doi:10.11959/j.issn.1000-436x.2016071

软件定义数据中心网络混合路由机制

蔡岳平，王昌平
（重庆大学通信工程学院，重庆 400030）

摘 要：针对数据中心网络流量大小分布不均匀、传输性能需求不相同的特征，提出了面向传统树型数据中心网络结构的软件定义混合路由机制 SHR（software-defined hybrid routing）。SHR 通过统计计算将数据流分为大流和小流，为满足其不同的传输性能需求，对大流采用自适应路由算法，对小流采用流量无视路由算法。SHR 在 OpenFlow 机制的基础上，将部分控制权从控制器下放至交换机，以减轻网络的额外负载。在 Fat-Tree 网络拓扑结构上建立流量模型进行性能分析与仿真实验，结果表明，与传统的等价多路径转发 ECMP 算法相比，SHR 能够提高网络吞吐量，降低数据流丢弃率和分组端到端时延，同时减轻网络的额外负载。

关键词：云计算；数据中心网络；软件定义网络；路由算法；开放流协议
中图分类号：TP393 **文献标识码**：A

Software defined data center network with hybrid routing

CAI Yue-ping, WANG Chang-ping
(College of Communication Engineering, Chongqing University, Chongqing 400030, China)

Abstract: In the current data center networks, the flow size distribution is not uniform and the transmission performance requirements of elephant flows and mice flows are different. To address this issue, a software-defined hybrid routing (SHR) scheme was proposed. SHR differentiate data flows by statistical calculat nt flows utilize the adaptive routing algorithm while the mice flows use the oblivious routing algorithm. SHR extends the OpenFlow scheme by offloading some basic functions such as flow statistical detection and mice flow forwarding to switches to reduce the switch-controller interaction overhead. Performance evaluations of SHR were carried out using the fat-tree network topology. Results show that SHR can effectively increase network throughput and reduce the flow dropping rate as well as packet delay compared with the traditional ECMP algorithm.

Key words: cloud computing, data center network, software defined network, routing algorithm, OpenFlow

1 引言

随着社交网络、搜索引擎、在线地图等数据密集型互联网应用的快速发展，作为云计算的基础设施和数据处理平台，数据中心内部的通信量正以指数级的速度增长，对数据中心网络的带宽需求不断增加[1]。传统的数据中心网络架构普遍采用树型分层结构[2]，典型的拓扑将交换机分为 3 层进行互联，分别为边缘层交换机、汇聚层交换机和核心层交换机。传统用于数据中心网络的典型树型结构[3]存在着可扩展性差、部署代价高和单点失效等问题，无法满足数据中心内部数据密集型应用业务对网络带宽的需求。近年来研究人员提出了许多新型的数据中心网络体系结构[2, 4~9]，如 Fat-Tree[2]、BCube[4]、DCell[6]、MDCube[10]等，它们均使用数量充裕的网络设备和链路提供路由转发服务，以获

收稿日期：2015-05-05；修回日期：2015-08-16
基金项目：国家自然科学基金资助项目(No.61301119)；教育部高等学校博士学科点专项科研基金资助项目(No.20120191120025)；教育部留学归国人员启动基金资助项目(No.1020607820140002)
Foundation Items: The National Natural Science Foundation of China (No.61301119), Research Fund of Young Scholars for the Doctoral Program of Higher Education of Ministry of Education (No.20120191120025), Scientific Research Foundation for the Returned Overseas Chinese Scholars of Ministry of Education (No.1020607820140002)

得更好的网络性能和可靠性。

在传统树型分层结构中，广泛使用等价多路径转发 ECMP（equal-cost-multipath）算法[11]进行路由。ECMP 针对到达同一目的地址的数据流量，当存在多条可用等价最佳路径的情况下，采用静态散列将数据流量均匀分配到多条等价路径上完成路由。因此，能够充分利用树型分层结构中大量的冗余链路，实现数据的快速转发和网络的负载均衡。同时，ECMP 具有相对协议负载低、配置方便、故障恢复机制良好等优点。然而，研究人员分析发现，数据中心网络中大部分的数据流量携带在小部分的数据流中[12, 13]。90%的数据流大小不超过 1 MB，99%的数据流大小不超过 100 MB，而 90%的数据流量都集中在大于 100 MB 的数据流中。在应用中，大小不同的数据流其传输性能需求有所不同，小流对时延要求较高，而大流则对吞吐量要求较高[14]。文献[3]研究发现，传统的 ECMP 算法在针对小流时较为有效，而对于持续时间较长的大流，ECMP 可能将多条大流散列到同一条链路上，这会造成数据流的碰撞，形成网络瓶颈，导致网络负载的不均衡，同时造成了冗余链路的浪费。由于 ECMP 在对大流路由前未检测网络负载状况，造成网络拥塞，降低了网络吞吐量。

