基于SDN的数据中心网络多路径流量调度算法

金 勇 刘亦星 王欣欣
(重庆邮电大学通信与信息工程学院 重庆 400065)
(重庆邮电大学移动通信技术重庆市重点实验室 重庆 400065)

摘 要 针对数据中心网络带宽利用率低、网络性能差的问题,提出一种基于SDN架构下,结合多因素的多路径流量调度算法(MSF)。算法利用SDN架构中控制与转发分离的特性以及利用控制器集中控制的方式来为数据流计算路由,首先计算出源主机和目的主机间所有可达路径中跳数最少的路径集,然后找出最短路径集中关键度最小的数条路径,最后结合流特征找出代价最低的路径作为最终流表的下发路径。实验结果表明,在不同的流量模型下,与ECMP和Hedera两种算法相比,所提算法提升了链路带宽利用率和吞吐量,减少了流量的平均往返时延,从而提高了数据中心的整体网络性能。

关键词 数据中心网络,软件定义网络,流量调度
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.06.012

SDN-based Multipath Traffic Scheduling Algorithm for Data Center Network

JIN Yong LIU Yi-xing WANG Xin-xin
(School of Telecommunication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)
(Chongqing Key Lab of Mobile Communications Technology, Chongqing University of Post and Communications, Chongqing 400065, China)

Abstract In order to solve the problems of low bandwidth utilization and poor network performance in data center networks, this paper proposed a multi-path traffic scheduling algorithm considering multiple factors(MSF) based on SDN. The algorithm utilizes the characteristics of control and forwarding separation in Software Defined Network(SDN) architecture and the centralized control of the controller to calculate the route for the data stream. Firstly, this algorithm calculates all the path sets with the shortest hops from all feasible paths between source host and destination host, then finds out the paths with the least criticality in the shortest path sets, and finally seeks out the lowest-cost path as the down-forwarding path in final flow table. Experimental results show that the proposed algorithm improves the network bandwidth utilization and throughput, and reduces the average round-trip time of traffic compared with the ECMP algorithm and Hedera algorithm under different traffic models, thus improving the overall network performance of data center.

Keywords Data center network, SDN, Traffic scheduling

1 引言

随着大数据、云计算和社交网络的迅速发展,数据中心内部的通信量呈指数级增长,对数据中心网络的带宽需求不断增加[1]。为了使数据中心网络获得更高的带宽和更好的容错性,研究人员提出了多种新型的数据中心网络体系结构[2],如Fat-Tree[3],BCube[4],DCell[5]等,它们均采用多根树型结构,提供多路径的传输方式,以获得更高的网络带宽保证网络的可靠性。

传统的等价多路径转发(Equal-Cost Multi-Path, ECMP)[6]算法被广泛地应用于数据中心网络。该算法通过对流的数据包头进行哈希运算,将数据流随机分配到多条等价的路径上进行传输,因此能够充分利用网络中大量的冗余链路,实现数据的快速转发和网络的负载均衡。然而,研究表明[7],数据中心中流的特性呈大象流和老鼠流的特性,即80%的流不超过10kB,10%的流占据绝大部分的数据带宽。研究发现,传统的ECMP算法在对小流的处理上较为有效;但对于持续时间长且带宽敏感的大流而言,ECMP可能将多个不同的大流进行哈希运算后分配到同一条链路上,这样就会产生数据流碰撞,造成链路拥塞。

到稿日期:2018-05-07 返修日期:2018-09-24 本文受长江学者和创新团队发展计划项目(IRT_16R72)资助。
金 勇(1974-),男,硕士,高级工程师,主要研究方向为移动通信,E-mail:18602340505@wo.com.cn(通信作者);刘亦星(1992-),男,硕士,主要研究方向为数据中心网络、软件定义网络,E-mail:317392661@qq.com;王欣欣(1994-),女,硕士,主要研究方向为数据中心网络、软件定义网络。

