

文章编号: 2096-1618(2018)01-0018-05

SDN 网络性能测量系统设计与实现

陈蕾衣, 张新有

(西南交通大学信息科学与技术学院, 四川 成都 610031)

摘要:作为一种新型网络结构,软件定义网络(SDN)将数据转发平面与控制平面解耦合,实现了网络业务的可编程。而对网络性能参数的测量,是掌握一个网络运行状况的重要指标之一。在充分利用SDN网络特点的基础上,借鉴传统网络中对性能参数的定义及测量方法,结合主动测量与被动测量方法,对SDN网络中带宽、时延、丢包率、链路利用率等网络性能参数进行测量。仿真结果显示,文中实现的网络性能测量系统测量的数据与预期结果一致。

关键词:软件定义网络;OpenFlow;带宽;时延;丢包率

中图分类号:TP393

文献标志码:A

doi:10.16836/j.cnki.jcuit.2018.01.004

0 引言

软件定义网络SDN是由美国斯坦福大学clean slate研究组提出的一种新型网络架构^[1],其目的在于改变难以创新的传统网络架构。与传统网络中数据转发平面与控制平面紧密耦合不同的是,SDN网络的最大特点就是数据转发平面与控制平面解耦合^[2]。整个网络由逻辑上的控制器集中管理,底层网络设备仍保留传统网络中的数据转发功能,从而有效屏蔽来自底层设备的差异性。

根据开放网络基金会(open networking foundation, ONF)提出的SDN三层架构来看(如图1所示^[3]),控制层与物理层之间通过SDN南向接口协议OpenFlow进行通信,应用层通过控制层提供的编程接口对物理层进行编程,并将网络控制权向用户开放,用户可自定义所需的网络转发路径和转发策略等,故SDN又称为可编程网络。

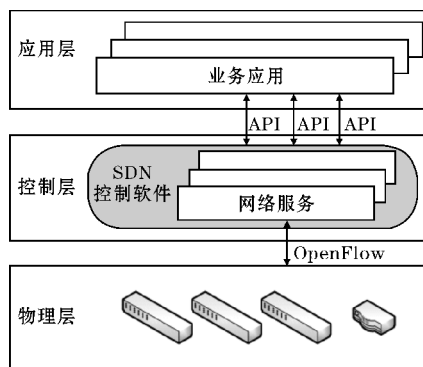


图1 SDN架构图

OpenFlow是斯坦福大学Nick McKeown教授在ACM Communications Review上发表的文章OpenFlow: Enabling Innovation in Campus Networks^[4]中首次提出,是基于流设计的一种SDN南向接口协议。OpenFlow交换机维护一个流表,并根据流表携带的相关信息进行转发,而流表的生成、下发和维护则由控制器完成。控制器通过OpenFlow协议对OpenFlow交换机中的流表进行管理控制,进而实现对整个网络的控制管理。

通过对网络性能参数的测量^[5],能够让用户第一时间了解网络的动态,并根据网络动态合理分配网络资源,有效提供网络资源利用率。针对此问题设计并实现了一个基于SDN的网络性能测量系统。

1 相关研究工作

基于OpenFlow的SDN网络由OpenFlow交换机、Flow Visor和Controller 3部分组成^[6]。OpenFlow交换机根据控制器下发的流表规则对数据转发进行管理,因而交换机中保存着全部流表以及流表中每一个流表项的统计信息。控制器通过与交换机通信获得这些统计信息,进而掌握网络的运行状况及性能参数,从而使SDN网络自身就具备了一定的性能检测能力。

OpenFlow原生测量采用被动测量方式^[7],在测量过程中控制器向交换机下发查询控制命令,交换机则负责提供和更新相应的流表信息,配合测量工作。这种测量方法得出的结果较为准确,并且由于被动测量的缘故,无需使用额外的测量工具向网络中注入流量,操作轻松实现简单。但此方法的一个明显缺点即在大型OpenFlow网络中,为及时获得交换机中的统计信息,控制器需要频繁地与各个交换机进行交互,使控制器的负担加重。控制器可能会因负载过大而无法完成