针对数据中心网络的路由和流调度问题，研究人员提出的解决方案主要有以下几种。1) 采用固定的转发规则，例如 Fat-Tree 架构使用两层查找表的转发方式，使每个主机之间的流量转发路径都是固定的，此种方法快速、便捷，但是无法保证网络的负载均衡。2) 采用随机转发的方式，例如 VL2[5] 给出了 2 种路由方案，VLB 方案将流量随机转发给中间节点，再由中间节点完成路由，而 ECMP 采用随机的流量分发方式来均衡网络流量分配，如前文所述，这种方法对突发大数据流易造成拥塞。3) 使用集中控制策略通过全局的控制器来进行路由选择。随着软件定义网络 SDN（software defined network）技术的快速发展，利用 SDN 集中控制的思想和技术来解决数据中心网络的路由问题已经成为研究的热点。例如 Hedera[15]、Mahout[3]通过监测数据中心网络中的流，用集中式的控制器为流计算路径。然而数据中心网络中存在大量的小流，SDN 在实现控制网络以及信息统计的过程中产生了大量的额外负载，同时增加时延敏感的流的时延[16]以致错过截止时间。4) 最小化小流的完成时间。小流对时延更敏感，研究人员通过最小化其完成时间来降低时延。D^3[16]是一种截止时间可知的流控制协议，根据数据流的截止时间控制其传送速率，能够有效改善小流的时延，提高网络吞吐量。然而该方案中到达较迟的小流可能会被阻塞在瓶颈链路处。PDQ[17]是一种优先调度截止时间较短的流调度协议，该协议的目标是在截止时间内完成流传送的同时实现数据流平均竞争时间的最小化。该方案的问题在于流之间竞争的公平性较差。文献[18]优先考虑延迟敏感的数据流，通过增加一个新的跨层网络协议栈来减少流完成时间的长尾巴特性。由于数据中心小流的数量大，这种方案会增加计算的复杂度。此外，针对数据中心网络不同的性能需求，近年来研究人员提出了一些解决方案。文献[19]提出一种适合于数据中心网络规则拓扑的通用路由算法 FAR, FAR 借助拓扑结构的规则性简化路由学习，同时在交换机处引入新的负路由表来减少路由表项的数量。文献[20]针对基于流的路由策略带来较低的网络利用率和较长的时延问题，提出面向 CLOS 网络架构基于分组的循环路由算法 DRB，该算法对同一源目的节点对之间的流量采用间隔循环选择核心层交换机的方法均衡流量到多条路径中，以避免大量拥塞的发生。文献[21]在 CLOS 网络架构的基础上引入了基于马尔可夫链的分布式自适应路由策略。该马尔可夫链模型建立在输出交换机上，通过不断调整网络路由策略，直到网络中的流量实现无阻塞路由。

考虑到数据中心网络流量的特征以及 ECMP 等传统路由方式的缺点，利用 SDN 技术集中控制和具有全局视角的优势，本文针对树型分层网络架构提出了软件定义数据中心网络混合路由机制 SHR（software-defined hybrid routing）。SHR 采取了一种折中的方法，对数据流进行分类，小流采用流量无视路由算法，大流则采用自适应路由算法，同时采用 SDN 技术进行集中控制。具体来讲，首先通过设定判定机制将数据流分为大流和小流。就传输性能需求而言，大流对吞吐量要求较高，小流则对时延要求较高。因此，SHR 对于大量时延敏感的小流以设定的概率采用流量无视路由算法完成路由。对于少量的大流，则由控制器利用自适应算法计算一条符合截止时间约束、队列长度最短的转发路径完成路由，从而实现数据中心网络流量的负载均衡。为了避免因 SDN 控制器和交换机频繁交互

产生的大量额外负载，本文在 OpenFlow 机制的基础上，将流的判定以及小流的路由功能下放至边缘层交换机，控制器只负责统计、计算流分类的阈值以及大流路径的选择，以此来减轻控制器的负担以及网络的额外负载。本文在 Fat-Tree 架构上建立流量模型并进行性能分析与实验对比，结果表明，与 ECMP 算法相比，SHR 能够提高网络吞吐量、降低数据流丢弃率和分组端到端时延、减轻网络的额外负载。

本文的贡献总结如下。

1) 设计流的动态判定机制，对大流和小流分别采用自适应路由和流量无视路由算法完成路由，满足其不同的传输性能需求以及实现网络负载均衡。

2) 将流的判定及小流的路由处理功能从控制器下放至交换机，而流的统计及网络状态信息的收集则以周期性打包统计的方式完成，减轻了控制器的计算负担以及网络的额外负载，同时减小了对时延敏感的流的处理时延。

2 软件定义数据中心网络混合路由设计

本节对软件定义数据中心网络混合路由机制 SHR 的设计进行介绍。图 1 是对 SHR 架构的简要描述。

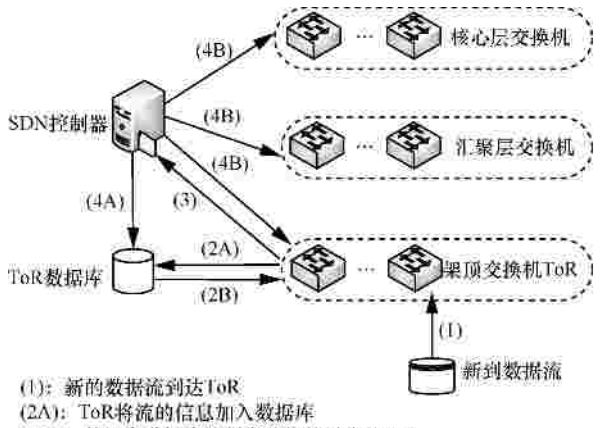

图 1 SHR 总体设计

本文面向传统的树型分层结构的数据中心网络，建立混合路由机制 SHR。SHR 基于 SDN 工作，SDN 将控制平面和数据平面分离，能进行流层面的控制，同时具有网络全局的视角，从而达到简化网络和数据流管理的目的。如图 1 所示，SHR 针对的树型分层架构分为核心层交换机、汇聚层交换机和架顶交换机 ToR 3 层。由 SDN 控制器周期性统计计算流分类阈值，架顶交换机 ToR 根据此阈值对数据流进行分类，并对不同的流按照不同的算法路由。对于大流的路由，由控制器进行计算，而小流的路径计算则直接由架顶交换机完成。