近年来,研究者们提出了一种新型的网络架构,即软件定义网络[8](Software Defined Network,SDN),它的核心思想是将控制与转发分离,采取集中控制的方式。这种新的架构给数据中心网络的多路径路由问题提供了新的思路。因此,将SDN 集中控制的思想运用于数据中心网络已成为研究的热点。例如,针对大象流识别的调度研究,Al-fates 和 Curtis 等提出了 Hedera[9] 和 Mahout[10] 两种流量管理系统。其中,Hedera 采用了两种流量调度算法,即全局首次适应算法(Global First Fit,GFF)和模拟退火算法(Simulated Annealing,SA);Mahout 采用的是全局首次适应算法。首次适应算法在路径的选取上并没有充分考虑网络中所有路径的带宽使用情况,因此无法保证网络全局动态负载的均衡。Fincher[11] 是一种基于稳定匹配的大象流调度算法,该算法首先获取大流的需求带宽和交换机剩余带宽之间的最佳匹配,如果在设定的阈值内满足大流的需求带宽,则对其进行调度。该算法的不足之处在于匹配精度上存在误差且复杂度较高,从而造成网络性能不稳定。彭大芹等[12] 提出了一种多路径路由算法,该算法分别对大小流进行调度,但在对小流的处理上,算法选取可用剩余带宽最大的路径进行路由,这样会导致众多的小流影响路径上大流的传输,进而造成链路拥塞。

综上所述,考虑到数据中心网络多路径的传输方式和流量特征,以及当前研究存在的一些不足,本文提出一种结合多因素的多路径流量调度算法,其主要考虑跳数、关键度和代价 3 个因素进行递进方式的找路。针对数据中心网络中源主机和目的主机间的通信存在无数多条路径的问题,我们通过 KSP(K Shortest Paths)算法从中选取跳数最少的路径集;然后利用 SDN 架构所具有的全网视图的特性获取交换机和流表信息,计算出路径的关键度,并从中找出关键度最小的路径集;最后结合数据中心流特征,选取链路的最大平均剩余带宽和平均时延作为代价权值,同时选取代价最低的路径作为最终数据流的下发路径。

2　多路径流调度算法的设计

2.1　算法的思想

该算法是一种基于 SDN 的适用于数据中心网络的多路径流量调度策略。它主要通过以下 3 个步骤为数据流找到最佳路径:首先,计算出源主机和目的主机之间跳数最少的路径集;然后,从跳数最短路径集中找出关键度最小的数条路径;最后,找出代价最低的路径作为数据流下发流条目的最终路径。

算法的具体实现如图 1 所示,具体描述如下:边缘交换机收到数据包时,查看自身流表项中是否有相匹配的流表,若有,则根据指令直接转发;若没有,则通过 PACKET_IN 请求控制器处理。控制器在接收到 PACKET_IN 消息后首先会计算出源主机和目的主机间所有可达路径中跳数最少的路径;然后通过周期性地获取网络的实时状态和交换机的端口信息,找出关键度最小的路径集,同时找出代价最低的路径作为最终流表的下发路径;最后向路径的交换机下发 PACKET_OUT 消息,交换机根据消息进行数据包的转发。

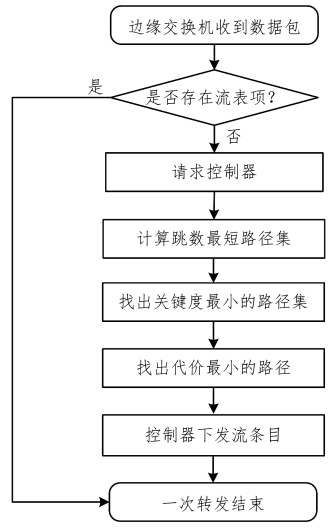

图 1　基于 SDN 的数据中心网络多路径流量调度算法的流程图

Fig. 1　Flowchart of SDN-based multipath traffic scheduling algorithm for data center network