测量,并且频繁通信也会对测量结果的实时性产生影响,因此控制器的处理能力和通信能力容易成为测量系统的瓶颈。

OpenSketch^[8]是一个可软件定义的流量测量方案,利用SDN网络数据面与控制面分离的特点,在数据面上设置了一个3层管道用于多任务的分配。在控制面上,OpenSketch提供了一个测量库为测量任务自动配置管道类型和分配资源。测量人员可以通过制定不同的规则提取需要的测量数据,相比于OpenFlow的原生统计信息,OpenSketch能够更好地满足不同的测量需求。但是依然存在控制器负载过重的情况,并且由于OpenSketch测量的是交换机中的数据流,不具备主动测量的能力,无法实现端到端的网络测量,具有一定的局限性。

Open TM^[9]是OpenFlow网络的流量矩阵估计系统,通过控制器中的路由信息获得交换机中的流量统计信息,可以十分精确地获得网络流量信息,而不会产生过多的额外负载。Open TM是一个不错的OpenFlow网络监测方案,但Open TM系统在为每一条流选择交换机的时候都要进行优化轮询,这就需要Open TM周期性地查询交换机流量信息,导致查询频率增大,增加了控制器的负担。并且测量的参数只有网络流量信息,使得该系统的使用存在很大的局限性。

Ofpeck^[10]是另一个用于OpenFlow网络状态和性能测量的方案。通过主动测量的方法,使用额外的流量工具获取有关的网络性能参数。Ofpeck方案可以单独运行在与交换机相连的任意主机上,可测量与其相连的交换机的流表建立时间,以及指定IP交换机的往返时延(双向时延)、丢包率等。Ofpeck方案简单易行,但是通过此方案只能收集本地主机信息,所以可测量的性能参数种类有限。

参考以上文献,文中考虑了SDN网络中利用流表项中的统计数据,再结合主动测量与被动测量,对带宽、时延、丢包率及链路利用率进行测量。

2 网络性能测量系统设计

2.1 延迟测量

对于时延的测量思路如图2所示。控制器首先下发由源交换机(简称SW)到目的交换机(简称DW)的流表项。流表项安装完毕后控制器向源交换机下发一个Packet-out报文,报文的数据段携带了任意一个约定好的协议报文,其报文的数据段携带了控制器下发报文时的时间戳,Packet-out报文的动作指示源交换机将数据包转发给下一跳交换机。当目的交换机收到上一跳的交换机转发来的数据包后,无法匹配对应流表

项,从而向控制器发送Packet-in消息,将数据包发回控制器。控制器收到数据包后用当前时间减去报文内的时戳,得到时间差 $T1$,其中包含了数据包从控制器到源交换机的时延、源交换机到目的交换机的时延及目的交换机到控制器的时延,即图2中点a至f方向整个过程所用的时间。因为此时所得的时间只包含了单向传输时延,所以同理,控制器向目的交换机下发一个携带控制器下发报文时的时间戳的Packet-out报文,数据包从目的交换机发送给源交换机,再由源交换机发送回控制器,用此时的时间减去时间戳得到时间差 $T2$,其包含了数据包从控制器到目的交换机的时延,目的交换机到源交换机的时延以及源交换机到控制器的时延,即图2中点f至a的整个单向过程。

