15 Panwaree P, Kim J, Aswakul C. Packet delay and loss performance of streaming video over emulated and real Open-Flow networks. In: Proceedings of the 29th International Technical Conference on Circuit/Systems Computers and Communications, At Phuket, 2014
- 16 Fu C, John W, Meirosu C. EPLE: an efficient passive lightweight estimator for SDN packet loss measurement. In: Proceedings of IEEE Conference on Network Function Virtualization and Software Defined Networks, Palo Alto, 2017. 192–198

- 17 John W, Meirosu C. Low-overhead packet loss and one-way delay measurements in service provider SDN. ONS, 2014. http://www.usenix.org/sites/default/files/ons2014-poster-john.pdf
- 18 Li Y, Miao R, Kim C, et al. LossRadar: fast detection of lost packets in data center networks. In: Proceedings of the 12th International on Conference on Emerging Networking Experiments and Technologies, Irvine, 2016. 481–495
- 19 Cheng P, Ren F Y, Shu R, et al. Catch the whole lot in an action: rapid precise packet loss notification in data centers.
 In: Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation, Seattle, 2014.
 17–28
- Chan Edmond W W, Chen A, Luo X P, et al. TRIO: measuring asymmetric capacity with three minimum round-trip times. In: Proceedings of the 7th Conference on Emerging Networking Experiments and Technologies, Tokyo, 2011
- 21 John W, Meirosu C. Low-overhead packet loss and one-way delay measurements in service provider SDN. ONS, 2014. http://www.sjalander.com/wolfgang/publications/ons2014-poster-john.pdf
- 22 Sezer S, Scott-Hayward S, Chouhan P K, et al. Are we ready for SDN? implementation challenges for software-defined networks. IEEE Commun Mag, 2013, 51: 36–43
- 23 Yassine A, Rahimi H, Shirmohammadi S. Software defined network traffic measurement: current trends and challenges. IEEE Instrum Meas Mag, 2015, 18: 42–50
- 24 Tootoonchian A, Ghobadi M, Ganjali Y. OpenTM: traffic matrix estimator for OpenFlow networks. In: Proceedings of the 11th International Conference on Passive and Active Measurement, Zurich, 2010. 201–210
- 25 Chowdhury S R, Bari M F, Ahmed R, et al. Payless: a low cost network monitoring framework for software defined networks. In: Proceedings of IEEE Network Operations and Management Symposium, Ahmed, 2014
- 26 Rasley J, Stephens B, Dixon C, et al. Planck: millisecond-scale monitoring and control for commodity networks. In: Proceedings of the 2014 ACM Conference on SIGCOMM, Chicago, 2014
- 27 Sekar V, Reiter M K, Willinger W, et al. cSamp: a system for network-wide flow monitoring. 2008. https://www.usenix.org/legacy/event/nsdi08/tech/full_papers/sekar/sekar_html/
- Yu C, Lumezanu C, Zhang Y P, et al. Flowsense: monitoring network utilization with zero measurement cost. In: Proceedings of International Conference on Passive and Active Network Measurement, Hong Kong, 2013
- 29 Benson T, Anand A, Akella A, et al. MicroTE: fine grained traffic engineering for data centers. In: Proceedings of the 7th Conference on Emerging Networking Experiments and Technologies, Tokyo, 2011
- 30 Suh Junho, Kwon T T, Dixon C, et al. Opensample: a low-latency, sampling-based measurement platform for commodity SDN. In: Proceedings of the 34th International Conference on Distributed Computing Systems, Madrid, 2014
- 31 Yu M, Jose L, Miao R. Software defined traffic measurement with OpenSketch. In: Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, Lombard, 2013
- 32 Charikar M, Chen K, Farach-Colton M. Finding frequent items in data streams. In: Proceedings of International Colloquium on Automata, Languages, and Programming, Turku, 2004
- 33 Cormode G, Muthukrishnan S. What's new: finding significant differences in network data streams. IEEE/ACM Trans Netw, 2005, 13: 1219–1232
- 34 Li Y L, Miao R, Kim C, et al. FlowRadar: a better netflow for data centers. In: Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation, Santa, 2016
- 35 Liu Z X, Manousis A, Vorsanger G, et al. One sketch to rule them all: rethinking network flow monitoring with UnivMon. In: Proceedings of the 2016 ACM Conference on SIGCOMM, Florianopolis, 2016
- Schweller R, Li Z C, Chen Y, et al. Reversible sketches: enabling monitoring and analysis over high-speed data streams. IEEE/ACM Trans Netw, 2007, 15: 1059–1072
- 37 Huang Q, Jin X, Lee P P C, et al. Sketch Visor: robust network measurement for software packet processing. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication, Los Angeles, 2017
- Alipourfard O, Moshref M, Yu M L. Re-evaluating measurement algorithms in software. In: Proceedings of the 14th ACM Workshop on Hot Topics in Networks, Philadelphia, 2015
- Moshref M, Yu M L, Govindan R, et al. DREAM: dynamic resource allocation for software-defined measurement. In: Proceedings of the 2014 ACM Conference on SIGCOMM, Chicago, 2014

