集成 SDN 框架的启发式数据流调度算法研究

黄 润[1] 肖志良[1,2]

[1](佛山职业技术学院电子信息学院 广东 佛山 528137)

[2](武汉大学信息管理学院 湖北 武汉 430072)

摘 要 为了解决光数据中心的流调度问题和最大化云服务供应商的长期收入,提出最小拥塞和服务时间优先 MC-STP(Minimum Congestion and Service Time Priority)的调度算法,以及基于拥塞的循环调度 CBL(Congestion Based Loop)算法,并将其集成到软件定义网络(SDN)框架,以执行业务流调度和光路重构。其中:MC-STP 向服务时间较短的业务流给予较高的优先级,使其先于其他流被容纳;CBL 是为了弥补 MC-STP 的业务流饥饿问题,在计算出业务流的拥塞因子后,通过业务流的拥塞因子选择要调度的流,提供流之间的公平性。仿真结果表明,与端到端的调度算法、离散粒子群调度算法相比,该算法可明显降低拒绝率,提高波长利用率,有效提高云服务供应商的平均收入。

关键词 数据中心 调度算法 循环调度 最小拥塞 软件定义网络

中图分类号 TP393 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2019.04.024

HEURISTIC DATA FLOW SCHEDULING ALGORITHM INTEGRATING SDN FRAMEWORK

Huang Run[1] Xiao Zhiliang[1,2]

[1](*College of Electronic Information, Foshan Polytechnic, Foshan 528137, Guangdong, China*)

[2](*College of Information Management, Wuhan University, Wuhan 430072, Hubei, China*)

Abstract In order to solve the flow scheduling problem of the optical data center and maximize the long-term revenue of cloud service providers, we proposed a minimum congestion and service time priority(MC-STP) scheduling algorithm and a congestion based loop scheduling(CBL) algorithm, which were integrated into a software defined network(SDN) framework to perform traffic scheduling and optical path reconstruction. Among them, MC-STP gave higher priority to service flow with shorter service time, so that it could be accommodated before other flows. CBL was to make up for the starvation problem of MC-STP traffic. After calculating the congestion factor of traffic flow, we chose the flow to be scheduled through the congestion factor of traffic flow to provide fairness between flows. The simulation results verified the efficiency of the proposed algorithm. Compared with the end-to-end scheduling algorithm and discrete particle swarm optimization(PSO), this algorithm can significantly reduce the rejection rate, improve the wavelength utilization, and effectively improve the average revenue of cloud service providers.

Keywords Data center Scheduling algorithm Loop scheduling Minimum congestion SDN

0 引 言

近些年,云数据中心的规模越来越大,托管着大量主机。由于海量数据的生成,数据中心面临着 ToR 交换机之间的巨大通信量需求的严峻挑战[1]。因此,ToR 之间的业务流调度问题逐渐成为云服务供应商的一个难题,需要通过流准入决策来实现某些特定的目标,如收入、能量效率或资源利用率最大化[2-3]。

目前,已经有一些研究成果。如文献[4]研究了

收稿日期:2018-10-08。广东省自然科学基金项目(9451009001002711)。**黄润**,讲师,主研领域:智能算法,云计算应用。**肖志良**,副教授。

光数据中心网络的分组级调度,其特点是网络逻辑拓扑的频繁重构。文献[5]提出了针对数据中心环境的流调度算法,可应用于多根分层式树结构的动态流调度。该算法对网络链路上负载进行动态估计,并将数据流从重负载链路移动到轻负载链路,由此确保了网络链路间的负载平衡。文献[6]基于广域电分组交换网络背景,利用交换机发送的显式拥塞通知包,在广域网中跨多条路径执行动态流量工程。文献[7]将流调度问题转化成背包问题求解,提出基于离散粒子群 DPSO 的流调度算法,以两次迭代冲突流个数差值作为目标函数,但该方法需要分组交换机,由此增加了功耗和布线复杂度。文献[8]根据网络资源使用状态,提出自适应请求选择策略,即自适应从频谱资源方面选取请求。对选出来的请求进行重新服务,利用混合整型线性规划模型进行数学建模。

本文旨在最大化云服务供应商的总收入,同时满足波长连续性约束和带宽容量约束。其设计理念是在每个时隙后对光路进行动态重构,将不再使用的光路从逻辑网络拓扑中移除,同时,活跃的流也能够迁移到新光路中。在此基础上,设计了一个集成的 SDN 框架,以执行业务流调度和光路重构。仿真结果验证了本文算法的高效性。

