

文章编号:1007-5321(2020)02-0046-06

DOI:10.13190/j.jbupt.2019-105

基于时延和能耗的 SD-DCN 的路由优化算法

姚 赞, 王 颖, 邱雪松, 文禹棋

(北京邮电大学 网络与交换技术国家重点实验室, 北京 100876)

摘要:为了更好地实现数据中心网络的节能,基于交换机链路速率级的能耗特点,基于软件定义网络技术,提出一种 Floyd-Warshall 动态规划和局部重路由的节能服务质量路由优化算法. 控制器在保障流的时延性能前提下,采用流在空间和时间上均衡传输的策略,依次为每个流计算传输路径和传输速率;在选路失败的情况下,尽量用较少的开销提高网络的接受率. 仿真结果表明,该算法有效地降低了能耗,同时提高了网络流的接受率.

关键词:软件定义数据中心网络;节能路由;截止时间感知

中图分类号:TP393

文献标志码:A

Deadline-Aware and Energy Efficient Routing Optimization Algorithm in SD-DCN

YAO Zan, WANG Ying, QIU Xue-song, WEN Yu-qi

(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Based on link speed-scaling energy consumption strategy, Floyd-Warshall dynamic planning & rerouting parts of flows strategy based routing optimization algorithm was proposed. Considering the condition of flow deadline-aware and balanced transmission of flows strategy in space and time, controller in software defined data center network sorts the online incoming flows and chooses the route and calculates the transmission rate for every flow. In case of routing failure, the algorithm improves the acceptance rate of network traffic with less overhead. Simulations show that the algorithm effectively reduces energy consumption and improves the acceptance rate of network traffic.

Key words: software defined data center network; energy efficient routing; deadline-aware

数据中心网络设计中大多数时间内的网络设备利用率很低,网络能耗效率极低. 构建节能网络对降低网络运营成本、节能减排等均具有着重要意义^[1-2]. 以搜索、社交网络为代表,影响数据交换性能的关键因素是流的截止期限^[2-3]. 因此,在数据中心网络中,在保障业务流的时延性能的同时,如何为到来的流选取服务质量(QoS, quality of service)路由,使得网络的能耗和开销最少,是一个值得研究的问题.

软件定义网络(SDN, software defined network)技术可以对网络流量进行精细化管理,因此其可以更好地在保障业务性能的同时,实现网络节能目标. 近几年有很多相关研究,从网络节能的角度,可以分为设备级、链路级和链路速率级节能3类. 文献[4-9]主要从设备级和链路级进行节能. Ba等^[4]提出了一种数据平面的多个拓扑在线切换的机制,在满足流量动态需求的同时,将部分设备和端口休眠从而实现节能; Fernandez-Fernandez等^[5]、Wu等^[6]、

收稿日期:2019-05-30

基金项目:国家重点研发计划项目(2018YFE0205502)

作者简介:姚 赞(1987—),女,博士生.

通信作者:王 颖(1976—),女,副教授, E-mail: wangy@bupt.edu.cn.

Chen 等^[7]和 Heller 等^[8]均提出了最小化活跃链路数目的启发式节能路由算法,在约束条件下,按一定策略对业务流进行排序,按次序对流进行路由,路径的集合即为节能拓扑。Xu 等^[9]采用带宽独占用的策略,提出一种基于带宽感知的高能效路由算法。以上方法均会选择交换机以及其端口的开关,实时性较差。针对数据中心的应用,更需要考虑端口速率级的节能,其可分为独占式和共享式两种带宽占用方式。Wang 等^[10]针对时延保障的流,基于松弛和随机舍入技术,提出了一种可抢占的独占式路由和调度方案。基于链路速率级的能耗模型特征,笔者将采用流共享带宽的策略,且流速率可变,从而在空间和时间上进行均衡分配,进一步提高链路速率级的节能。

笔者在数据中心网络拓扑、交换机以及其相关端口开关与否都已确定的场景下,对基于时延和能耗的数据中心网络路由优化问题进行了建模,提出一种改进的 Floyd-Warshall 动态规划算法,采用非抢占、流速率可变的带宽共享模式,保证时延的同时,从时间和空间两个维度实现节能;当出现寻路失败的情况时,设计了局部流重路由策略来实现路由优化,以较少的网络开销,提高网络接受率。