在该架构中，每隔周期时间 T，架顶交换机将流的基本信息打包发送给控制器，所有交换机将相关状态信息上报至控制器。控制器根据收集的流信息计算 T 时间内的流分类阈值并下发至架顶交换机。当有新流到达架顶交换机 ToR 时，ToR 将流的统计信息加入数据库，数据库根据流的判定阈值进行流的判定，并将判定结果告知 ToR。若判定为小流，ToR 按流量无视路由算法完成路由，若判定为大流，则将流的信息发送给控制器，控制器根据其掌握的全网运行信息，利用自适应路由算法计算该流的路径并下发至相关交换机。交换机将相应的信息写入流表的匹配项中以完成路由。

2.1 流的统计及判定

全局的流路由机制需要实时统计全网的相关信息，SHR 同样沿用了这种方法。研究发现数据中心中超过 80%的流的持续时间小于 10 s[15]。Hedera 将统计间隔设为 5 s，然而文献[22]指出，5 s 的统计间隔仅仅在 ECMP 的基础上将链路利用率提高了 1%~5%，并通过实验发现，将统计间隔设为 1 s 是比较适合的。因此 SHR 将统计间隔 T 设为 1 s。

SHR 只对边缘层交换机的流信息进行统计。在 nT 时刻，交换机将第 n 个统计周期 T 内的数据流统计信息打包后发送给控制器，控制器收集到各交换机的统计信息后按照流分类阈值的计算方法计算该周期内的阈值 Th_{nT}，并将其下发至各边缘层交换机。在收到第 n 个周期的阈值信息之前，交换机按照上一周期的阈值 $Th_{(n-1)T}$ 进行流的判定。收到之后则按阈值 Th_{nT} 进行判定，大于 Th_{nT} 的流判定为大流，反之则为小流。

基于统计中控制限的思想，控制器根据 T 时间内统计的流大小计算其均值 E 和标准差 d，以 $Th_{nT}=E_{nT}+2d_{nT}$ 为第 n 个周期统计的流分类阈值。控制器统计计算的第 n 个周期统计的流分类阈值 Th_{nT} 指导下一个周期内流分类的判定。对于第 1 个周期内数据流的判定阈值，设定初始阈值 Th=100 MB。

2.2 路由机制

在数据中心网络中，大小不同的流所对应的传

输性能需求有所不同，小流对时延要求高，而大流则对吞吐量要求高。因此 SHR 根据流的分类对其按不同的算法路由。如前所述，小流的容量较小、数量极多，对时延敏感。而流量无视路由算法的特点就是计算速度快，且能实现较好的网络负载均衡。因此本文对小流采用流量无视路由算法。而大流则所占流量较大，数量较少，对时延不是很敏感，但对吞吐量要求较高，因此本文通过带截止时间约束的基于队列长度的自适应路由算法计算一条当前最空闲的路径完成路由，在提高吞吐量的同时实现网络的负载均衡。

具体来讲，架顶交换机根据控制器下发的流分类阈值对流进行判定，分为小流和大流。对于小流，无需上报控制器，直接按照流量无视路由算法以可用链路的带宽计算概率随机选取路径。ECMP 能够统一地随机均匀选择输出端口，带来负载均衡。但其要求等价的多路径，因此在不规则的拓扑中负载均衡较差。如在一条 1 Gbit/s 的链路和一条 10 Gbit/s 的链路上，ECMP 均分流量其实是不合理的。基于此，在 SHR 的流量无视路由算法中，以链路的带宽计算概率分布选择路径。在上例中，以 $\frac{10}{11}$ 的概率在 10 Gbit/s 的链路转发，而以 $\frac{1}{11}$ 的概率选择 1 Gbit/s 的链路。SHR 利用 valiant 负载均衡的思想，分 2 个阶段完成路由，第一阶段以可用链路的带宽进行概率计算选取中间节点 x，然后再从 x 路由至目标节点 d。记 b_i 为第 i 条可选链路的带宽，则选择第 i 条可选链路的概率为 $p_i=\frac{b_i}{\sum_{i=1}^{l} b_i}$，$l$ 为可选路径总数。SHR 约束路径必须为最短路径，约束中间节点 x 必须在路由的最短路径上。SHR 针对的是树型分层架构，约束中间节点 x 必须为源地址和目标地址的共同祖先。在第一阶段，以概率 p_i 选择上行路径，直到遇见源地址和目标地址的共同祖先 x，然后开始第二阶段按固定路径完成路由。

对于大流，SHR 采取自适应路由算法。该自适应路由算法以路径上总的缓冲队列长度作为度量值，路径 p 的总队列长度 $Q(p)=\sum_{i=1}^{m} q_i$，q_i 为节点 i 的队列长度。文献[23]研究发现，在 1 s 的时间周期内数据中心网络流量并没有明显的变化，因此 SHR 中控制器通过周期性统计各交换机的缓冲队列长度为大流计算路径。每隔周期时间 T，所有交换机将其缓冲队列长度上报至控制器，控制器根据此信息计算可选路径 p 的总队列长度 $Q(p)$，然后比较其大小为大流做出路径选择。文献[16]研究指出，通常应用要求 200~300 ms 时间内完成，而当平均截止时间为 40 ms 时，依然有 7%的流错过其截止时间。为了尽可能避免大流错过其截止时间，本文先根据其截止时间的大小由小到大在边缘层交换机对流进行排队，交换机按优先级将大流信息上报至控制器。控制器首先比较时延是否符合截止时间约束条件，若符合，则根据掌握的全网状态计算一条队列长度最小的路径并下发至交换机完成路由。若不符合，则给交换机下发一条丢弃的指令。因此，路由问题的优化目标即为选择一条最优路径 p，则该最优路径问题为

