不断增长的数据中心网络规模和业务需求对数据中心网络负载均衡提出了严峻的挑战,而基于 SDN 新型网络架构的出现给负载均衡研究带来新的思路和机遇.SDN 网络中的控制器能够获取全局网络的拓扑、节点、链路以及流信息,进而可以对网络流进行集中控制和调度.SDN 网络控制平面和转发平面的解耦分离实现底层交换机资源的抽象和高速转发,并且控制器提供的北向接口可以方便开发人员根据业务需求开发新的功能和研究创新[1,2].利用 SDN 网络架构的特点可以实现网络流量的灵活细粒度的集中控制,进而通过流管理和调度实现网络负载均衡.考虑到数据中心网络中大多数是小流且仅占用 10%左右的网络带宽,而少数的大流却占用了 80%的左右的网络带宽[3,4],所以对于数据中心网络负载均衡的实现主要是考虑大象流的调度.当前的大多数数据中心网络研究主要集中在流量分析以及大象流发现方面,而对于大象流的调度研究较少并且存在链路带 0 宽使用碎片化的问题.带宽碎片化问题可能导致多条路径剩余带宽之和能够满足当前流带宽需求,而无其中任何一条路径剩余带宽能够满足流带宽分配需求,这就可能造成潜在的网络拥塞.流调度算法不但要考虑当前待调度的流路径选择问题而且要全局考虑网络链路带宽剩余碎片化问题,并且集中的剩余链路带宽可以分配更多的大流,减少链路拥塞的发生,提高整体网络带宽利用率.针对带宽碎片化问题,本文提出一种最大概率路径调度算法(maximum probability path scheduling algorithm,简称 MPP_SA),算法不但考虑了网络带宽利用率以及流带宽大小而且考虑了带宽碎片问题,在带宽利用率和带宽碎片化之间做了权衡.

本文第 1 节对现有流量调度算法的研究进行对比总结,并从流带宽、网络带宽利用率以及带宽碎片化方面分析每种方法的不足.第 2 节对提出基于 SDN 的数据中心流量控制框架.第 3 节提出最大概率路径流调度算法,分析算法的流程以及优势.第 4 节对现有并常用的流调度算法进行实现并与提出的最大概率路径流调度算法进行吞吐量和平均延迟的对比.第 5 节对提出的流调度算法进行总结以及展望.

1 相关研究

网络流量合理的调度可以提高链路带宽的利用率,缓解网络拥塞,为应用提供有效的带宽支持进而可以保证服务质量,满足用户的需求.传统 IP 网络中使用最短路径优先路由协议(如 OSPF)进行链路间负载的分配,交换机之间通过交换静态的链路权重计算出转发的最短路径,然而,基于最短路径路由策略不能将流量均匀分配到多个后续路径上,无法有效地进行负载均衡.

ECMP(equal-cost multi-path)[5,6]通过对流的数据包头部进行哈希取模运算,将数据流映射到不同的转发路径,采用类似于 ECMP 的流调度方案比如 VLB(valiant load balancing)[7],可以将到达同一个节点的不同流分散到后续的多条等价可用路径,算法流程如图 1 所示.然而,该类流分配调度方案并没有考虑流大小、持续时间和链路带宽剩余情况,可能多个不同长命(持续时间长)大象流被哈希分配到同一条链路上,就会造成链路拥塞.ECMP 和 VLB 算法虽然效率很高但不能很好地进行负载均衡,所以该类算法仅仅能应对网络中存在大量的小流,而大象流很少的情况.