1 基于时延和能耗的路由问题模型

1.1 链路速率级节能

数据中心网络可以表示为一个无向图 $G = (V, E)$,其中 V 表示所有交换机的集合, E 表示所有链路 e 的集合。每条链路的电力消耗函数 $f(x_e)$ 是链路传输速率 x_e 的函数^[1],可通过式(1)进行计算。

$$f(x_e) = \sigma + \mu x_e^\alpha, 0 \leq x_e \leq \beta C \quad (1)$$

其中: σ 、 μ 、 α 是和链路类型相关的常量; σ 是链路空闲状态时的能耗; C 是链路的最大传输速率; β 是链路冗余度参数, $0 < \beta < 1$ 。 $f(\cdot)$ 是可加的, $\alpha > 1$ 。通过观察单条链路能耗模型发现,当 $\alpha > 1$ 时, $(a + b)^\alpha > a^\alpha + b^\alpha$ 。不难发现在网络拓扑确定的情况下,当流尽量在时间和空间上均衡传输时,网络能耗最小。

接下来给出一个简单的例子,可以直观上体现流在时间上的均衡传输对节能效果的影响。假设 u, v 为一对相邻节点,且由链路 e 相连。对于每个流 $j_i = \{p_i, q_i, r_i, d_i, b_i\}$,其中 p_i, q_i 分别表示其源和目的节点交换机, r_i, d_i 分别表示其到达和最晚截止时间, b_i 表示流的大小。假设在 e 上有两条流 $j_1 = \{u,$

$v, 0, 7, 35\}$ 和 $j_2 = \{u, v, 0, 12, 42\}$,分别使用带宽共享式和带宽独占式两种方案进行传输,如图 1 和图 2 所示。将能耗函数中 α 设置为 2, μ 设置为 1,由式(1)进行计算可得能耗增量分别为 539 和 616,对比可知图 1 带宽共享式传输方案的能耗增量较小。

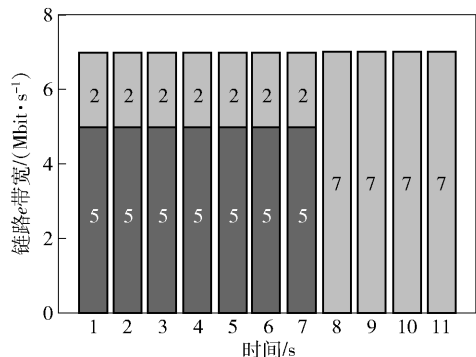


图1 带宽共享式传输

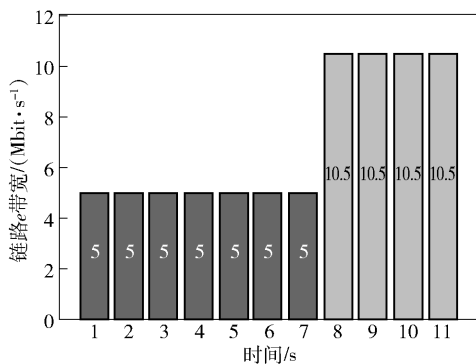


图2 带宽独占式传输

仅需要获取路径上所有链路每个时刻的带宽占用的最高值,流在空间上的均衡传输问题即可转化为单链路的时间均衡传输问题。因此以上结论可以由单条链路扩展到多条链路连接而成的路径,从而证明空间均衡可以有效降低能耗增量。

1.2 问题描述

假设一个较长时间周期内,根据数据中心网络流量的统计规律,开启特定的交换机和相关端口,当前时刻到达的流量矩阵为 J 。对于每个流 $j_i \in J$,主要研究在保证其时延传输性能的条件下,为了达到能耗增量最小的目标,SDN 控制器如何根据掌握的全局拓扑和负载情况,在线实时地为每个新到来的流 $j_i \in J$ 计算传输路径 p_i 、传输区间 $[r'_i, d'_i]$ 以及传输速率 $s_i(t)$ 。更进一步,当寻路失败时,研究如何选择部分之前流(后文统称为局部流)重新计算路由,以留出空间来传输新流,实现以较少的网络开销来提高网络的接受率。

1.3 模型建立

以能耗增量 $\Delta\phi$ 最小为主要优化目标. 首先假设网络拓扑已经确定, 即交换机以及链路的开通与否已经确定, 故网络节点能耗模型简化为速率粒度的能耗模型. 网络的能耗增量为主要优化目标, 由网络中所有链路在新流分配前后的能耗差值得到, 如式(2)所示.