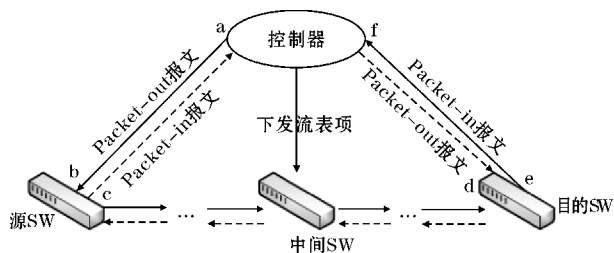


图2 时延测量

此时得到的两个时间差里均包含了数据流经交换机到控制器的往返时间,用时间差减去这部分时间即为所测量的往返时延。幸而在SDN网络中很容易获得交换机与控制器的往返时延,通过控制器分别向源交换机与目的交换机发送带有时戳的Echo request消息,交换机收到后分别向控制器回复Echo reply消息,控制器收到回复消息以后用当前时间分别减去时戳即为源交换机和目的交换机与控制器的往返时延 T_s 和 T_d 。

所以往返时延计算方法如式(1)所示。

$$Delay = T1 + T2 - T_s - T_d \quad (1)$$

使用上述方法的不足是,在链路中的中间交换机数量较多的情况下,测量用时过长,控制器无法掌握中间路由器的转发过程和转发时间,只能等待目的交换机传回数据包。假设数据包还没有到达目的交换机就被传回控制器,在这种情况下,控制器会认为是数据包已到达目的交换机,最后导致测量结果不准确。事实上在实际SDN网络中,这种情况不是没有可能发生的。如果一条链路过于复杂,当前段的中间交换机延迟较大时,导致后面的交换机还未收到数据包就已经超时,交换机中的流表项被删除。等到该交换机的上一跳交换机将数据包转发过来时,没有流表项匹配,交换机会将数据包直接转发给控制器。这样,数据包就在目的交换机之前转发回了控制器。由于控制器一次可以发送多个数据包,因此为了防止上述情况发生,将链路中 N 个交换机均分解为 $N-1$ 个子链路,对每一个

子链路的时延进行测量,各部分的时延和即为所求,计算公式:

$$Delay = \sum_{i=1}^n SubDelay(i) \quad (2)$$

对于单向时延的测量,可借鉴上述双向时延的思想,唯一不同的是,只需对点 a 至 f 这个方向进行测量。该过程完成后,与前相似,就能计算出源交换机与目的交换机的单向时延测量,具体细节和后续步骤与双向时延测量方式相似。

2.2 丢包率测量

传统网络中对于丢包率的计算方法采用主动测量^[11],首先要测量出链路源节点发出的数据包数量 N ,和目的节点收到的数据包数量 M ,再通过公式(3)对丢包率 $LossRate$ 进行计算。

$$LossRate = (N - M) / N \times 100 \% \quad (3)$$

与传统网络不同的是,SDN 网络可以利用自身的特点获取数据包的统计信息。每个 OpenFlow 交换机都有一张各自的流表,而流表项中的计数器可以统计收发数据包的数量。在这里,有两种方法可以获得计数器中的统计量信息,一种是控制器定期查询交换机的统计量信息,进而计算丢包率,具体步骤如下:

(1)通过控制器的全局拓扑得到源交换机到目的交换机的路由信息,然后向中间交换机下发流表项,指示交换机的数据转发;

(2)当数据包被转发至目的交换机时,目的交换机流表没有匹配流表项,向控制器发送 Packet-in 消息;

(3)控制器连续向源交换机发送若干 Packet-out 消息,得到发送数据包数量 N 。发送完成后等待若干时间,统计控制器收到的 Packet-in 消息,得到收到的数据包 M ,通过公式(3)计算丢包率。

还有另一种测量方法是利用 OpenFlow 协议中的 Flow-removed 消息,如图 3 所示。测量思路是当某一在交换机流表中找不到相应的流表项匹配时,交换机会向控制器发送 Packet-in 消息,并携带第一个数据包。而后控制器通过路由抉择,在 T_0 时刻向交换机下发流表项,指导其余数据包后续转发工作。当该流结束后,交换机会保留一段时间流表项,直到流表项 idle_timeout 或 hard_timeout 到达,流表项被删除。此时,若流表项的 Flag 位置 1,交换机会向控制器发送一条 FlowRemoved 消息,图 3 中两个交换机分别在 T_1 、 T_2 时刻向控制器发送消息,并携带若干统计数据,其中包含数据包计数和字节计数。通过一条链路源节点和目的节点的 FlowRemoved 消息包含的信息,同样可以按照公式(3)计算丢包率。