1) min $Q(p)$；

2) s.t. $D(p)<T_{\text{deadline}}$。

对于该问题的求解，首先滤除可选路径集 P 中不符合截止时间约束的路径，得到新的路径集 P_c。然后计算一条路径集中队列长度最小的路径即为最优路径 p。

2.3　路由算法

下面对 SHR 的算法进行分析。

算法 1　SHR 算法

输入：G//数据中心网络拓扑

F_{nT}//第 n 个周期内流的统计信息集

$Th_{(n-1)T}$//上一周期的流分类阈值

Th=100 MB//流分类阈值的初始值

输出：{ <*e.method, e.path*>, $\forall e \in F$ }//流的处理方法和路由路径

1) if 新流 f 到达 then

2)　F_{nT}? F_{nT}+{ f }；

3)　if ($f.size \le Th_{(n-1)T}=E_{(n-1)T}+2d_{(n-1)T}$) then

4)　　< p > ?　RandomSelectPath(G, f);

　　//触发小流流量无视路由算法

5)　　$f.path?\ p$;

6)　　return

7)　　{< *e.method, e.path* >, $\forall e \in F$};

8)　else

9)　　按照截止时间由小到大排队;

10)　　将流的信息上报控制器；

11)　　< t, p > ?　PathCalculation(G, f);

　　//触发大流自适应路由算法

12) *f.method*? *t*; *f.path*? *p*;
13) return
14) {< *e.method*, *e.path* >, ∀ *e*∈*F*};
15) end
16) end

算法 1 是对 SHR 的整体描述。 T **表示统计的周期时间，** F_{nT} **表示第** n **个周期内到达边缘层交换机的流的集合，** $Th_{(n-1)T}$ **表示控制器计算出的上一个周期的流分类阈值，该阈值对本时间间隔内到达的流的分类具有指导作用。当一个新流到达一台边缘层交换机的时候，算法被触发。算法需要为每个目标流提供一条可用路径以及对该流的处理办法（丢弃或路由）。首先，当新流** f **到达后，交换机会将其统计信息添加至第** n **个周期内流的统计信息集** F_{nT} **中并存储在缓存当中，在** nT **时刻打包发送给控制器。与此同时，算法对该流的大小** *f.size* **进行判断，若** *f.size* **小于等于交换机当前的流判定阈值** $Th_{(n-1)T}$**，则判断为小流，触发子算法** RandomSelectPath(.)**来计算该流的路径及处理方法，交换机按照算法返回的计算结果对该流进行处理（算法第** 3)**至** 6 **行)。相反，若** *f.size* > $Th_{(n-1)T}$**，则以截止时间为优先级按由小到大的顺序排队，交换机按优先级将大流的信息上报至控制器。当控制器收到该信息后，触发子算法** PathCalculation(·)**来计算该流的路径和处理方法，控制器将算法结果下发给交换机，交换机依据此结果完成相应的操作。**

子算法 RandomSelectPath(·)**和** PathCalculation(·) **分别在算法** 2 **和算法** 3 **中进行了描述。算法** 2 **在边缘层交换机处，用来为小流选择路径。本文方案针对数据中心网络树型分层结构，对于目标流** f**，首先根据其源地址和目标地址的匹配关系寻找其共同祖先集** x **以及从源节点到** x **的最短路径集** P**。例如在** Fat-Tree **结构中，假设交换机端口数为** k**，则** 2 **台服务器之间共有** $\left(\frac{k}{2}\right)^2$ **条可选最短路径，在此路径集里以概率** r_i **选择路径** p_i**。从共同祖先** x **到目标节点，则计算其固定路径** p'**，更新路径** p **即为所选路径。概率** r_i **的计算如前文所述（算法第** 4)~6)**行)。**

算法 2 流量无视路由算法
输入：G//**数据中心网络拓扑**
F//**处理目标流**
r_i//**选择路径** p_i **的概率**
输出：{<*t*, *p*>}//**目标流的处理方法和路由路径**
1) *P*? GetCAPathSet(*f.source*, *f.destination*);
2) return {p_i, p_i∈*P*, 0<*i*≤*n*};
//**计算源地址和目的地址的共同祖先路径集**
3) if (*P*?*F*) then
4) for (*i*=1;*i*≤*n*;*i*++)
5) **以概率** r_i **选择路径；**
6) return <*p*>;
7) end for
8) *p'*? GetFixedPath (*f.source*, *f.destination*);
//**计算固定路径** *p'*
9) *p*? *p*+*p'*;
10) return <*Forwarding*, *p*>;
11) else
12) return <*Nopath*, - 1>;
13) end

算法 3 自适应路由算法
输入：G//**数据中心网络拓扑**
f//**处理目标流**
输出：{<*t*, *p*>}//**目标流的处理方法和路由路径**
1) *P*? GetAvailablePathSet(*G*, *f*);
2) return {p_i, p_i∈*P*, 0<*i*≤*n*};
//**计算可用路径集**
3) *D*(*P*)? GetAvailablePathDelaySet(*G*, *f*);
//**计算可用路径集的时延**
4) *Q*(*P*)? GetAvailablePathQueueSet(*G*, *f*);
//**计算可用路径集的队列长度**
5) if (*P*?*F*) then
6) for (*i*=1; *i*≤*n*; *i*++)
7) if (*D*(p_i)≤*f.deadline*) then
8) *P*? *P*- p_i
9) end for
10) if (*P*=*F*) then
11) return <*Drop*, - 1>;
12) else
13) *p*? GetMinQueuePath(*Q*(*P*));
//**计算最短队列路径**
14) return <*Forwarding*, *p*>;
15) end
17) else
18) return <*Nopath*, - 1>;
19) end