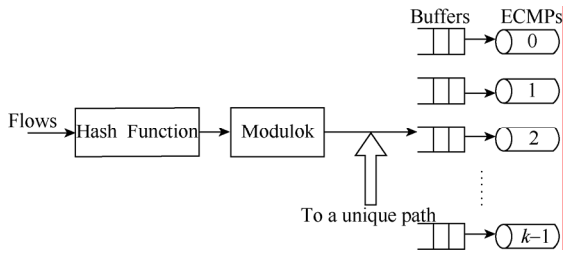

Fig.1 ECMP algorithm of flow scheduling process

图 1 ECMP 算法流调度流程

SDN 网络的控制器能够计算流的转发路径根据流大小和链路负载信息,实现多路径之间的负载均衡.基于 SDN 的流调度与传统流调度相比的主要优势有:集中式转发决策,并且路径转发分配考虑流大小特征和可用链路带宽.基于大象流识别调度研究[8]提出 Hedera 流量管理系统,通过结合网络负载动态的为大象流计算后续路径,实现网络的负载均衡.文中提出两种流调度算法全局首次适应算法(global first fit)以及模拟退火算法(simulated annealing)并对这两种算法进行了实验结果对比,首次适应算法的思想是:已知流带宽,根据流目的接入层交换机与流起始交换机确定可选路径,然后线性的搜索可选路径,如果搜寻到某条路径上链路剩余最小带宽能够满足流带宽需求,然后将该路径作为流转发路径.该算法并没有充分考虑满足流带宽需求的所有路径的

带宽使用情况,并且可能导致链路剩余带宽的零碎化,所以不能很好地实现全局动态负载均衡.

文献[9]提出一种针对多路径的胖树型网络的动态负载均衡算法(dynamic load balancing,简称 DLB),算法主要采用贪心策略的思想进行链路寻找.通过深度优先递归的策略,DLB 算法从源节点开始进行对比并选取与其相连的链路剩余带宽最大的链路,并将链路的另一节点作为下一次搜寻的起点,通过不断地递归遍历确定一条路径.通过仿真实验作者验证了 DLB 算法,并与传统负载均衡算法进行了性能评估,对于胖树网络拓扑 DLB 算法在带宽利用率和网络传输延迟方面表现出很高的性能.DLB 算法只是根据局部链路剩余带宽选优而不是从全局把握路径剩余带宽,采用这种算法选择的路径很可能不是全局最优的并且可能会导致被选路径的部分链路拥塞,这样不但不能达到负载均衡的效果反而会使网络整体的性能下降.

为了解决链路带宽碎片问题,考虑借鉴磁盘调度算法中的最佳适应(best fit)算法.采用 Best Fit 算法进行流调度的思路是:控制器根据流信息获得源接入交换机和目的接入交换机,然后确定能满足流带宽需求的所有路径,将路径上最小链路带宽进行排序,把最小链路带宽的路径作为备选路径.算法不但考虑了链路带宽剩余情况和流带宽大小,而且考虑流分配后链路带宽剩余,减少了链路剩余带宽碎片.因为该算法是优先选择链路带宽剩余较小的路径进行流分配,所以可能导致剩余带宽较小的路径被分配的流越来越多,而带宽剩余较大的路径会被分配的流很少,这样就会造成网络中一些链路负载较重而有些链路负载很轻甚至空载.所以虽然该算法考虑了链路带宽碎片化问题但仍可能造成链路负载不均衡的问题.

2 基于 SDN 的数据中心流量控制框架

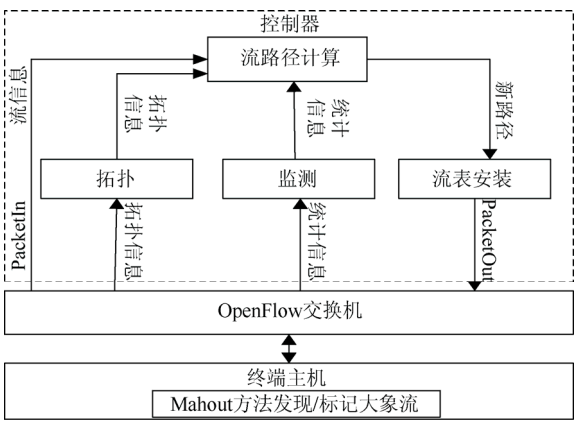

Fig.2 Data center flow control framework based on SDN

图 2 基于 SDN 的数据中心流量控制框架

基于SDN的数据中心流量控制框架包括3个核心部分:1) 在端主机处采用Mahout方法进行大象流的发现并标记(比如修改 TCP 协议的 Tos 字段).2) 根据拓扑信息和链路带宽情况,使用 ECMP 和 MPP_SA 算法对数据中心的流路径进行计算,MPP_SA 主要是为大象流计算路径.3) 计算好的路径以流表的形式下发到 Openflow 交换机,流量整体框架如图 2 所示.