对上述两种方法进行分析。第一种方法属于主动测量,向网络中注入额外流量,影响网络运行状态,并

且控制器需要连续向交换机发送若干 Packet-out 消息,这将使控制器负担加重,影响控制器正常使用。而后者属于被动测量,不会给网络或控制器增加负担,但需要等待交换机中流表超出空闲时间后,删除流表项,控制器才能收到数据包计数消息,不能确定等待时间。而前者则可以控制测量时间,随时可进行测量。但在实际网络中,交换机几乎每秒都会有流表项被删除,原因是网络中有大量的流会流经交换机,而每条流的持续时间只有几秒。所以,可使用一条或多条流的丢包率近似交换机的丢包率。这样,被动测量丢包率的实时性就可以保证了。因此在该测量系统中,对于丢包率的测量方法采用后者。

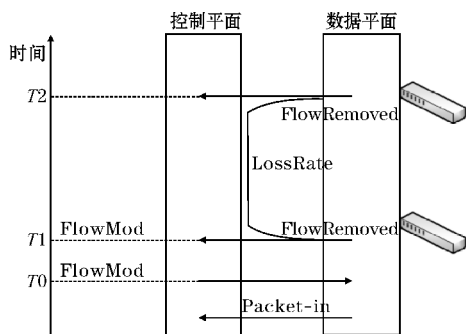


图3 丢包率测量

2.3 带宽测量

对于带宽的测量,文中沿用传统网络中主动测量方法^[12],即根据单向时延变化情况对带宽进行逼近。首先以一定速率向网络中发送测量数据包,然后调用时延测量模块,以一个较大的速率发送测量包。若单向延迟呈现增大趋势,说明当前速率大于链路带宽;若单向延迟没有变化,说明当前速率小于链路带宽。所以发送端可以根据上一次发送测量包时单向延迟的变化情况动态调整此次发送测量包的速率,直到单向延迟不再发生增大趋势为止。然后,用最近两次发送测量包速率的平均值来估计链路带宽 BW。

2.4 链路利用率测量

对于链路利用率的测量,需要先测量出链路总带宽,再与链路已用带宽进行比较,计算出链路利用率。一条链路的带宽由两个端口的能力决定,所以通过主动测量获取端口的流量,从而得到链路的流量。Open-flow 协议中可以通过统计报文获取端口、流表、流表项、组表的统计信息。控制器通过周期性下发 Port-statistics 消息获得交换机端口的统计信息,从返回的消息中可以获取到收发数据包数、字节数及持续时间。如果把两个不同时间的统计消息的字节数相减,再除以两个消息的时间差,就可以得到当前链路已使用带宽。具体步骤如下:

(1)控制器发送 request 消息到指定交换机,获取交换机指定端口的接收和发送的字节数 bytes1,记录当前时间 time1,交换机通过 reply 消息将交换机上指定端口的统计数发送给控制器。

(2)一段时间后,控制器再次向指定交换机发送 request 消息,获得交换机的接收和发送的字节数 bytes2,记录当前时间 time2。

接下来可以通过式(4)对当前链路已使用带宽进行计算。

$$usedBw = (bytes2 - bytes1) / (time2 - time1) \quad (4)$$

最后根据式(5)得出最后的链路利用率。

$$Utilization = usedBW / BW \quad (5)$$

3 仿真结果及分析

3.1 仿真参数设置

利用 floodlight 作为控制器,采用 mininet 软件搭建 SDN 网络仿真拓扑。使用工具 Iperf 来为网络设置流量,使用 NetEm 模拟网络延迟和丢包,仿真网络拓扑如图 4 所示,8 个交换机(A-G)的 11 条链路,参数设置如表 1 所示。