优化目标:

$$\min \Delta\phi = \sum_{e \in E} f'(x_e) - \sum_{e \in E} f(x_e) \quad (2)$$

约束条件:

$$\int_{r_i}^{d_i} s_i(t) dt = b_i \quad (3)$$

$$0 \leq x_e \leq \beta C \quad (4)$$

$$\sum_{v \in N(u)} (f_i^{uv} - f_i^{vu}) = \begin{cases} b_i, & \text{若 } u = p_i \\ -b_i, & \text{若 } u = q_i \\ 0, & \text{其他} \end{cases} \quad (5)$$

式(3)表示性能约束条件, 每个流必须在其最晚截止时间前完成; 式(4)表示资源能力约束条件, 每条链路的带宽占用不能超过链路容量的 β 倍; 式(5)表示流量守恒限制, 从源交换机流出的流量等于流入目的交换机的流量, 所有中间交换机流入的流量等于其流出流量^[1], 其中 $N(u)$ 表示交换机的邻居交换机集合, f_i^{uv} 表示流 j_i 部署在链路 uv 上的流量大小, 在不分流的情况下, 这个数值为 w_i 或者 0.

2 基于时延和能耗的路由优化算法

笔者提出一种基于链路速率级节能的 QoS 路由优化方案, 主要分为 2 部分: 首先, 对到来的流按照截止时间早晚进行排序, 采用 Floyd-Warshall 动态规划算法依次为每个流寻求最节能的 QoS 路由, 详见 2.1 小节; 其次, 为了提高网络流的接受率, 同时降低通信开销, 在寻路失败情况下, 选取之前正在传输的部分流进行局部调整, 从而为新流找到合适路径, 实现路由优化, 详见 2.2 节. 在 2.3 节详细给出了基于时延和能耗的软件定义数据中心网络 (SD-DCN, software defined data center network) 路由优化的整个过程.

2.1 Floyd-Warshall 算法

Floyd-Warshall 算法是解决任意两点间的最短路径的一种动态规划算法. 笔者选择 Floyd-Warshall 算法为新到来的流寻找节能 QoS 路径. 首先定义邻接矩阵 W , 每个元素 $w_{ij} \in W$ 表示链路的权

重, 其代表增加新流后的链路能耗增量, 如式(6)所示.

$$w_{uv} = \begin{cases} 0, & \text{若 } u = v \\ f(x'_e) - f(x_e), & \text{若 } u \neq v \text{ 且 } (u, v) \in E \\ \infty, & \text{若 } u \neq v \text{ 且 } (u, v) \notin E \end{cases} \quad (6)$$

对于任意节点 $u, v \in V$, Floyd-Warshall 算法考虑从节点 u 到节点 v 的所有中间节点均取自集合 $\{1, 2, \dots, k\}$ 的路径 (该集合是 V 的一个子集). 设 p 为最短路径, 对于任意的 $k > 0$, 通过分析可以看到, 中间顶点不超过 k 的节点 u 到 v 的最短路径有 2 种可能:

1) 该路径不含中间顶点 k , 那么路径 p 上的所有中间节点都属于集合 $\{1, 2, \dots, k-1\}$;

2) 该路径含中间顶点 k , 则可将路径分解为 2 条路径, 分别是节点 u 到节点 k 和从节点 k 到节点 v 的路径.