大象流发现有大量的相关研究,并且有很多的方案可以使用.我们可以选择在接入层交换机处进行大象流发现并标记,比如采用 DevoFlow 和 Hedera,也可以在端主机处,比如采用 Mahout[10],这两种情况都是通过判断传输的字节数据是否达到预先设定的阈值(DevoFlow 是 1 MB ~100 MB,Hedera 是网卡带宽的 10%,Mahout 是 100 KB)进行大象流标记.文献[10]对不同的方法进行了充分的对比,端主机处发现大象流 Mahout 是最有效的方法,所以我们的框架也是采用 Mahout 方法进行发现并标记大象流.Mahout 方法充分利用了端主机 shim 层来监视 TCP 套接字缓存区,当缓冲区字节数在给定的时间超过了提前设定的阈值时 shim 层就会鉴别并标记当前流为大象流.

SDN控制器通过链路发现协议(link layer discovery protocol,简称LLDP)协议进行链路发现,然后根据发现协议搜集的信息识别和管理网络拓扑结构.SDN 控制器存储的链路拓扑信息为计算流路径模块提供支持.框架中的检测模块主要是用来查询、整合和存储 openflow 交换机的统计信息然后用于大象流路径选择(选择路径剩余带宽比较适合的路径).通过该模块控制器会定期的向 openflow 交换机发送 read-state 消息获取每个流表、每个流实体以及每个端口的统计信息.信息以快照的形式存储并且每个快照用系统时间作为其唯一标识,最终将标识比较接近的进行整合计算.流路径计算模块首先对流信息进行判断如果不是大象流就根据流信息存储的拓扑信息采用 ECMP 算法为流计算路径,如果是大象流则会根据拓扑信息和链路带宽情况采用 MPP_SA 算

法为流计算比较合适的路径.最后控制器将流路径以流表的形式下发给 OpenFlow 交换机.

3 面向 SDN 数据中心网络最大概率路径流量调度算法

基于 SDN 的数据中心大象流调度分配有两种不同的场景:一是大象流到达接入层交换机,没有匹配到流表中的任何流实体,然后通过 PacketIn 消息通告控制器,控制器结合流调度算法并且根据流信息为大象流计算一条路径,最后下发流表到 OpenFlow 交换机[11];二是如果链路发生拥塞,交换机通告控制器发生拥塞的端口[12],控制器通过交换机获取拥塞链路上的流信息,然后对占用链路带宽比例比较高的大流进行调度.两种不同的场景都是 SDN 控制器结合流调度算法对大象流进行调度,怎样调度大象流既能保证其带宽需求又能为其他流预留带宽减少带宽碎片是概率路径调度算法所解决的问题.

3.1 相关定义

定义 1(网络链路). 网络中任意相连节点之间的连通路径是网络链路或链路,比如两交换机之间的链路或者交换机与端主机之间的链路,用字母 L 表示一条网络链路.

定义 2(网络路径). 网络中从以源节点到目的节点之间的通路是网络路径或路径,一条网络路径可能包含一条或者多条链路,用字母 P 表示一条网络路径.

定义 3(带宽比). 链路剩余带宽与流带宽的比值称为带宽比,用字母 p 表示带宽比.

定义 4(路径概率). 满足流带宽需求的路径可能被选择的概率,用字母 λ 表示路径概率值.

我们提出的 MPP_SA 算法首先根据待调度流信息计算能满足其带宽需求的路径集合,然后计算路径集合中每一条路径的最小链路带宽,而后用流带宽与每一个路径的最小链路剩余带宽作比较计算其带宽比,最后根据所有路径带宽比计算每一条路径的路径概率.最后 SDN 控制器利用概率机制(比如轮盘赌算法)选择路径,概率较大的路径被选择的机会较大,而路径最小链路剩余带宽较大的虽然概率比较小但同样会被选择.MPP_SA 算法缓解了 Best Fit 算法因路径最小链路剩余带宽越小的被调度分配的流越多,而其他路径负载很小的情况.通过概率路径算法对负载均衡和带宽碎片进行了一个合理的权衡.

3.2 MPP_SA算法

综合考虑链路带宽利用率和链路带宽碎片问题,我们提出 MPP_SA 算法,描述如下:

Step 1. SDN 控制器通过 OpenFlow 交换机获得流相关信息.

Step 2. 控制器通过一定的方式对流进行判断,判断要是否是大象流,然后选择不同的方法进行路径的分配.对于大象流的发现可以使用端主机判断并设置标识的方式、SDN 控制器主动采样分析的方式或者流收集器采样收集然后控制器分析的方式.