2.2　算法的关键问题

2.2.1　链路关键度的描述

在路径计算的步骤中,需要计算出最短路径集下关键度最小的数条路径。首先,通过定义链路关键度并对其进行量化,计算出链路的平均期望负载,如式(1)所示:

$$AVE(l)=\frac{f_l(s,d)}{N_l} \tag{1}$$

其中,$f_l(s,d)$表示节点对之间通过链路 l 的所有流量之和,N_l 表示链路 l 上大流的数目。数据中心网络中大流占据绝大多数的数据带宽,因此式(1)能够很好地反映出链路的平均期望负载。可以看出,平均期望值越大,说明该链路质量越好,选择该链路的概率就越大。

然后通过链路的平均期望负载计算链路的关键度,如式(2)所示:

$$\rho(l)=\frac{AVE(l)}{C_l} \tag{2}$$

其中,C_l 表示链路 l 的总容量。可以看出,链路的关键度值越大,表示该链路发生拥塞的概率就越大,因此我们选择关键度较小的路径,以避免链路拥塞。

2.2.2　大流的检测问题

通过控制器向交换机发送 REQUEST_STATS 消息,请求获取交换机的端口统计信息和流的统计信息;交换机回复 OFPPortStatsReply 和 OFPFlowStatsReply 消息,据此便可以获取统计信息并将其保存。而流的大小则通过控制器获取的统计信息计算得来,主要通过端口的字节数变化求得。本文结合 Hedera[10] 的思想,设定当平均传输速率大于链路带宽的 10%时,将其定为大流。检测大流的伪代码如算法 1 所示。

算法 1　大流检测算法

输入:数据流 flows

输出:大流 flow

1. for flow in flows;

2.　　if flow.duration<0.1;

```
3.      flow.duration=0.1
4.   speed=(flow.byte_count * 8.0/flow.duration)/(Bandwidth)
5.   if speed>0.1;
6.      判定 flow 为大流
```

其中,第 2—3 行是为了防止在流条目传输字节数较少且生成时间极短的情况下对大流进行误判;第 4 行中乘以 8.0 的操作是进行单位转化。

2.2.3 链路代价的描述

在选出若干条关键度最小的路径的基础上,找出代价最小的路径作为最终下发数据流的路径。研究表明,数据中心流量特征主要有对带宽敏感的大象流和对时延敏感的老鼠流,因此选择带宽和时延两个指标作为一个代价权值。具体如式(3)所示:

$$C=\alpha * b_m+\beta * d_m \tag{3}$$

其中,$\alpha+\beta=1$,b_m 表示可用路径 m 的平均已占用带宽,d_m 表示可用路径 m 的平均时延。对于权重因子 α,β 的取值,因为数据中心内带宽敏感的流量占据了大部分的带宽,所以 α 的取值比 β 的取值更大一些。

2.2.4 路径计算

本文通过采用 K 最短路径算法(KSP)来寻找跳数最少的路径;然后找出关键度最小的若干条路径;最后找出代价最小的路径作为最终下发数据流的路径。控制器为数据流计算路径的伪代码如算法 2 所示。

算法 2 路径计算算法

```
输入:网络拓扑(V,E)
输出:最优路径 Paths
1.拓扑中任意主机间的路径 P
2.Ph←跳数最短路径集,Pk←关键度最小的路径集,Pc←代价最小的
  路径
3.for path in nx.shortest_simple_paths(p,weight);
4.      Ph=p.append(path)
5.ρ(l)←链路 l 的关键度,ρ(p)←路径的关键度∑ρ(l)
6.for path in Ph;
7.   if len(path)>1;
8.      for i in xrange(len-1);
9.         ρ(p)=min(∑ρ(l)[i])
10.  else
11.        Ph.append(path)
12.  Pk=Ph[ρ(p)]
13.路径代价 cost←α * bm+β * dm
14.for path in Pk;
15.  if cost_path<min_cost_of_path;
16.     min_cost_of_path=cost_path
17.     Pc=path
18.  Paths=Pc
```