1 数据中心的流调度问题

本文研究的两层数据中心架构如图 1 所示。假设数据中心中存在 M 个 ToR 交换机。每个 ToR 交换机通过光纤连接到核心光交换机,每条光纤最多可承载 W 个波长,即:一个 ToR 交换机可以通过光路同时到达 W 个 ToR 交换机。在没有波长转换器的情况下,穿过光交换机的两个 ToR 交换机之间的光路必须具备波长连续性。

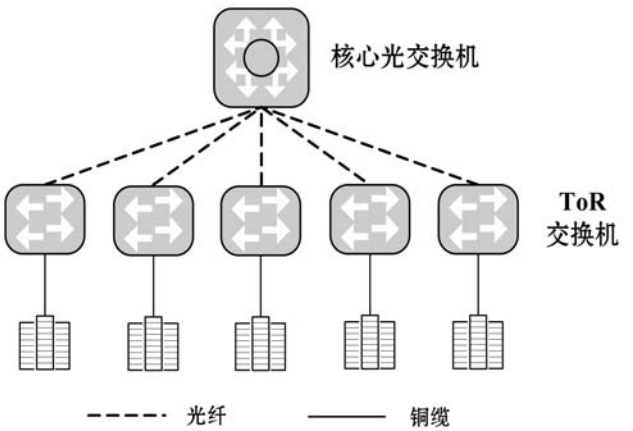

图 1 两层光数据中心架构

另外,每个业务流要求一个波长的最大带宽容量,一对 ToR 交换机之间所容纳的流数量,必须低于将这两个 ToR 连接到光交换机的光纤所承载的波长数量,否则应该丢弃一定数量的业务流。

设 F_t 为在时隙 t 开始时处于活动状态的所有业务流集合,即包括在时隙 $(t-1)$ 中网络容纳的所有流和所有被提交的流。每个流 $f \in F_t$ 表示为元组 (s_f, d_f, u_f, e_f),其中,s_f、d_f、u_f 和 e_f 分别表示流 f 在网络中的源 ToR 交换机、目的地 ToR 交换机、服务时间、已经过的服务时间。数据中心的目标是最大限度增加云服务供应商的长期总收入,该目标函数可表示如下:

$$R = \sum_{t=1}^{T} \left(\sum_{f \in F_t} c^{\text{unit}} x_f - \sum_{f \in F_t} c^{\text{unit}} e_f (1 - x_f) \right) \quad (1)$$

式中:c^{unit} 为每时隙容纳一个流的单位成本;x_f 为二元变量,表示流 f 是否被数据中心容纳过。式(1)第二项表示:如果以往时隙中被容纳过的流,在当前时隙中被拒绝,则从总收入中扣除通过该流从以往时隙中所得到的所有收入。

给定已经被容纳于数据中心内的流 f,设 y_f 为决策变量。准入决策和波长分配均需要满足光纤容量约束和波长连续性约束。光纤容量约束表示为[9]:

$$\sum_{f \in F_t, s_f = i} x_f + \sum_{f \in F_t, d_f = i} x_f \leqslant W \quad i = 1, 2, \cdots, M \quad (2)$$

波长连续性约束表示为:

$$\sum_{f \in F_t, s_f = i, y_f = w} x_f + \sum_{f \in F_t, d_f = i, y_f = w} x_f \leqslant 1$$
$$i = 1, 2, \cdots, M \quad w = 1, 2, \cdots, W \quad (3)$$

该约束确保了对于某个特定 ToR,将该 ToR 连接至核心光交换机的光纤所承载的波长 w 最多仅使用过一次。

现在定义光数据中心的流调度问题的形式化表达:给定一组业务流,每个流 f 表示为一个元组 (s_f, d_f, u_f, e_f),确定一个准入决策和一个波长分配策略,以使得服务供应商的长期总收入最大化。

$$\text{maximize}: R = \sum_{t=1}^{T} \left(\sum_{f \in F_t} c^{\text{unit}} x_f - \sum_{f \in F_t} c^{\text{unit}} e_f (1 - x_f) \right) \quad (4)$$