Step 3. 如果不是大象流,根据流的的信息和控制器维护的全局拓扑信息可以获得流的可选路径,然后使用 ECMP 算法为其任意选择一条路径.

Step 4. 如果控制器判断流 F 是大象流,则选择概率路径算法为其计算一条满足流带宽需求的路径.

(1) 根据流 F 信息确定与源和目的相连的接入层交换机 S_1 和 S_2.

(2) 根据控制器维护收集的网络拓扑信息选择 S_1 和 S_2 之间的所有路径集 $P\{P_1,P_2,\ldots,P_n\}$ 其中,$P_i\{L_1,L_2,\ldots,L_n\}$,$L_i$ 为路径 P_i 上的一条链路.

(3) 在集合 C 中选择路径上链路剩余带宽能满足流 F 带宽 Bandwidth 需求的路径,从而得到能满足流 F 带宽需求的路径集合 P_1,如果没有满足其带宽需求的路径则转到第 3 步继续使用 ECMP 算法计算路径.

(4) 从集合 P 中计算每一条路径上的剩余带宽最小的链路,从而得到集合 $L\{L_1,L_2,\ldots,L_n\}$,其中,$L_i=P_i\{L_1,L_2,...,L_n\}_{\min}$.

(5) 计算集合 L 中每条路径上链路剩余最小的带宽与流 F 带宽 Bandwidth 的带宽比,从而得到集合 $p\{p_1,p_2,...,p_n\}$,其中,p_i=Bandwidth/value(L_i).

(6) 计算最终流被分配到每条路径是的路径概率集合为 $\lambda=\{\lambda_1,\lambda_2,\ldots,\lambda_n\}$,其中 $\lambda_i=p_i/(p_1+p_2,\ldots,+p_n)$.

Step 5. 控制器通过计算为流 F 选择了一条路径,然后将流表下发到 OpenFlow 交换机,流 F 匹配流表项然后根据动作进行数据转发或者将流 F 剩余数据匹配新流表项进行转发.算法伪码见表 1.

Table 1 Flow scheduling algorithm pseudo code

表 1 MPP_SA 结合 ECMP 流调度算法伪码

```
Input  F:flow
1   if F.assigned then
2       return collectionProp(1); //用 1 表示使用流 F 之前分配的路径
3   propSum=0;
4   foreach p ∈ P_src→dst do
5       if p.used + F.rate < p.capacity then
6           collectionP.add(p); //所有能满足流 F 带宽需求路径集合
7   if collectionP is empty then
8       return collectionProp(0); //用 0 表示未找到能满足流 F 带宽的路径
9   foreach p ∈ collectionP do
10      foreach l ∈ p do
11          if l.remainder < p.minLink
12            p.minLink←l.remainder;
13      p.p←F.bandwidth/p.minLink;
14      sumProp+=p.p;
15  foreach p ∈ collectionP do
16      p.prop←p.p/sumProp;
17      collectionProp.add(p.prop);
18  return collectionProp; //所有能满足流 F 带宽需求路径概率集合
```

4 实验验证

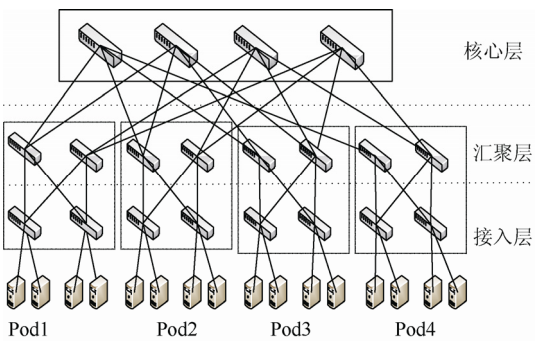

Fig.3 The fat tree topology graph

图 3 实验采用的胖树拓扑结构图

我们在轻量级仿真工具 Mininet[13]上进行实验验证,将 MPP_SA 算法在开源控制器 Floodlight[14](其稳定性、易用性已经得到 SDN 专业人士以及爱好者们的一致好评,并因其完全开源所以成为目前最流行的 SDN 控制器之一)上实现,同时为了比较算法性能,我们也实现了 DLB、ECMP 和首次适应算法(GLB).我们采用数据中心网络使用比较多、认可度较高的胖树网络拓扑进行实验仿真,Fat-Tree 拓扑结构简单、易于部署而且相对安全稳定、易于扩展.模拟实验拓扑如图 3 所示,其中的交换机均为 OpenFlow 交换机.