其中,第 3 行中的 *nx.shortest_simple_paths* 调用 NetworkX(Python 的一个软件包)中内置的 KSP 算法,根据参数"*weight*"属性来设置权重进行最短路径的计算,本文选择"*hop*"(跳数)作为权重;第 6—12 行获得关键度最小的路径集;第 14—18 行获得代价最小的路径,并将其作为最终流表下发路径。

下面对 MSF 算法的复杂度进行简要分析。在 Fat-Tree 架构中,假设交换机端口数为 k,则 2 台主机间的可选路径为 $(\frac{k}{2})^2$ 条,流量总数为 F,从跳数最短路径集中选取关键度最小的 t 条路径,最后从关键度最小的数条路径中选取代价最小的一条路径 p。由上可得,算法为数据流计算路径的时间复杂度为 $O(F(\frac{k}{2})^2)+O(F(t)+O(F(p)))$。同理,作为对比算法的等价多路径算法(ECMP)计算路径的时间复杂度为 $O(F(\frac{k}{2})^2)$;而 Hedera 的时间复杂度因为分大流和小流的调度,所以我们假设大流总数为 N,小流总数为 M,则其时间复杂度为 $O(N(\frac{k}{2})^2)+O(M)$。综上所述,在时间复杂度方面,所提 MSF 算法比 ECMP 高,与 Hedera 相当。

3 实验验证与分析

3.1 实验环境设置

在 Linux 系统上使用 Ryu 控制器和 Mininet[13] 平台对算法的性能进行验证评估。其中,Mininet 是一种轻量级仿真工具。路由的选取全都由集中式 Ryu 控制器进行控制,在 Ryu 控制器上实施流量调度算法。通过 Mininet 构建一个 Fat-Tree 网络拓扑结构进行仿真,该结构以优越的性能在数据中心网络中得到了广泛应用。搭建的拓扑结构如图 2 所示,拓扑中的交换机均为 OpenFlow 交换机。实验设置拓扑的链路带宽为 100 Mbit/s,通过 Iperf 和 Ping 两种工具产生数据流。算法参数 α 和 β 的取值通过仿真实验分析来确定。具体的参数仿真结果如表 1 所列。

表 1 不同参数取值时算法结果的对比

Table 1 Comparison of algorithm results with different parameters

取值	$\alpha=0.5$ $\beta=0.5$	$\alpha=0.6$ $\beta=0.4$	$\alpha=0.7$ $\beta=0.3$	$\alpha=0.8$ $\beta=0.2$	$\alpha=0.9$ $\beta=0.1$
吞吐量	808.97	756.62	794.25	835.03	815.52
时延	60.48	74.16	63.5	48.23	58.93

由表 1 可得,当 $\alpha=0.8$,$\beta=0.2$ 时,算法的整体效果相对较好。因此在接下来的仿真实验中,将参数设置为 $\alpha=0.8$,$\beta=0.2$。

实验模型使用数据中心常用的通信模式,一种是 Random 模式,该模式的通信方式是主机以相同的概率向其他的主机发送数据而形成的流量模型;另外一种是 Staggered Prob(EdgeP,PodP)概率型,表示在接入层交换机以下的流量和 Pod 间的流量以不同的概率发送数据所形成的流量模型,例如"stag_0.6_0.2"表示在接入层交换机下面的流量占 60%,Pod 间的流量占 20%。

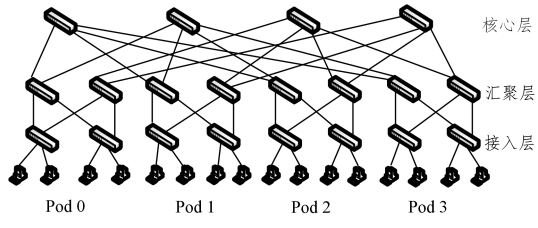

图 2 Fat-Tree 拓扑结构

Fig. 2 Topology of Fat-tree network

3.2　实验结果分析

为了验证算法的性能，选取网络平均吞吐率、标准化总吞吐量、链路带宽利用率和流量的平均往返时延作为实验评价指标。同时，为了验证算法的可行性，主要将其与 ECMP 和 Hedera 这两种算法进行实验对比分析。实验过程中，每种流量模型重复实验 15 次，并取实验结果的平均值。