满足:

$$\sum_{f \in F_t, s_f = i} x_f + \sum_{f \in F_t, d_f = i} x_f \leqslant W \quad i = 1, 2, \cdots, M \quad (5)$$

$$\sum_{f \in \mathcal{F}_t, s_f = i, y_f = w} x_f + \sum_{F \in \mathcal{F}_t, d_f = i, y_f = w} x_f \leqslant 1$$
$$i = 1, 2, \cdots, M \quad w = 1, 2, \cdots, W \quad (6)$$

求解上述问题不具备计算可行性,原因是:1) 问

题的规模,即决策变量的数量非常大;2）由于输入业务流的动态到达,当前时隙的准入决策会影响到未来时隙的准入决策,由此影响到总体收入;3）两个 ToR 交换机之间的光路波长选择会影响到未来 ToR 连通性。因此,本文提出求解上述问题的启发式算法。

2　SDN 框架下的启发式流调度

2.1　最小拥塞和服务时间优先的调度

由于流的服务时间也将影响到 ToR 的未来连通性,服务时间越长,则 ToR 因为波长连续性约束而失去连通性的时间越长。因此,本文方法向服务时间较短的业务流给予较高的优先级,使其先于其他流被容纳。提出的最小拥塞和服务时间优先算法的伪代码如下:

Input:网络状态
Output:准入和波长分配决策
1. **for** $t=1\cdots T$ **do**
2.　　执行光路重构;
3.　　**while** $F_t^{\text{in}}\neq 0$ **do**
4.　　计算式(7)定义的 $C_f, f\in F_t^{\text{in}}$;
5.　　得到具有最小拥塞因子和服务时间的流 f;
6.　　**if** 可在 ToR s_f 和 ToR d_f 间建立起一条光路
7.　　则确定最优波长;
8.　　在 ToR s_f 和 ToR d_f 之间建立光路;
9.　　将流 f 容纳在网络中;
10.　　更新波长使用情况;
11.　　**else**
12.　　通知流 f 的拒绝消息;
13.　　**end if**
14.　　$F_t^{\text{in}}\leftarrow F_t^{\text{in}}\setminus\{f\}$;
15.　**end while**
16.　**return** 准入控制和波长分配;
17. **end for**

假设在时隙 t 开始时,对时隙 $t-1$ 过程中达到的所有业务流进行调度。已知当前网络状态和留在网络中的当前业务流集合,算法执行光路重构。设 F_t^{in} 为时隙$(t-1)$过程中到达的业务流集合,则提出的算法会逐个处理,直到所有的输入流均得到判定。在每次迭代中,算法先根据式(7),计算出所有剩余输入流的拥塞因子。

流 f 的拥塞因子[10]定义如下:

$$C_f=\frac{tf(s_f,d_f)}{|F_t^{\text{at}}|}\frac{\mathcal{A}(s_f,d_f)}{W} \tag{7}$$

式中:$tf(s_f,d_f)$表示 ToR s_f 和 ToR d_f 之间已经容纳的业务流数量;F_t^{at} 为网络中已经容纳的业务流的总数量;$\mathcal{A}(s_f,d_f)$表示将 ToR s_f 和 ToR d_f 连接到核心光交换机的光纤中可用的共同波长的数量。

然后,选择具有最低的拥塞因子和服务时间的流 f。若 ToR s_f 和 ToR d_f 之间可以建立起一条光路,即 ToR s_f 和 ToR d_f 之间存在共同波长,则确定新光路的最优波长,并对波长的使用情况进行更新以反映在下一次调度中。否则,该业务流将因为波长约束而被拒绝。算法继续处理下一个输入流,直到完成所有流的处理。

2.2　基于拥塞的循环算法

应用上述算法会为网络建立较好的连通性,由此增加网络中容纳业务流的数量。然而,其可能会导致业务流饥饿问题,即:具有更短服务时间的新业务流的动态到达导致一些流永远无法被容纳到网络中。为实现流之间的公平性,本文使用了循环方法,而不是基于流服务时间进行优先级排序,即基于拥塞的循环 CBL 算法。首先,在计算出业务流的拥塞因子后,通过业务流的拥塞因子来选择要调度的流。由于许多流可能有着相同的源和目的地,这些流可能有相同的拥塞因子。然后将所有流分入不同集合中,每个集合有一个不同的拥塞因子。在应用循环调度时,在每个调度轮,从每个集合中选出一个流进行调度,且从具有最低拥塞因子的集合开始。