由于在仿真实验中缺少从实际数据中心网络中获取的流量日志信息,因此本实验使用文献[15]中研究工作组所提供的一组体现数据中心网络各种通信模式的测试套件作为本实验流量产生模式.该测试套件中主要体现了下列数据中心网络常见的通信模式:(1) Random:在该模式下,一台仿真主机节点会以相同的概率向网络中其他的仿真主机节点发送数据.(2) Stride(i):仿真主机节点会以向编号为$(x+i)$mod(host sum)发送数据,其中 x 为主机的编号.(3) Staggered Prob(P_{edge},P_{pod}):仿真主机向连接同一接入层交换机的其他主机以 P_{edge} 概率发送数据,而向在同一 Pod 的主机以 P_{pod} 概率发送数据,最后主机会以 $1-P_{edge}-P_{pod}$ 概率向其他剩余主机发送数据.网络延迟和吞吐量是评价网络性能的重要参数,能够反应网络的拥塞程度,因此我们分别从吞吐量和延迟两方面对算法进行了对比.通过增加大流所占比例来增大网络负载,图中显示不同负载情况下在不同调度算法下产生的网络延迟情况.

采用不同算法网络的平均延迟如图 4 所示,通过增多大象流的数目来增大网络负载,网络平均延迟不断的增大,而 DLB 算法增加的比较缓慢但负载相同时延迟比较大,相同负载下使用我们的算法所产生平均延迟最低.我们的流调度框架使用 ECMP 算法能够快速对小流进行路径分配,而使用 MPP_SA 算法能够分配调度更多的大流,所以其性能相对较好.

从图 5 中可以看出当负载较轻的时候 3 种算法的吞吐量相差并不大,而通过增加大象流的比例增大负载时,使用 3 种算法所产生的吞吐量之间的差距越来越明显.当负载达到 0.8 时网络吞吐量达到最大,而链路负载超过 0.8 时网络会出现拥塞,从而导致网络延迟和丢包,所以吞吐量也随着下降.

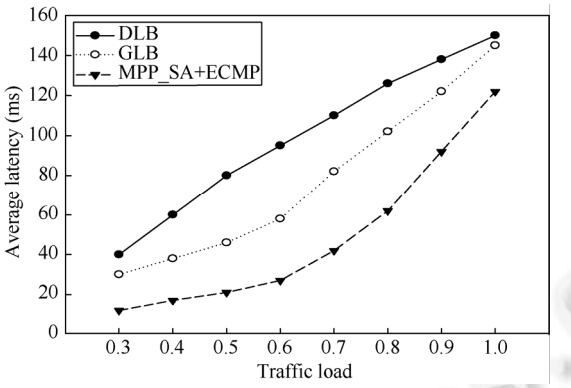

Fig.4 Network average delay using different scheduling algorithms

图 4 使用不同调度算法的网络平均延迟

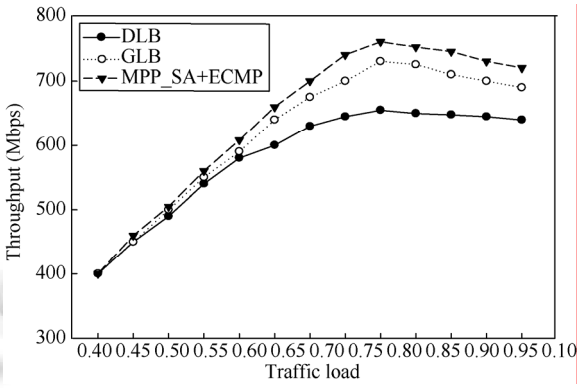

Fig.5 Network average throughput using different scheduling algorithms

图 5 使用不同调度算法的网络平均吞吐量

通过模拟实验可以看出,MPP_SA 算法不论是在减少网络延迟还是增大网络吞吐量性能表现都是较其他两种算法优秀.因为 ECMP 算法在分配调度非大象流比较高效所以在负载较轻的时候延迟比较低,而 MPP_SA 算法在调度大象流实现负载均衡、增大吞吐量性能较好.我们使用 MPP_SA 和 ECMP 算法对数据中心网络流路径分配,从而可以很好地实现提高网络带宽利用率和吞吐量,进而实现提高数据中心网络的整体性能.针对网络拥塞而发生的流进行调度,其原理一样,通过设置交换机缓存队列阈值,当队列大小超过阈值时主动向控制器通告拥塞端口信息,控制通过维护的拓扑和流信息,对其中的大象流进行调度.