主要通过以下几项指标来考量网络的性能。

(1)平均吞吐率和标准化总吞吐量

平均吞吐率是指网络在当前流量模型下所获得的单位时间吞吐量的平均值；标准化总吞吐量是指在当前流量模型下获得的总吞吐量与最大吞吐量的比值。这两个指标都能够反映流量在网络中的传输能力。选取 6 组流量模型进行仿真实验，从图 3 和图 4 中可以看出本文提出的 MSF 算法有 5 次获得了最高吞吐量。在机柜内流量占比较低的情况下，MSF 与另外两种方法获得的吞吐量有较为明显的差距；在机柜流量占比比较高时，三者的网络吞吐量相差不多。在机柜流量占比较低的情况下，MSF 与 ECMP 和 Hedera 获得的吞吐量差距比较大，这是因为 ECMP 在机柜流量较低时，大流的碰撞概率越大。Hedera 通过结合网络负载，动态地为大流做调度，并且采用 ECMP 算法对小流进行处理，因此比 ECMP 的效果好。MSF 一开始就采用 SDN 集中调度的方法对数据流选取最优路径，因此较以上两种方法获得的吞吐量更高。从表 2 中我们可以看出所提 MSF 算法的网络总吞吐量相比 ECMP 和 Hedera 均有相应的提升，比 ECMP 提升了 12.4%，比 Hedera 提升了 6.0%。以上数据有力地证明了所提 MSF 算法的可行性和有效性。

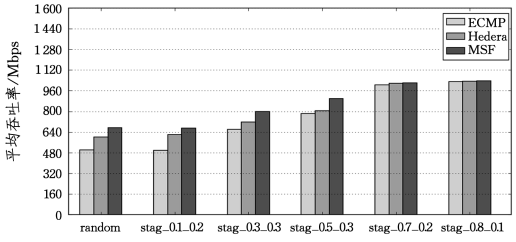

图 3　网络的平均吞吐率

Fig. 3　Average throughput of network

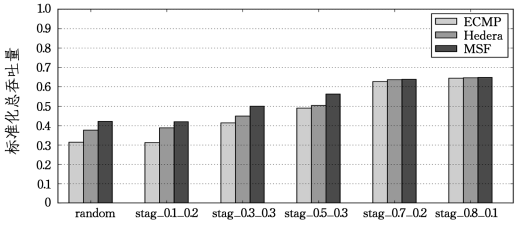

图 4　网络标准化总吞吐量

Fig. 4　Normalized total throughput of network

表 2　MSF 的总吞吐量和相对提升

Table 2　Total throughput and contrast improvement of MSF

调度方法	ECMP	Hedera	MSF
总吞吐量	4469.41	4798.19	5101.30
相对提升/%	12.4	6.0	

(2)链路带宽利用率

链路带宽利用率是指网络中每条链路带宽的具体使用情况，即每条链路实际使用到的带宽与链路总带宽的比值，该指标反映了网络资源的利用情况。同样选取 6 组模型进行实验，从图 5 中可以看到，在机柜流量占比较低的情况下，三者的带宽利用率相差较大，这是因为 ECMP 在机柜内流量占比较小的情况下，大流的碰撞概率较大，容易造成链路拥塞，导致带宽利用不充分。而 Hedera 在对大流进行调度时采用的是模拟退火算法和全局首次适应算法，而全局首次适应算法并没有考虑整个网络的带宽使用情况，因此带宽利用并不是非常充分，但带宽利用率比 ECMP 高。MSF 一开始便对数据流采用集中控制的调度，充分利用了网络中的冗余链路，因此带宽利用率较高。而在机柜流量较高的情况下，三者的带宽利用率比较接近。另外，从图 5 中可以看到 MSF 比 Hedera 和 ECMP 在链路带宽利用率低于或等于 10%时的占比要低；从表 3 中可以看出 MSF 比 Hedera 和 ECMP 在带宽利用率低于 10%的数目上都有所减少，其中比 Hedera 减少了 6.25%，比 ECMP 减少了 29.21%。综合可得，所提 MSF 算法的链路带宽利用率总体都在 10%以上，因此相对于其他两种算法，MSF 的网络资源得到了更好的利用，从而证明了该方法的有效性。