表 1 给出了根据不同优先级方法得出不同调度顺序的输入流样例。若应用 2.1 节的算法,其调度顺序为 f_1、f_2、f_3、f_4。若使用循环方法,其调度顺序为 f_1、f_3、f_2、f_4。已知将 ToR 2 连接至核心光交换机的光纤具有 2 个可用波长,则根据算法 f_3 和 f_4 将被拒绝。若根据循环方法则将拒绝 f_2 和 f_4。因此在应用算法时,服务时间较短或较长的流被容纳于网络中的机会均等。

表 1　使用不同的优先方法进行流调度的样例

流	源	目的地	服务时长	拥塞因子
f_1	ToR 1	ToR 2	3	0.3
f_2	ToR 1	ToR 2	4	0.3
f_3	ToR 2	ToR 3	5	0.4
f_4	ToR 2	ToR 3	6	0.4

2.3　基于 SDN 的流调度框架

在 SDN 控制器下,光数据中心流调度的总体框架设计如图 2 所示。

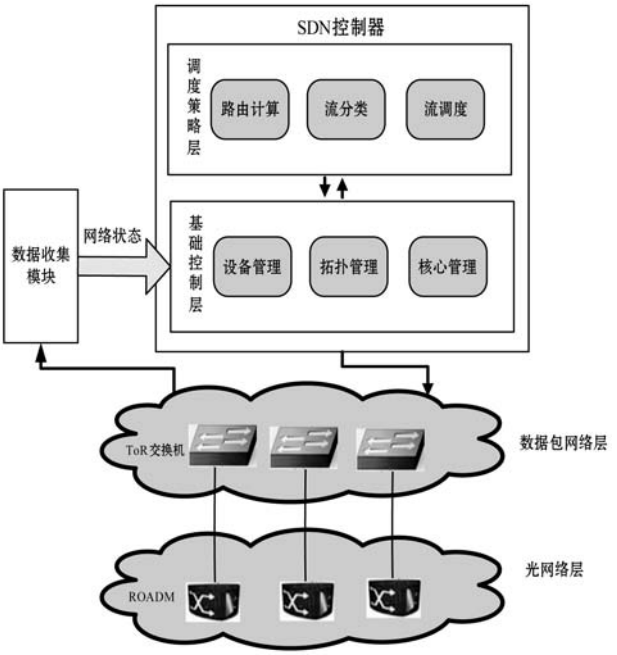

图 2 光数据中心流调度的 SDN 框架图

数据收集模块接收到达每个交换机的输入流信息。基于网络状态和数据收集模块所接收到的输入流信息、控制器运行的调度算法,取决于云服务供应商选择的调度算法。波长分配被转发至光路配置模块,以调用电路交换,并在 ToR 交换机之间建立光路。准入决策则发送至 ToR 交换机,以开始准入流的数据传输并丢弃其他流。云服务供应商还可决定运行算法的频率,以实现性能最大化,即通过每个时隙的持续时间实现性能最大化。在 SDN 的支持下,上述框架可利用支持 OpenFlow 的交换机[11]实现,且文献[12]已经证明了在光网络上进行 SDN 控制的可行性,由此可以灵活地执行光路配置。图 2 中的可重构光分插复用器[13](ROADM)是光网络的一个重要光子交换设备。通过波长选择光交换机,ROADM 能够对光路丢弃或添加多个波长,且不需要将光信号转换为电信号。由于 ROADM 设计了一个管理控制平面,并提供 OpenFlow 协议,使得 SDN 控制器可以远程控制波长的变化。

在 SDN 框架下,MC-STP 的算法流程步骤总结如下:

1) SDN 控制器通过光网络层中 ROADM 的 OpenFlow 协议,远程执行光路重构。

2) 支持 OpenFlow 的 ToR 交换机计算拥塞因子,得到具有最小拥塞因子和服务时间的流。

3) 如果能在某两个 ToR 交换机间建立一条光路,则确定最优波长。

4) SDN 控制器通过调度决策层将该流容纳在网络中,并更新波长使用情况。

5) SDN 控制器通过基础控制层将波长分配转发至光路配置模块。

在 SDN 框架下,CBL 与 MC-STP 不同的主要体现在:CBL 在每个调度轮中,从每个集合中选出一个流进行循环调度,而不是基于流服务时间进行优先级排序。这样可以避免业务流饥饿问题。