5 结束语

论文针对数据中心网络大象流调度提出一种概率路径算法,并且考虑到控制器开销和调度延迟,在开源控制器 Floodlight 中结合 ECMP 算法进行了实现,对非大象流使用 ECMP 而大象流采用 MPP_SA 算法.MPP_SA 算法为每个满足流带宽需求的路径计算概率,并且路径上链路最小剩余带宽越接近流带宽的路径概率越大,最后按照概率进行路径选择.算法不但考虑了流调度可能产生带宽碎片问题而且考虑了流调度负载均衡.通过模拟仿真实验,MPP_SA 与 ECMP 算法对数据中心网络流调度进行了验证,结果显示无论在网络流吞吐量还是网络延迟方面都相对 DLB 和 GLB 算法性能较好.

References:

[1] Zuo QY, Chen M, Zhao GS, Xing CY, Zhang GM, Jiang PC. OpenFlow-Based SDN technologies. Ruan Jian Xue Bao/Journal of Software, 2013,24(5):1078−1097 (in Chinese). http://www.jos.org.cn/1000-9825/4390.htm [doi: 10.3724/SP.J.1001.2013.04390]

[2] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. Openflow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008,38(2):69−74.

[3] Benson T, Akella A, Maltz D. Network traffic characteristics of data centers in the wild. In: Proc. of IMC, 2010.

[4] Benson T, Anand A, Akella A, Zhang M. Understanding datacenter traffic characteristics. In: SIGCOMM WREN Workshop, 2009.

[5] Lin W, Liu B, Tang Y. Achieving optimized traffic sharing over equal-cost-multi-paths. Sciencepaper Online. http://www.paper.edu.cn

[6] Hopps C. Analysis of an equal-cost multi-path algorithm. RFC 2992, Internet Engineering Task Force, 2000.

[7] Greenberg A, Jain N, Kandula S, Kim C, Lahiri P, Maltz D, Patel P, Sengupta S. VL2: A scalable and flexible data center network. In: Proc. of the ACM SIGCOMM. 2009. 51–62.

[8] Al-Fares M, Radhakrishnan S, Raghavan B, Huang N, Vahdat A. Hedera: Dynamic flow scheduling for data center networks. In: Proc. of the NSDI. 2010.

[9] Li Y, Pan D. OpenFlow based load balancing for fat-tree networks with multipath support. In: Proc. of the ACSD. 2012.

[10] Curtis AR, Kim W, Yalagandula P. Mahout: Low-Overhead datacenter traffic management using end-host-based elephant detection. In: Proc. of the 30th IEEE Int'l Conf. on Computer Communications. 2011. 1629–1637.

[11] The OpenFlow Switch Consortium. http://www.openflowswitch.org/

[12] Li L, Fu BZ, Chen MY, Zhang LX. Nimble: A fast flow secheduling strategy for OpenFlow networks. Chinese Journal of Computers, 2015,38(5):1056−1068 (in Chinese).

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

[14] Floodlight openflow controller. http://www.projectfloodlight.org/ floodlight/

[15] Al-Fares M, Loukissas A, Vahdat A, Scalable A. Commodity data center network architecture. In: Proc. of the ACM SIGCOMM. 2008.

附中文参考文献:

[1] 左青云,陈鸣,赵广松,邢长友,张国敏,蒋培成.基于 OpenFlow 的 SDN 技术.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]

[5] 林伟,刘斌,唐毅.等价多路径间基于 LRU Cache 和计数统计的流量分配调度算法研究.中国科技论文在线. http://www.paper.edu.cn

[12] 李龙,付斌章,陈明宇,张立新.Nimble:一种适用于 OpenFlow 网络的快速流调度策略.计算机学报.2015,38(5):1056−1068.

陈琳(1975－),女,福建陇海人,博士,主要研究领域为数据中心网络资源管理,网络自动配置技术,大数据分析与融合.

张富强(1990－),男,硕士,主要研究领域为网络优化,网络管理.