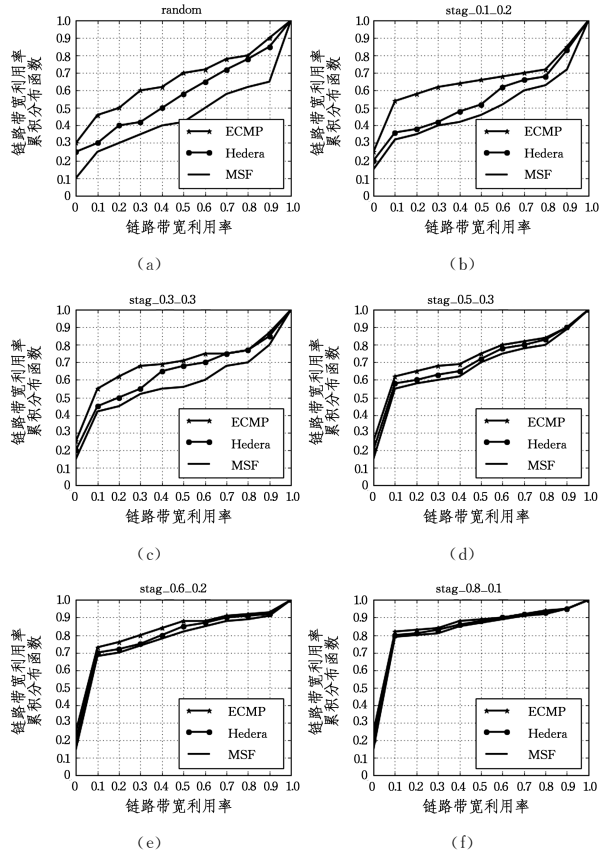

图 5　链路带宽利用率

Fig. 5　Link bandwidth utilization of network

表 3　MSF 的网络带宽利用率和相对提升

Table 3　Bandwidth utilization and contrast improvement of MSF

(单位：%)

调度方法	ECMP	Hedera	MSF
小于等于 10%的链路占比	62	51	48
相对减少	29.21	6.25	

(3)流量的平均往返时延

平均往返时延是指数据从源端到目的端的平均时延,该指标能够反映业务的体验情况,时延越大,表示体验程度越不好。选取 5 组模型进行实验观察,从图 6 可以看到所提 MSF 算法所获得的平均往返时延最低。总体上来说,在前两种模型下,ECMP 和 Hedera 的平均往返时延较大,当机柜流量较大时三者的时延都相对较低。在前两种模型下,因为 ECMP 在机柜流量相对较小时,大流产生碰撞的次数较多,造成了一定的拥塞,而在数据中心网中有很多小流,这样导致小流的传播速度也相对减慢,所以时延相对较大。Hedera 在对大流进行调度时采用全局首次适应算法,对小流采用 ECMP 进行处理,这样在一定程度上避免了大流对小流的传输影响,因此时延比 ECMP 要低。而所提 MSF 算法一开始就对数据流采取 SDN 集中调度的方式,选取最优路径,并且在算法中结合数据中心网络流特征,考虑带宽和时延两个因素,这样对数据中心网络中的大流和小流都做到了较为合理的路由,因此时延是最低的。当机柜流量较小时,3 种方法都获得了较低的时延。在机柜流量较大的情况下,ECMP 则等价选取多路径进行传输,Hedera 通过对大流的合理调度降低了链路拥塞的概率,因此两者的平均往返时延都相应降低;而本文的 MSF 算法获得了更低的平均往返时延。