- 40 Sun P, Yu M L, Freedman M J, et al. Hone: joint host-network traffic management in software-defined networks. J Netw Syst Manage, 2015, 23: 374–399
- 41 Xu H L, Yu Z L, Qian C, et al. Minimizing flow statistics collection cost using wildcard-based requests in SDNs. IEEE/ACM Trans Netw, 2017, 25: 3587–3601
- 42 Yu Y M, Zhang J, Zhao Y L, et al. Field demonstration of multi-domain software-defined transport networking with multi-controller collaboration for data center interconnection. J Opt Commun Netw, 2015, 7: 301–308
- 43 Fu Y H, Bi J, Wu J P, et al. A dormant multi-controller model for software defined networking. China Commun, 2014, 11: 45–55
- 44 Wang J L, Shou G C, Hu Y H, et al. A multi-domain SDN scalability architecture implementation based on the coordinate controller. In: Proceedings of International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Chengdu, 2016
- 45 Liu C, Malboubi A, Chuah C N. OpenMeasure: adaptive flow measurement inference with online learning in SDN. In: Proceedings of IEEE Conference on Computer Communications Workshops, San Francisco, 2016
- 46 Farshad A, Georgopoulos P, Broadbent M, et al. Leveraging SDN to provide an in-network QoE measurement framework. In: Proceedings of IEEE Conference on Computer Communications Workshops, Hong Kong, 2015
- 47 Dacier M C, Dietrich S, Kargl F, et al. Network attack detection and defense security challenges and opportunities of software-defined networking. Dagstuhl Rep, 2017, 6: 1–28
- 48 Saxena M, Rakesh K. A recent trends in software defined networking (SDN) security. In: Proceedings of the 3rd International Conference on Computing for Sustainable Global Development, New Delhi, 2016
- 49 Hong D K, Ma Y D, Banerjee S, et al. Incremental deployment of SDN in hybrid enterprise and ISP networks. In: Proceedings of the Symposium on SDN Research, Santa Clara, 2016
- 50 Canini M, Feldmann A, Levin D, et al. Panopticon: incremental deployment of software-defined networking. ACM Symposium on SDN Research, 2016. https://www.opennetworking.org/images/stories/downloads/sdn-resources/IEEE-papers/panopticon-incremental-deplyment-of-sdn.pdf

An overview of software-defined network measurement technologies

Heng ZHANG, Zhiping CAI* & Yang LI

College of Computer, National University of Defense Technology, Changsha 410073, China

 $\ ^*$ Corresponding author. E-mail: zpcai@nudt.edu.cn

Abstract In recent years, with the continuous expansion of the size of networks, the complexity of network structure and functionality and the volume of network traffic have immensely increased. This has resulted in severe challenges in configuring, operating and managing traditional networks. As a prominent way of monitoring and handling network status in order to optimize network structure, enhance the quality of service and achieve network error detection and recovery, software-defined networks, as a new emerging network structure, has considered all of these problems and suits these network applications well. This paper is a survey of the state-of-the-art research on measurement technologies in software-defined networks, of which we detail the design concept, performance approach, measurement objects, technical advantages and disadvantages from three standpoints: network performance, topology, and traffic measurement. We mainly cover latency, packet loss and bandwidth measurements when considering network performance measurement. Finally, the technological approaches are summarized and future research issues are discussed.

Keywords software-defined networking, latency measurement, packet loss measurement, bandwidth measurement, traffic measurement



Heng ZHANG was born in 1991. He is a Ph.D. candidate at the College of Computers, National University of Defense Technology and a student member of the CCF. His research interests include network measurement and network security.



Zhiping CAI was born in 1975. He is a professor at the College of Computers, National University of Defense Technology. His research interests include network security, network measurement and big data.



Yang LI was born in 1990. He received a M.S. degree in computer science from the National University of Defense Technology in 2017. His research interest includes network measurements.