因此, 设 $d_{uv}^{(k)}$ 表示从节点 u 到节点 v 的所有中间节点全部取自集合 $\{1, 2, \dots, k\}$ 的一条短路径的权重. 当 $k=0$ 时, 表示节点 u 到节点 v 的路径没有任何中间节点, 权重等于两点之间的链路权重; 当 $k \geq 1$ 时, 权重选择经过 k 和不经过 k 的权重较小的那个. 根据上面的讨论, 递归定义如式(7)所示.

$$d_{uv}^{(k)} = \begin{cases} w_{ij}, & \text{若 } k=0 \\ \min(d_{uv}^{(k-1)}, d_{uk}^{(k-1)} + d_{kv}^{(k-1)}), & \text{若 } k \geq 1 \end{cases} \quad (7)$$

根据以上递归公式, 采用自底向上的算法以递增次序来计算 $d_{uv}^{(k)}$ 的值, 从而得到最短路径, 伪代码详见算法 1. 其中采用带宽共享、速率可变的均衡策略来计算流 j_i 的传输速率 $s_i(t)$, 伪代码详见算法 2 速率计算模块.

Floyd-Warshall 算法首先获取网络的节点数目, 并对链路的网络能耗增量初始化, 如算法中的第 1~2 行所示. 第 3~20 行以自底向上的方式递归寻路, 在每次寻路的过程中, 第 6~8 行首先将两段路 $p_{uk}^{(k-1)}, p_{kv}^{(k-1)}$ 合起来, 并调用算法 2 速率计算模块, 在满足时延约束的条件下, 重新计算两段路上的流的最节能的传输速率; 然后判断此路径是否有能力传输该流, 如果是, 则更新对应路径的带宽占用情况以及能耗增量, 反之则继续寻找新的路径, 如第 9~16 行所示. 最后第 21~24 行在算法执行完后, 更新所有最短路径的新的负载, 并返回 j_i 的传输速率和最短路径.

算法1 Floyd-Warshall 算法

输入: $G = (V, E)$, j_i , 网络的邻接矩阵 W .

输出: j_i 的传输速率 $s_i(t)$ 和传输路径 p_i .

```

1   $n \leftarrow \text{row}[W]$ ; // 获取网络节点数目
2   $d_{uv}^0 \leftarrow 0$ ; // 网络能耗增量初值, 均设为 0
3  For  $k \leftarrow 1$  to  $n$ 
4    Do for  $u \leftarrow 1$  to  $n$ 
5      Do for  $v \leftarrow 1$  to  $n$ 
6        For  $e \in p_{uk}^{(k-1)} \cup p_{kv}^{(k-1)}$  do
7          调动速率计算模块; // 计算流可能传
            输的速率
8        End for
9        If  $s_i(t) \neq 0$  then // 检测是否分配成功
10         For  $e \in p_{uk}^{(k-1)} \cup p_{kv}^{(k-1)}$  do
11            $x'_e \leftarrow x_e + s_i(t)$ ; // 更新链路速率
12         End for
13          $d_{uk}^{k-1} \leftarrow \sum_{e \in p_{uk}^{(k-1)}} [f(x'_e) - f(x_e)]$ ;
14          $d_{kv}^{k-1} \leftarrow \sum_{e \in p_{kv}^{(k-1)}} [f(x'_e) - f(x_e)]$ ; // 更新
            路径上链路的权重, 并求和
15          $d_{uv}^k \leftarrow \min(d_{uv}^{(k-1)}, d_{uk}^{(k-1)} + d_{kv}^{(k-1)})$ ; //
            选出能耗增量值较小的路径
16          $p_{uv}^k \leftarrow \min(p_{uv}^{(k-1)}, p_{uk}^{(k-1)} \cup p_{kv}^{(k-1)})$ ; // 更
            新当前最短路径
17       End if
18     End for
19   End for
20 End for // 以自底向上的方式递归寻路
21 For  $e \in p_{p_i q_i}^n$  do
22    $f(x_e) \leftarrow f(x'_e)$ ; // 选完路之后, 仅更新最
            短路径上的网络负载
23 End for
24 Return  $s_i(t)$ ,  $p_i \leftarrow p_{p_i q_i}^n$ ,  $\Delta\phi_i \leftarrow d_{p_i q_i}^n$ ; // 返回新
            流  $j_i$  的传输速率、传输路径以及网络能耗增量

```

在算法 2 速率计算模块, 首先第 1 行选取新流的可传输时间段作为调度区间, 在计算之前, 第 2 行先获取 $p_{uk}^{(k-1)}$ 和 $p_{kv}^{(k-1)}$ 路径上同一时间的不同链路带宽占有的最高值 $A(t)$, 将其转化为单链路的时间均衡问题. 接下来的 3 行定义一个参数平均带宽 $M = (\text{可调度区间占用的总带宽} + \text{新流的带宽}) / \text{可调度区间}$. 不难看出, 当 $A(t) > M$ 时, 表示此时刻不能传输新流. 接着第 4 ~ 13 行通过反复计算, 剔除高于平均带宽的区间, 直到找出可传输的区间.