本文两个算法的时间复杂度为 $O(n\log n)$,其中 n 为输入流的总数量。因此,在执行调度时,本文算法不会为控制器带来较大开销,在 SDN 框架下具有一定的可行性。

3 性能分析

3.1 设 置

本文研究的光数据中心网络及其架构如图 1 所示,核心交换机连接着 48 个 ToR 交换机,将 ToR 交换机连接至核心交换机的光纤承载了 25 个波长。每个波长的容量为 1 Gbit/s。从 ToR 集合中随机选出源 ToR 和目的地 ToR 以生成输入流,从时隙[5, 20]范围中随机选出每个流的服务时长,并假定每个流要求一个波长的整个容量。

本文对以下 4 个算法进行性能检验:

1) 本文最小拥塞和服务时间优先(MC-STP)算法:服务时间短和拥塞因子小的业务流将得到更高的优先级。2) 本文基于拥塞的循环(CBL)算法:确保流之间的公平性。3) 文献[4]基于端到端(E2E)的流调度:使用先到先服务原则,基于业务流到达顺序对输入流进行调度。4) 文献[7]基于离散粒子群(DPSO)算法的流调度:应用智能算法,基于网络状态判定每个流的准入或丢弃。

所有算法均运行 2 000 个接收输入流的时隙。使用以下度量对算法进行性能评价:

1) 拒绝率:丢弃流的数量与输入流数量间的比率。

2) 平均收入:根据式(1)计算。

3) 波长利用率:$2L/(MW)$。其中:L 为网络中创建的光路总数量;M 为 ToR 的总数量;W 为光纤承载的波长数量,取算法运行 2 000 个时隙的均值。

3.2 结果与分析

3.2.1 总体性能

图 3 给出了相对于每时隙到达的不同流数量,各算法所生成的拒绝率。可以看到,MC-STP 和 CBL 算法性能优于其他算法。在拒绝率低于 10% 的区间(实际应用有意义的情形),MC-STP 和 CBL 的拒绝率明显低于 E2E[4] 和 DPSO[7]。随着流数量的增加,在不采用任何优先排序方法的情况下,容纳数据流会导致网

络性能变得很低,因为容纳某个特定流会造成整个网络堵塞,使得随后到达的所有流均被丢弃。此外,应用 CBL 为输入流之间带来公平性,但会造成拒绝率小幅上升。

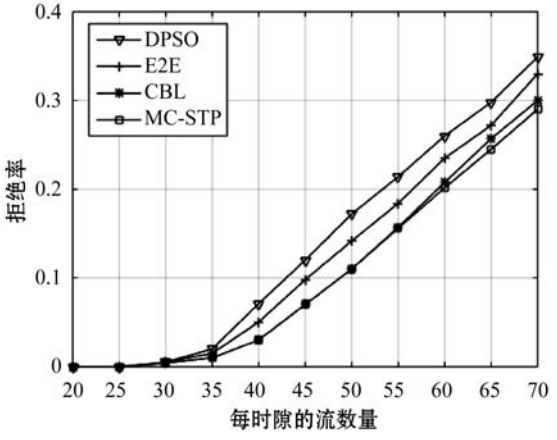

图 3　业务流到达的拒绝率变化情况

图 4 给出了从经济角度看,2 000 个时隙后云服务供应商得到的平均收入。结果表明:与 E2E[4]、DPSO[7]相比,MC-STP 和 CBL 最高提升了 3% 的平均收入。从中还可观察到,CBL 在平均收入方面的性能稍优于 MC-STP。这是因为 MC-STP 算法给予服务时长较短流更高的优先度。由于业务流的动态到达,造成波长(光路)利用率碎片化,短空闲时间更多。由此,容纳服务时间更长的流将使得平均收入更加稳定。结果表明,MC-STP 的拒绝率低于 CBL,但 MC-STP 产生的收入也低于 CBL。供应商可根据需要选择合适的算法集成到所提框架中进行流调度。