算法 3 **是控制器为大流根据网络状况选择路径的自适应路由算法。当收到交换机的流路由请求信**

息后，根据网络拓扑状况和流的信息获取其可选最短路径集 P 以及路径上的时延 $D(P)$、队列长度 $Q(P)$。算法首先滤除时延大于流的截止时间 $f.deadline$ 的路径（算法第 6)至 9)行），然后在剩余路径集 P 中选择队列长度最小的路径 p。

在本算法中，数据流在路径 p 上的总时延为 $D(p)=\sum_{i=1}^{m}\sum_{j=1}^{m}d_{ij}+\sum_{i=1}^{m}\left(d_i+\frac{L}{b_i}\right)$，$i$、$j$ 为数据流经过的节点，其中，d_{ij} 为节点 i 和节点 j 之间的传播时延，d_i 为节点 i 的排队时延加处理时延，$\frac{L}{b_i}$ 为节点 i 的传输时延，L 为流长度，b_i 为节点 i 的输出端口带宽。路径 p 的总队列长度 $Q(p)$如前文所述。

下面对 SHR 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，小流的总数为 N，大流的总数为 F，2 台服务器之间共有 $\left(\frac{k}{2}\right)^2$ 条可选最短路径，计算大流路径的时间复杂度为大流数 F 和任意 2 个服务器间可选最短路径数乘积的函数，即 $O(F\left(\frac{k}{2}\right)^2)$，计算小流路径的时间复杂度为 $O(N)$，则总时间复杂度为 $O(F\left(\frac{k}{2}\right)^2)+O(N)$。在控制器中，需要 k^3 个存储单元存储网络的链路状态结构，则需要 F 个存储单元存储大流的信息。因此其空间复杂度为 $O(k^3+F)$。

3　实验与结果分析

本节在典型的 Fat-Tree 架构上对 SHR 进行仿真实验分析。将对比当前数据中心网络广泛使用的 ECMP 路由算法，展示 SHR 在网络吞吐量、流丢弃率、分组端到端时延方面的性能优势。此外，将对比截止时间约束以及在 OpenFlow 的基础上将部分功能从控制器下放至交换机对网络吞吐量的影响。

3.1　仿真设置

网络拓扑：使用典型的 Fat-Tree 架构进行仿真分析。服务器到架顶交换机之间的链路带宽设定为 1 Gbit/s，各交换机之间的链路带宽设定为 10 Gbit/s，设定各交换机的缓存为 64 MB，端口数为 48，因此共有 27 648 台服务器。

流量模型：根据 Benson[24]等对数据中心内部流量特征的研究分析结论来模拟流量。假设 95%的流大小为 1~100 KB，5%的流大小在 100 KB~100 MB，且均服从均匀分布。流的源服务器和目标服务器随机选择，仿真时根据参考文献[21]的研究结论设架顶交换机处活动流的数量为 1 000~5 000 条/秒，流的到达速率服从泊松分布(到达的时间间隔服从负指数分布)，共仿真 10 000 个流。

排队模型：在对分组端到端时延进行分析的过程中，选取 M/M/1 排队模型。分组到达为泊松过程，即分组间相继到达系统的间隔时间服从参数为 l 的负指数分布，同时节点的处理时间服从负指数分布，均值为 $\frac{1}{u}$ s。参照当前数据中心网络使用较为广泛的交换机性能参数，在本文仿真中设交换机节点具有相同的分组处理速率 u=9.6 Mpacket/s，设定分组到达速率 l =0.88 Mpacket/s。

时延模型：数据中心网络分组端到端时延由传播时延 T_{prop}、传输时延 T_{tran} 以及分组在交换机中的排队时延和处理时延 4 部分组成。传播时延与链路的传输介质和距离有关，由于数据中心内交换机之间的连接使用光纤且距离较短，其传播时延非常小，对分组端到端时延影响很小，可以忽略不计。传输时延 T_{tran} 取决于交换节点的端口带宽和分组的长度，若分组长度为 L，端口带宽为 b，则 $T_{\text{tran}}=\frac{L}{b}$。分组在交换机中的时延（排队时延加处理时延）记为 T_{sys}。因此，网络中分组端到端时延为 $T_{\text{delay}}=\sum_{i=1}^{h}T_{\text{prop}i}+\sum_{j=1}^{s}(T_{\text{tran}j}+T_{\text{sys}j})$，其中，$h$ 和 s 分别表示分组路由所经过的总链路跳数和交换节点数，对于本文仿真使用的 Fat-Tree 架构，h=4，s=5。

3.2　性能评价

本文选取网络吞吐量、流丢弃率和分组端到端时延作为性能评价指标。网络吞吐量是指在单位时间内通过网络传输并成功接收的数据总量。流丢弃率指单位时间内数据流被丢弃的比率。分组端到端时延是指多个分组从发送端架顶交换机出发，经过中间节点和交换路径到达接收端架顶交换机之间的平均时延。