最后第 14 ~ 17 行通过平均带宽减去已占用带宽即可得到 j_i 的传输速率, 并输出.

算法2 速率计算

输入: $G = (V, E)$, j_i , 网络的邻接矩阵 W , 最短路径权重矩阵 D 以及路径 P .

输出: j_i 的传输速率 $s_i(t)$.

```

1   $T = [r_i, d_i]$ ; // 可调度的时间区间
2   $A(t) \leftarrow \max(x_e(t))$ ,  $t \in T$ ; // 得到  $p_{uk}^{(k-1)}$  和  $p_{kv}^{(k-1)}$ 
            路径上同一时间不同链路带宽占有的最高值
3   $M = (\int_{r_i}^{d_i} A(t) dt + b_i) / T$ ; // 计算平均带宽
4  Repeat // 剔除高于平均带宽的区间, 直到找出可
            调度的区间
5    For  $t \in T$ 
6      If  $A(t) \geq M$  then
7         $T' \leftarrow t$ ;
8         $T \leftarrow T - T'$ ;
9      End if
10   End for
11    $A(t) \leftarrow \max(x_e(t))$ ,  $t \in T$ ;
12    $M = (\int_{r_i}^{d_i} A(t) dt + b_i) / T$ ;
13 Until  $A(t) < M < BC$ 
14 For  $t = r_i$  to  $d_i$ 
15    $s_i(t) = M - A(t)$ ;
16 End for // 计算  $j_i$  的速率
17 输出  $j_i$  的传输速率  $s_i(t)$ 

```

2.2 局部重路由策略

经过观察发现, 有时整个网络容量虽然够大, 由于按照最节能的策略为流选取路径, 所以理论上改变之前流的传输路径, 能够给后来的流在空间上和时间上腾出余地. 因此为了提高流的接受率, 同时考虑与节能效果、重新调度引起的开销之间的均衡, 可以对局部流进行重新调度. 开销主要考虑由局部流重路由带来的控制通信开销, 可简化为重路由的流的数目来表示. 网络中流量非常大的情况下, 开销会非常大. 因此为了降低重新调度的开销, 提出局部流重路由策略. 和新流有着相同源和目的地的流可作为首选的重路由对象, 因此主要选取有相同源点和目的节点的流作为局部流, 并且继续使用 2.1 节所提的算法为局部流和新到达的流计算节能 QoS 路由.

2.3 算法步骤

综上所述, 采用 Floyd-Warshall 算法和局部重路

由策略来求解 SD-DCN 数据流的节能 QoS 路由优化问题. 具体步骤如下.

步骤1 排队. 控制器首先对当前到达的集合 J 里的流, 根据截止时间从早到晚进行排序.

步骤2 对集合中排序最小的流寻路. 1) 计算单个链路权重. 控制器获取当前网络拓扑和负载情况, 并计算每条链路是否具备传输新流的能力, 并由式(7)计算能耗增量. 2) 使用 2.1 节中的 Floyd-Warshall 算法寻找节能 QoS 路径. 3) 如果成功, 则输出新流的最短路径、传输速率, 跳到步骤4; 反之, 则跳到步骤3, 调用局部重路由策略.

步骤3 局部重路由.

1) 对和此新流具有相同起始交换机和目的交换机的所有正在传输的流组成新的集合 J' ;

2) 对集合 J' 里的流, 参考步骤1, 首先根据截止时间早晚排序;

3) 使用 2.1 节中的 Floyd-Warshall 算法依次重新进行选路;

4) 如果集合 J' 里的流均成功找到路径, 则通知控制器进行路由优化; 反之, 则放弃新流的请求.

步骤4 返回步骤2 继续为后续流选路, 直到集合 J 为空结束.