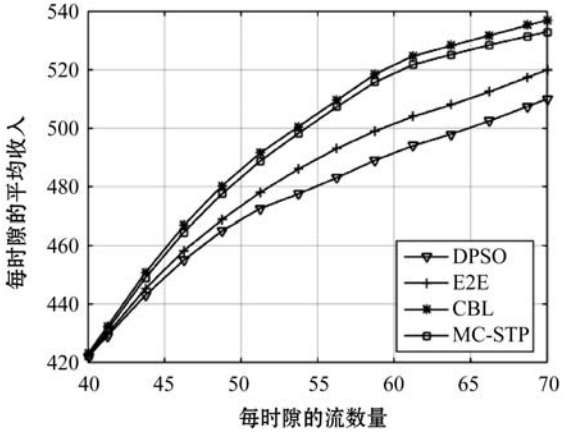

图 4　长期运行后的平均收入

图 5 给出了各算法的波长利用率。结果表明,本文算法对光纤载波的利用较好。当每时隙到达 70 个流时,本文算法将波长利用率从 86% 提升至 89%,这一提升得益于本文提出的优先方法。由于每条光路涉及到将源 ToR 和目的地 ToR 连接至核心光交换机的两条光纤,且优先容纳具有最小拥塞因子的流,实现了对 ToRs 相关光纤中的共同波长的更好利用。由此避免了源 ToR 的光纤中的可用波长在目的地 ToR 的光纤中不可用的情况。

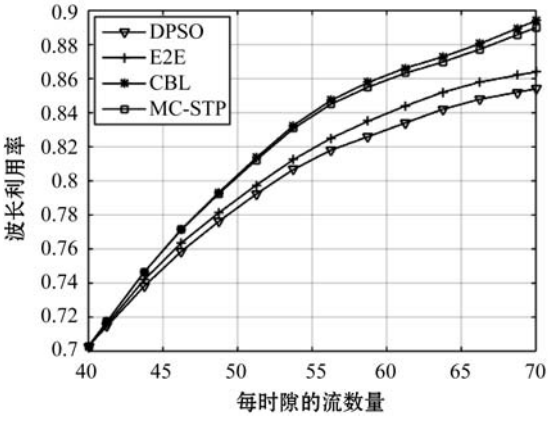

图 5　网络的波长利用率

3.2.2　波长再分配

本文在两个场景中运行所提算法,并测量拒绝率。1)带波长再分配(用 -1 标识):在每个时隙结束时,从网络的逻辑拓扑中移除不再需要的光路,而且对现有业务流所使用的所有活动光路进行修改,并再次分配新的波长。2)不带波长再分配(用 -2 标识):仅移除不再需要的光路。

本文算法在上述两个场景运行时的拒绝率如图 6 所示。结果表明:应用波长再分配能够显著提升性能。在拒绝率低于 10% 的区间内,带波长再分配的算法 MC-STP-1 和 CBL-1 的拒绝率明显低于不带波长再分配的算法(MC-STP-2 和 CBL-2)。当每时隙到达 70 个时隙时,与不带波长再分配相比,带波长再分配的算法能够将拒绝率最高降低 16%。如前文所述,波长再分配能够提升 ToR 间的连通性,以便容纳后续到达的更多流。同时,由于波长连续性约束,波长再分配增加了任何一对 ToR 的光纤中可用波长数量。在不带波长再分配的算法中,ToR 间的可用波长数量较少,因此拒绝率较高。值得一提的是,两类方法均不会增加或减少 ToR 连接到核心光交换机的光纤可用波长数量,但会影响 ToR 间的一些性能。

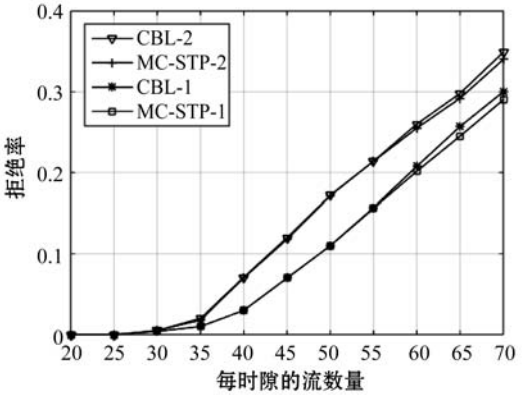

图 6　使用或不使用波长再分配时的拒绝率比较

3.2.3 增量拓扑与可重构拓扑的比较

通过仿真评价了本文算法使用增量拓扑时的性能。在增量拓扑中,即使不再需要一条光路,也不会将其从逻辑拓扑中移除。增量拓扑场景中,波长再分配也被禁用。比较结果如图 7 所示,其结果符合预期,增量拓扑案例(MC-STP-增量)的拒绝率大幅上升。当每时隙到达 20 个流时,可重构拓扑案例(MC-STP-重构)中未出现拒绝情况,而增量拓扑案例中拒绝率则达到 50%。造成这一现象的原因是业务流的动态到达,以及业务流的源 ToR 和目的地 ToR 的随机性,使得对于不同源与目的地 ToR,可用光路非常少,而其他 ToR 没有可用的光路来容纳到达的业务流,这在实际应用场景中是必须要避免的。而 MC-STP-重构的拒绝率大部分情况下低于 10%,在实践中可用。