表 1 参数设置

参数	取值
链路 1,2 延迟	20 ms
链路 8,9 延迟	30 ms
其它链路的延迟	40 ms
链路 3,5,6 带宽	30 Mb/s
链路 9,10 带宽	50 Mb/s
其它链路带宽	40 Mb/s
链路 2,8,9 丢包率	10 %
链路 3,5,6 丢包率	30 %
其它链路丢包率	20 %

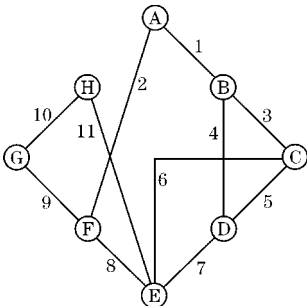


图 4 仿真网络拓扑

3.2 仿真结果

为了验证测量系统的准确性,从数据链路的时延、丢包率、链路已用流量和链路利用率入手,多次测量结果进行对比。图 5 所示为链路 2-9-10 段每一子段时延不同次数的对比,以及链路总时延与参考时延的对

比。由图可知,对于同一段子链路,多次测量结果趋于稳定,变化较小,测量总时延与参考时延曲线几近吻合。

图 6 所示为 2-9-10 段相隔 8 秒注入不同大小流量的两条流的丢包率变化情况。由图可知,在依次加大链路带宽的情况下,两条流的丢包率测量结果相差很小,并且十分接近链路实际丢包率。

图 7 所示为以 10 Mb/s 增长的速度依次向 6-11-10 段链路注入流量,得到各段链路的增长情况。当注入流量从 10 Mb/s 增长到 30 Mb/s 时,链路中已使用带宽随之增长;当流量增长到 30 Mb/s 后,由于之前已设置 6 段总带宽是 30 Mb/s,故已饱和;同理,当流量增长到 40 Mb/s 后,11 段已饱和。

图 8 所示为各段链路实际带宽与测量带宽的误差百分比。由图可知,误差保持在 2.5 % 以内,因此在误差允许范围内。测量出链路带宽和链路已用流量,即可计算出当前链路使用率。如果对链路利用率结果精度要求比较高,可使用流量工具为系统注入额外流量,这样要比程序自生成的流量稳定和易于控制。