3 仿真结果

在 Windows 7 系统下, 使用 Matlab 仿真软件进行编程实现所提算法, 硬件平台配置为 1 块 2.4 GHz CPU, 64 GB 内存. 选取 Fat-tree 数据中心网络拓扑, 包括 20 个 4 端口的交换机、16 个主机和 48 条链路. 能耗函数中 α 设置为 2, 网络链路带宽 C 设置为 180 Mbit/s, β 设置为 0.9. 分别使用 QoS 节能算法、局部重路由算法、独占式算法和离线算法 (共享式) 这 4 种算法, 对比流经过网络的能耗情况以及流的接受率.

为了更好地显示算法的效果, 同时鉴于硬件条件限制, 对 8 ~ 64 条工作流进行模拟, 且仿真结果只给出交换机端口速率相关的部分功率和能耗值. 首先选取 8 条工作流的开始时间和截止时间在 $[0, 16]$ s 内服从随机分布, 流的大小在 $[40, 200]$ MB 区间服从一致分布. 采用不同的 4 种路由算法来计算网络的能耗情况, 如图 3 ~ 图 5 所示. 图 3 表示网络的实时功率, 可以明显看到, 和其他 3 种算法相比, 因为使用独占式算法的流在时间上传输不均衡, 且有丢弃的流, 所以端口功率在时间上起伏较大. 其他 3 种均为共享式算法, 能耗是随着新流的到来而

变化的. 其中节能 QoS 算法放弃了一条流, 所以和其他两种相比, 中间能耗较低. 局部重路由和离线算法对所有流进行传输, 传输顺序有些许差异, 所以能耗值也有些许差异.

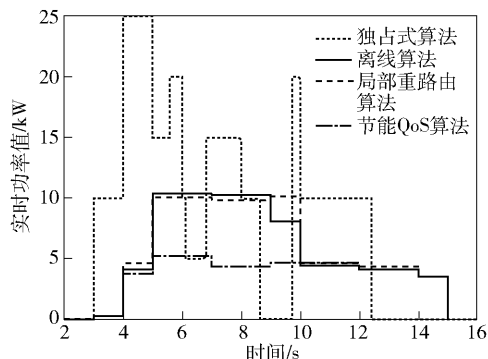


图3 实时能耗

图4的横坐标表示每条工作流的序号, 纵坐标代表新流到来所带来的能耗增量. 独占式算法所带来的能耗增量要高于节能 QoS 算法和局部重路由算法所引起的能耗增量. 通过对比图4中局部重路由与节能 QoS 算法的曲线可以发现, 在横坐标等于3时, 后者能耗值不变, 前者能耗值增高, 是因为第4条流到来时, 使用了局部重路由策略. 独占式算法的能耗值比其他3种算法的能耗值皆高. 离线算法因为按照工作流的截止时间对流进行了重新排序, 所以能耗值时而高时而低.

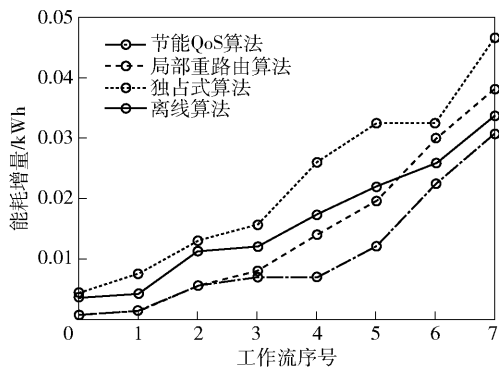


图4 能耗增量

图5给出了4种算法在8条流的整个传输区间的总体能耗, 可以看出独占算法 > 局部重路由算法 > 离线算法 > 节能 QoS 算法. 局部重路由算法比节能 QoS 算法提高了接受率, 所以能耗值较大; 同时, 因为其重调度的范围较小, 相比离线算法, 其能耗值也较大.