图 2 是在 Fat-Tree 仿真架构上分别采用 ECMP 和 SHR 路由的网络吞吐量仿真对比。仿真结果表明网络吞吐量随着数据流到达速率不断增加而逐步增加，当架顶交换机处的流到达速率为 5 000 条/秒时，SHR 的网络吞吐量比 ECMP 的网络吞吐量高 14.1%。这是由于 ECMP 在路径分配前并没有检测负载情况，只能提供静态的流量均衡。由于数据中

心网络中的流的大小和流的持续时间有很大的差异，网络利用率也会随时改变，另外静态散列分配方式很有可能将 2 个大流分配到同一条路径上，而相对较空闲的链路却没有分配到流，这容易导致拥塞进而造成数据流的丢弃、影响吞吐量。而 SHR 对于大流则按照可行路径上的负载情况选择最优路径，一定程度上避免了拥塞，提高了吞吐量。

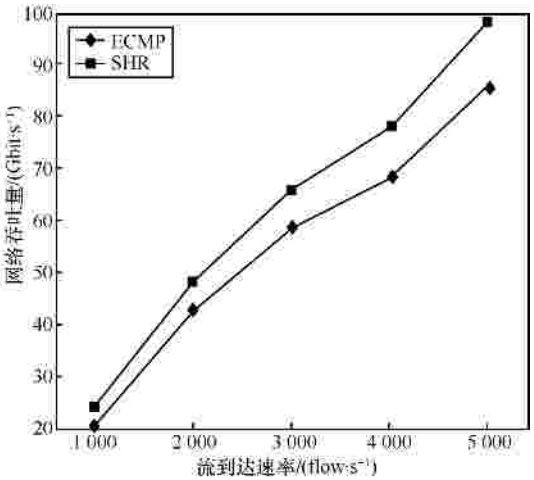

图 2 网络吞吐量仿真对比

图 3 描述的是仿真架构中的数据流丢弃率与数据流到达速率之间的关系。随着数据流到达速率不断增加，数据丢弃的概率也不断增加。当架顶交换机处的流到达速率为 5 000 条/秒时，SHR 路由的流丢弃率比 ECMP 路由的流丢弃率降低 18.3%。SHR 通过流分类将大流路由到拥塞程度最小的路径上，而 ECMP 采用的是随机的流量分发方式，算法不涉及当前网络利用率和流大小等动态信息，因此其丢弃的概率要高于 SHR。

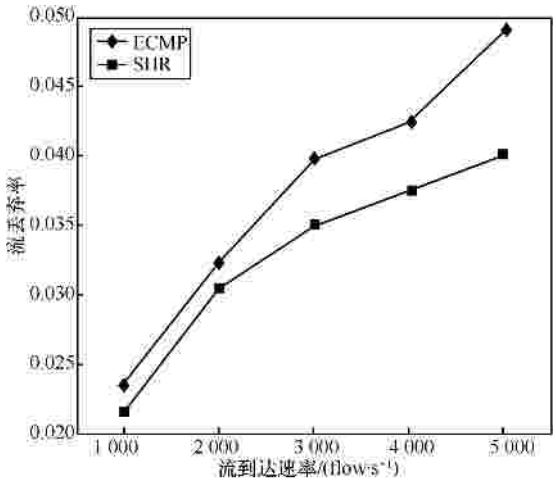

图 3 数据流丢弃率仿真对比

在对分组端到端时延分析仿真的过程中，考虑到实际网络中多条数据分组队列的相互影响，采用 Kleinrock 独立近似的处理方法[26]，将一条链路上传输的多个数据流合并起来，得到类似到达间隔时间与数据分组长度相互独立的情况。举例分析，数据分组到达率为l，当仿真拓扑中有 2 条路径可供选择时，对于流量无视路由算法，2 条路径均分流量。因此，每一条路径都是到达率为$\frac{l}{2}$的 M/M/1 队列。利用 Little 公式，可得分组在排队系统中的时延 $T_{sysr}=\frac{1}{u-\frac{l}{2}}=\frac{2}{2u-l}$，对于自适应路由算法，到达的数据分组进入排队较短的队列，这时系统相当于一个 M/M/2 排队系统，总的到达率为l，根据 M/M/n 排队系统的特性进行相应推导，数据分组在排队系统中的时延 $T_{sysm}=\frac{2}{(2u-l)(1+r)}$，其中，$r=\frac{l}{2u}$。可以看出，当有 2 条路径可供选择时，自适应路由算法能够使数据分组在排队系统中的时延减少到流量无视路由算法的$\frac{1}{1+r}$。

图 4 的仿真结果显示，随着网络负载的增加，2 种路由方式的分组端到端时延也随之增加，ECMP 的分组端到端时延要高于 SHR。并且随着负载的增加，其时延的增长速度同样高于 SHR。如前文所述，SHR 对绝大部分的数据流量采用自适应路由的方式，因此相比较 ECMP，其在时延方面具有优势。