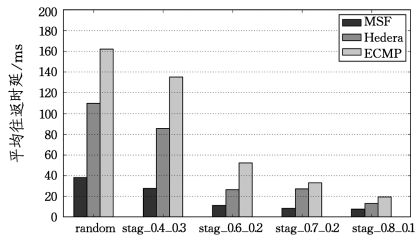

图 6 流量的平均往返时延

Fig. 6 Average round-trip time of traffic

结束语 本文在软件定义网络的架构下,提出了一种面向数据中心网络的多路径流调度算法。该算法结合多因素进行多级路径的筛选,找出最佳路径;采用 SDN 控制器集中控制的方式对数据流进行调度,通过其具有的全局视图特征,从跳数、关键度和代价 3 个方面为数据流找出最优路径。与 ECMP 和 Hedera 相比,所提算法在吞吐量、链路带宽利用率等指标方面有一定的提升。下一步将考虑其他类型的拓扑结构,同时考虑不同网络负载情况下的网络性能。

参 考 文 献

[1] CHEN Y,JAIN S,ADHIKARI V K,et al. A first look at inter-data center traffic characteri-stics via Yahoo! datasets[C]// INFOCOM,2011Proceedings IEEE. Shanghai:IEEE,2011:1-620-1628.

[2] WEI X L,CHEN M,FAN J H,et al. Architecure of the Data Center Network[J]. Journal of Software,2013,24(2):295-316. (in Chinese)
魏祥麟,陈鸣,范建华,等. 数据中心网络的体系结构[J]. 软件学报,2013,24(2):295-316.

[3] VAHDAT A,AL-FARES M,LOUKISSAS A. scalable commodity data cente-r network architecture[J]. Acm Sigcomm Computer Communication Review,2008,38(4):63-74.

[4] GUO C,LU G,LI D,et al. BCube:a high per-formance,server-centric network architecture for modular data centers[J]. Sigcomm,2009,39(4):63-74.

[5] GUO C,WU H,TAN K,et al. Dcell:a scalab-leand fault-tolerant network structure for data centers[J]. Acm Sigcomm Computer Commu-nication Review,2008,38(4):75-86.

[6] HOPPS C E. Analysis of an Equal-Cost Multi-Path Algorithm[J]. Journal of Allergy & Clinical Immunology,2000,109(1):S265.

[7] BENSON T,ANAND A,AKELLA A,et al. Understanding data center traffic characteristics[J]. Acm Sigcomm Computer Communicat-ion Review,2010,40(1):92-99.

[8] ZHANG C K,CUI Y,TANG H Y,et al. State of the Art Survey on Software-Defined Netw-orking(SDN)[J]. Journal of Software,2015,26(1):62-81. (in Chinese)
张朝昆,崔勇,唐翯祎,等. 软件定义网络(S-DN)研究进展[J]. 软件学报,2015,26(1):62-81.

[9] AL-FARES M,RADHAKRISHNAN S,RAGHAVAN B,et al. Hedera:dynamic flow sch-eduling for data center networks[C]// Usenix Symposium on Networked Systems Design and Implementation. San Jose:DBLP,2010:281-296.

[10] CURTIS A R,KIM W,YALAGANDULA P. Mahout:Low-overhead datacenter traffic ma-nagement using end-host-based elephant detection[C]// IEEE INFOCOM. Shanghai:IEEE,2011:1629-1637.

[11] ZHANG Y,CUI L,ZHANG Y. A stable matching based elephant flow scheduling algorithm in data center networks[J]. Computer Networks,2017,120:186-197.

[12] PENG D Q,LAI X W,LIU Y L. Multi-path Routing Algorithm for Fat-Tree Data Center Networks Based on SDN[J]. Computer Engineering,2018,44(4):41-45,65. (in Chinese)
彭大芹,赖香武,刘艳林. 基于 SDN 的胖树型数据中心网络多路径路由算法[J]. 计算机工程,2018,44(4):41-45,65.

[13] Mininet[EB/OL]. http://www.mininet.org.