对不同规模的4组工作流 (分别是8条、16条、32条、64条) 分别使用4种算法进行仿真, 以验证网

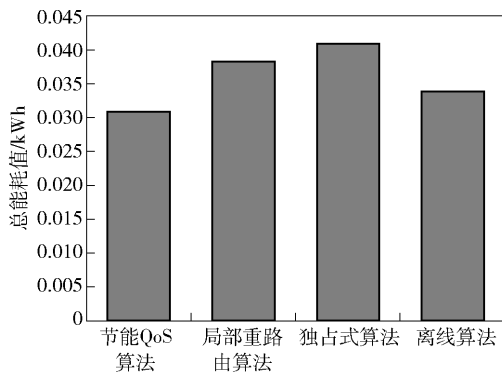


图 5 能耗总量

络对工作流的接受情况。如图 6 所示,独占式算法的接受率是最低的,其次是节能 QoS 算法。随着工作流规模的上升,网络拓扑中的流量强度逐渐增大,链路负载增加,链路容量接近其最大值。未进行重路由的节能 QoS 算法的接受率逐渐下降,而进行重路由的局部重路由算法以及离线算法的工作流接受率保持在一个大于 95% 的较高值。

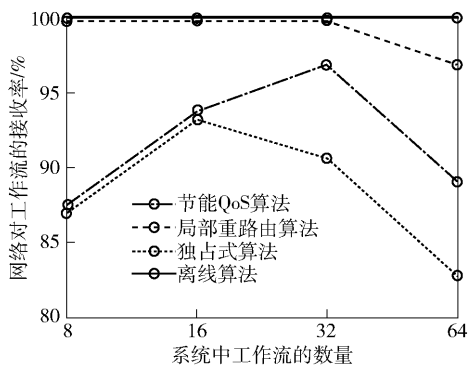


图 6 流的接受率

4 结束语

针对基于时延和能耗的数据中心网络路由优化问题,在数据中心网络拓扑、交换机以及相关端口开关与否都已确定的场景下,基于空间和时间上均衡可以降低能耗的原理,提出一种 Floyd-Warshall 动态规划方法和局部重路由策略的 QoS 路由优化方法,在线依次为每个流选择最优传输方案。本方法以较少的开销,有效地保证了流的时延性能,降低了网络能耗,同时提高了网络接受率。

参考文献:

- [1] Andrews M, Anta A, Zhang L, et al. Routing for power minimization in the speed scaling model[J]. IEEE/ACM Trans Netw, 2012, 20(1): 285-294.
- [2] Fan K, Wang Y, Ba J, et al. An approach for energy efficient deadline-constrained flow scheduling and routing [C]//2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). Arlington: IEEE Press, 2019: 469-475.
- [3] Yao Z, Wang Y, Ba J, et al. Deadline-aware and energy-efficient dynamic flow scheduling in data center network[C]//International Conference on Network and Service Management. Roma: IEEE Press, 2018: 1-4.
- [4] Ba J, Wang Y, Zhong, S X, et al. An SDN energy saving method based on topology switch and rerouting[C]//2018 IEEE/IFIP Network Operations and Management Symposium. Taipei: IEEE Press, 2018: 1-5.
- [5] Fernandez-Fernandez A, Cervello-Pastor C, Ochoa-Aday L. Achieving energy efficiency: an energy-aware approach in SDN[C]//2016 IEEE Global Communications Conference (GLOBECOM). Washington, DC: IEEE Press, 2016: 1-7.
- [6] Wu Z, Ji X, Wang Y, et al. An energy-aware routing for optimizing control and data traffic in SDN[C]//2018 26th International Conference on Systems Engineering (ICSEng). Sydney: IEEE Press, 2018: 1-4.
- [7] Chen Y, Chin T, Huang C, et al. Time efficient energy-aware routing in software defined networks [C]//2018 IEEE 7th International Conference on Cloud Networking (CloudNet). Tokyo: IEEE Press, 2018: 1-7.
- [8] Heller B, Seetharaman S, Mahadevan P, et al. Elastic-Tree: saving energy in data center networks[C]//Nsd. San Jose: [s. n.], 2010: 249-264.
- [9] Xu G, Dai B, Huang B, et al. Bandwidth-aware energy efficient flow scheduling with SDN in data center networks [J]. Future Generation Computer Systems, 2017, 68: 163-174.
- [10] Wang L, Zhang F, Zheng K, et al. Energy-efficient flow scheduling and routing with hard deadlines in data center networks[C]//2014 IEEE 34th International Conference on Distributed Computing Systems (ICDCS). Madrid: IEEE Press, 2014: 248-257.