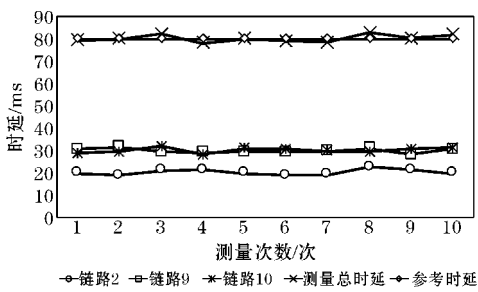


图 5 2-9-10 段单向时延

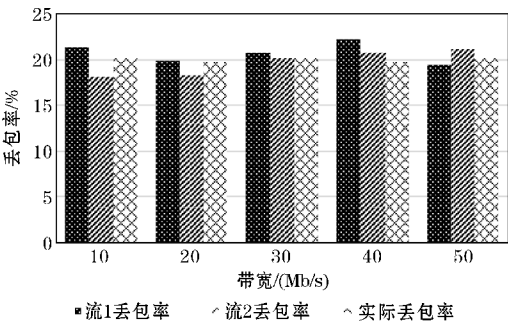


图 6 2-9-10 段丢包率

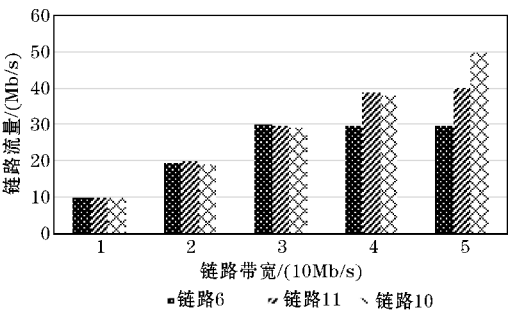


图 7 6-11-10 段链路已用流量

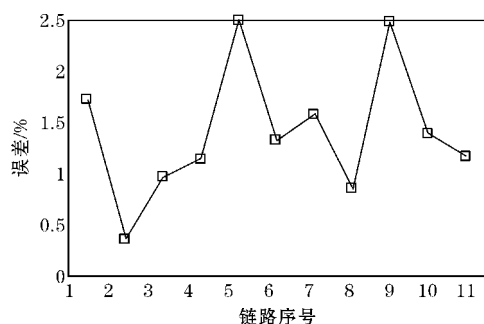


图8 各段链路带宽测量误差值

4 结束语

首先对已有的软件定义网络的测量方案进行了对比分析,根据 SDN 数据平面与控制平面解耦合的特点,借鉴传统网络中的测量思路,结合主动测量与被动测量的特点,设计并实现了 SDN 网络性能参数测量系统。最后根据多次仿真测量结果对该测量系统进行结果准确性验证,得出的测量值与参考值误差范围很小,基本上与目标值很接近。所以,在误差允许范围内,实现的系统可以正确测量性能参数。

参考文献:

- [1] 张朝昆,崔勇,唐骝骝,等. 软件定义网络(SDN)研究进展[J]. 软件学报,2015,(1):62-81.
- [2] 左青云,陈鸣,赵广松,等. 基于 OpenFlow 的 SDN 技术研究[J]. 软件学报,2013,(5):1078-1097.

- [3] 雷葆华,王峰,王茜,等. SDN 核心技术剖析和实战指南[M]. 北京:电子工业出版社,2013:90-93.
- [4] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, April 2008, 38(2):69-74.
- [5] 张冬艳,胡铭曾,张宏莉. 基于测量的网络性能评价方法研究[J]. 通信学报,2006,(10):74-79+85.
- [6] openflow[EB/OL]. <http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf>, 2011.
- [7] 谈杰,李星. 网络测量综述[J]. 计算机应用研究,2006,(02):5-8+13.
- [8] Yu M, Jose L, Miao R. Software defined traffic measurement with OpenSketch[C]. (to appear) NSDI. 2013.
- [9] Tootoonchian A, Ghobadi M, Ganjali Y. OpenTM: traffic matrix estimator for OpenFlow networks[C]. Passive and Active Measurement, International Conference, PAM 2010, Zurich, Switzerland, April 7-9, 2010. Proceedings. 2010:201-210.
- [10] Ofpeck[EB/OL]. <http://www.openflow.org/wk/index.php/Ofpeck>.
- [11] 林宇,程时端,邬海涛,等. IP 网端到端性能测量技术研究的进展[J]. 电子学报,2003,(08):1227-1233.
- [12] 杨峰,杨莘元,王恺,等. 基于 TCP/IP 的网络带宽测量方法及工具[J]. 计算机工程,2003,(18):43-45.

Design and Implementation of Performance Measurement System Used in Software Defined Network

Chen Lei-yi, Zhang Xin-you

(School of Information Science & Technology, Southwest Jiaotong University, Chengdu Sichuan 610031, China)

Abstract: Software Defined Network (SDN) is an emerging network architecture, which decouples the control plane from data forwarding plane and realizes programmable network services. The measurement of network performance parameters is one of the most important indexes to master a network operation. Based on full use of the features of SDN, and referred the measurement methods and definitions about network performance parameters in traditional network, this paper adopts the combination of active measurement method and passive measurement method, and it realizes the measurement and analysis of bandwidth, delay, link utilization, packet loss rate of SDN. Simulation results show that the measured value by performance measurement system is consistent with expected result.

Keywords: software defined network; OpenFlow; bandwidth; delay; loss packet rate