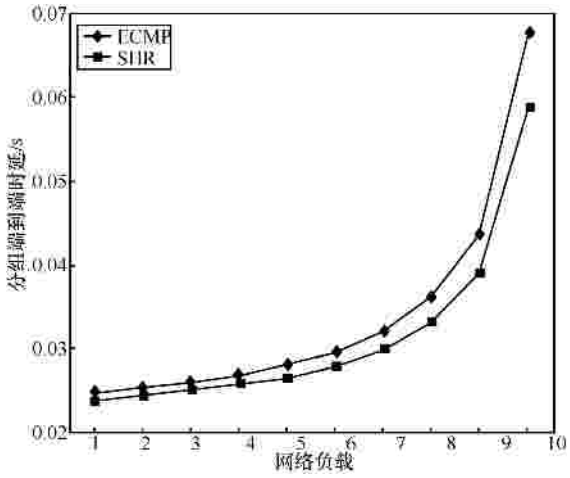

图 4 分组端到端时延对比

为了尽可能避免大流错过其截止时间，SHR 对

大流按其截止时间的大小进行了排队，并对其进行了截止时间约束。为了研究其对网络性能的影响，在本文所述的混合路由机制上进行了性能比较分析。图 5 的仿真结果表明，当架顶交换机处的流到达速率超过 4 000 条/秒时，有截止时间约束对网络吞吐量的提高是明显的。因为进行截止时间约束可以及时释放截止时间内无法完成的大流所占的带宽，进而减小网络产生拥塞的概率，提高网络吞吐量。

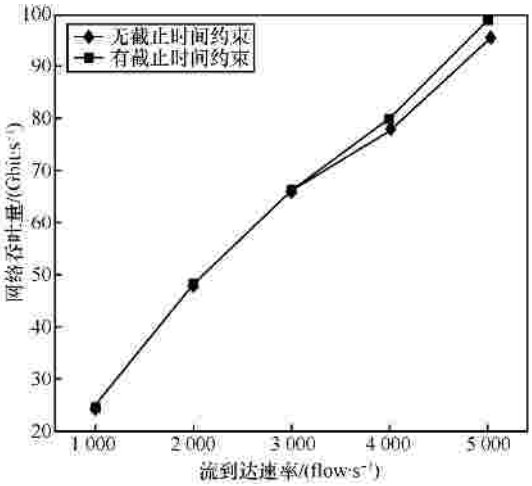

图 5　截止时间约束对网络吞吐量的影响

基于流的网络比传统网络对控制平面的依赖更高，因此带来更高的额外负载，这主要来自于交换机与控制器之间通信的带宽和时延，同时还有安装启用造成的负载。例如在有 N 台交换机的路径上安装双向的流，OpenFlow 要产生 $2N$ 个流表项安装分组，同时还有 4 个起始分组，这就是 $2N+4$ 个额外的分组。在数据中心网络中，80%以上的流为小流，因此面对数据中心大量的小流时，传统的 OpenFlow 机制会带来太多的额外控制流量。就网络负载而言，OpenFlow 单路径安装的额外负载大约为 $94+144N$ byte。如 3 台交换机的路径，则产生 526 byte 额外负载。基于此，SHR 在传统 OpenFlow 的基础上做了部分修改，对小流的处理全部安排在 OpenFlow 交换机完成，而且所有统计信息都是周期性的上报，这会减少网络的额外负载。

网络吞吐量仿真对比结果如图 6 所示，当流到达速率增大时，与传统的 OpenFlow 机制相比，本文做了部分修改后的 OpenFlow 机制对网络吞吐量的提高更为明显。当架顶交换机处流的到达速率为 5 000 条/秒时，网络吞吐量较传统 OpenFlow 机制能够增加 7.4%。这是由于 SHR 将大量小流处理的控制权交由交换机，减少了交换机与控制器之间的通信开销，释放较多额外负载占用的带宽，缓解了网络的拥塞，提高了网络吞吐量。

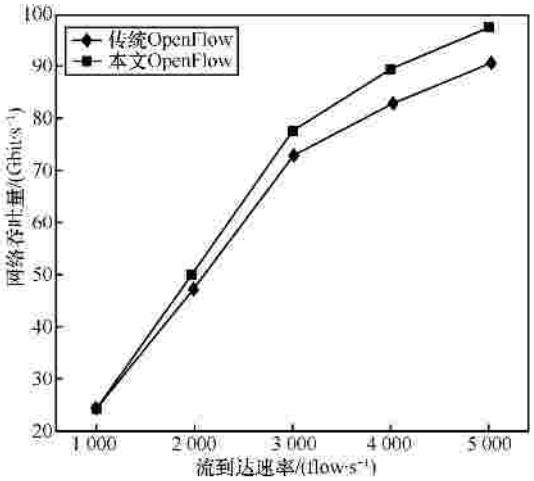

图 6　与传统 OpenFlow 技术对网络吞吐量的影响对比

4　结束语

本文提出了一种面向树型分层数据中心网络架构的路由机制——软件定义混合路由机制。该路由机制根据数据流的大小将其分类为大流和小流，并根据其不同的传输性能需求以不同的方式进行路由，以实现负载均衡，提高网络吞吐量和降低数据流丢弃率。本文对传统 OpenFlow 机制进行了改进，将流的分类以及小流的路由处理功能从控制器下放至交换机，减轻了控制器的计算负担以及网络的额外负载。对大流以截止时间为优先级排队和截止时间约束，减轻网络了的负担。本文通过建立合成流量模型对 SHR 的性能进行仿真评价，仿真结果表明，相比较传统的 ECMP 算法和传统的 OpenFlow 机制，SHR 能够提高数据中心网络的吞吐量，降低数据流丢弃率，减小分组端到端时延。

SHR 仅针对树型分层数据中心网络架构设计，并进行了实验分析，而在其他拓扑结构（如以服务器为中心的网络拓扑）下的应用情况还需要进一步研究分析。此外，分析 SHR 在链路利用率、容错能力等方面的性能将是下一阶段的工作。

参考文献：

[1] CHEN Y Y, JAIN S, ADHIKARI V K, et al. A first look at inter-data center traffic characteristics via Yahoo! datasets[C]//IEEE

INFOCOM'11. Shanghai, China, c2011: 1620-1628.
[2] AL-FARES M, LOUKISSAS A, VAHDAT A. A scalable, commodity data center network architecture[C]//SIGCOMM '08. Seattle, WA, USA, c2008: 63-74.
[3] ANDREW R, KIM W H, PRAVEEN Y. Mahout low-overhead DC traffic management using end-host-based elephant detection[C]//Proc of IEEE INFOCOM'11. Shanghai, China, c2011: 1629-1637.
[4] GUO C, LU G, LI D, et al. Bcube: a high performance, server-centric network architecture for modular data centers[C]//SIGCOMM'09. Barcelona, Spain, c2009: 63-74.
[5] GREENBERG A, HAMILTON J R, JAIN N, et al. VL2: a scalable and flexible data center network[C]//SIGCOMM'09. Barcelona, Spain, c2009: 51-62.
[6] GUO C, WU H, TAN K, et al. Dcell: a scalable and fault-tolerant network structure for data centers[C]//SIGCOMM'08. Seattle, WA, USA, c2008: 75-86.
[7] LI D, GUO C, WU H, et al. FiConn: using backup port for server interconnection in data centers[C]//SIGCOMM'09. Barcelona, Spain, c2009: 2276-2285.
[8] MYSORE R N, PAMBORIS A, FARRINGTON N, et al. PortLand: a scalable fault-tolerant layer 2 data center network fabric[C]// SIGCOMM '09. Barcelona, Spain. c2009.
[9] ABU-LIIBDEH H, COSTA P, ROWSTRON A, et al. Symbiotic routing in future data centers[C]//SIGCOMM'10. New Delhi, India, c2010: 51-62.
[10] WU H, LU G, LI D, et al. MDCube: a high performance network structure for modular data center interconnection[C]//ACM CoNext'09. Rome, Italy, c2009: 25-36.
[11] HOPPS C. Analysis of an equal-cost multi-path algorithm[S]. RFC 2992, IETF, 2000.
[12] BENSON T, ANAND A, AKELLA A, et al. Understanding data center traffic characteristics[C]//Proc of SIGCOMM'09. Barcelona, Spain, c2009: 92-99.
[13] Cisco Systems. Cisco data center infrastructure 2.5 design guide [EB/OL]. http://www.cisco.com/, 2013.
[14] LI D, XU M, ZHAO H, et al. Building mega data center from heterogeneous containers[C]//ICNP'11. Vancouver, BC Canada, c2011: 256-265.
[15] AL-FARES M, RADHAKRISHNAN S, RAGHAVAN B, et al. Hedera: dynamic flow scheduling for data center networks[C]//Usenix NSDI'10. California, USA, c2010: 19.
[16] WILSON C, BALLANI H, KARAGIANNIS T, et al. Better never than late: meeting deadlines in datacenter networks[C]//SIGCOMM'11. Toronto, Ontario, Canada, c2011: 50-61.
[17] HONG C Y, CAESAR M, GODFREY P B. Finishing flows quickly with preemptive scheduling[J]. ACM Computer Communication Review, 2012, 42(4):127-138.
[18] ZATS D, DAS T, MOHAN P, et al. DeTail: reducing the flow completion time tail in datacenter networks[J]. ACM Sigcomm Computer Communication Review, 2012, 42(4):139-150.
[19] SUN Y, CHEN M, LIU B, et al. FAR: a fault-avoidance routing method for data center networks with regular topology[C]//The 9th ACM/IEEE Symposium on Architectures for Networking and Communications Systems IEEE Press. San Jose, CA, USA, c2013: 181-189.
[20] CAO J, XIA R, YANG P, et al. Per-packet load-balanced, low-latency routing for clos-based data center networks[C]//The 9th ACM Conference on Emerging Networking experiments and Tech logies ACM. Santa Barbara, CA, USA, c2013:49-60.
[21] ZAHAVI E, KESLASSY I, KOLODNY A. Distributed adaptive routing convergence to non-blocking DCN routing assignments[J]. Selected Areas in Communications IEEE Journal, 2014, 32(1): 88-101.
[22] ANDREW R, JEFFREY C, et al. DevoFlow: scaling flow management for high-performance networks[C]//SIGCOMM'11. Toronto, Ontario, Canada, c2011: 254-265.
[23] BENSON T, ANAND A, AKELLA A, et al. MicroTE: fine grained traffic engineering for data centers[C]//The 7th Conference on Emerging Networking Experiments and Technologies ACM. Tokyo, Japan, c2011:1-12.
[24] BENSON T, AKELLA A, MALTZ D A. Network traffic characteristics of data centers in the wild[C]//IMC'10. Melbourne, Australia, c2010: 267-280.
[25] KANDULA S, SENGUPTA S, GREENBERG A, et al. The nature of data center traffic: measurements and analysis[C]//IMC'09. Chicago, Illinois, USA, c2009: 202-208.
[26] **樊平毅, 冯重熙. 现代通信理论基础（中册）**[M]. **北京: 清华大学出版社**, 2007:100-132.
FAN P Y, FENG C X. Fundamentals of modern communications[M]. Beijing: Tsinghua University Press, 2007:100-132.

作者简介：

蔡岳平（1980-），男，江苏丹阳人，重庆大学副教授、硕士生导师，主要研究方向为数据中心网络、光通信网络、未来互联网等。

王昌平（1984-），男，甘肃定西人，重庆大学硕士生，主要研究方向为数据中心网络、未来互联网等。