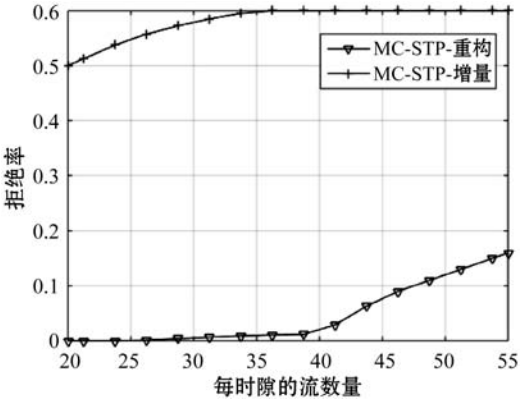

图 7 增量网络拓扑与可重构网络拓扑的性能比较

4 结 语

本文研究了数据中心网络中的流调度问题,并针对该问题提出的一个优化方法,以最大化云服务供应商的长期收入。由于流调度问题不具备计算可能性,本文采用了业务流调度的启发式算法,利用拥塞因子来确定业务流调度顺序。此外,还利用优化函数来确定最优波长,以确保 ToR 交换机的连通性。本文算法不但保证了云服务供应商的最大收入,而且确保业务流之间的公平性。仿真结果表明,本文算法的性能优于其他算法,最高能够降低 10% 的拒绝率。

参 考 文 献

[1] 王啸, 方滨兴, 刘培朋,等. Tor 匿名通信网络节点家族的测量与分析[J]. 通信学报, 2015, 36(2): 80-87.

[2] 孙迅. 异构蜂窝网络能量效率问题研究[D]. 南京:南京大学, 2015.

[3] Rahmani D, Ramezanian R. A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study[J]. Computers & Industrial Engineering, 2016, 98(12): 360-372.

[4] Wang C H, Javidi T, Porter G. End-to-end scheduling for all-optical data centers[C]//Computer Communications. IEEE, 2015: 406-414.

[5] Zhang X. Adaptive flow scheduling for modular datacenter networks[J]. Peer-to-Peer Networking and Applications, 2016, 10(5): 1-10.

[6] 卜佑军, 朱珂, 贺炜,等. 多路径网络中联合拥塞控制和流量工程的优化模型研究[J]. 数学的实践与认识, 2013, 43(21): 116-123.

[7] 林智华, 高文, 吴春明,等. 基于离散粒子群算法的数据中心网络流量调度研究[J]. 电子学报, 2016, 44(9): 2197-2202.

[8] 方文坚. 光数据中心网络中针对 NFV 服务链部署的多维资源联合优化研究[D]. 合肥:中国科学技术大学,2017.

[9] 周宇萌,邱昆,许渤,等. 超大容量光交换机中器件约束的 Clos 网络路径设计[J]. 光学学报,2013,33(8):43-48.

[10] 刘增全, 祁建清, 蒋昊. 一种抗链路拥塞攻击的匿名通信技术[J]. 信息工程大学学报, 2017, 18(3): 333-337.

[11] Batista B, Fernandez M. PonderFlow: A New Policy Specification Language to SDN OpenFlow-based Networks[J]. International Journal on Advances in Networks & Services, 2014, 7(4): 163-172.

[12] Parulkar G, Tofigh T, Leenheer M D. SDN control of packet over optical networks[C]//Optical Fiber Communications Conference and Exhibition. IEEE, 2015: 12-20.

[13] 张明, 董章龙, 全必胜,等. 新型可重构光分插复用器及其联网性能分析[J]. 光电工程, 2013, 40(12): 73-79.

(上接第 148 页)

[16] Ma X, Zhang K, Bai B, et al. Serial Concatenation of RS Codes with Kite Codes: Performance Analysis, Iterative Decoding and Design[EB]. arXiv:1104.4927, 2011.

[17] Plank J S, Xu L. Optimizing Cauchy Reed-Solomon Codes for Fault-Tolerant Network Storage Applications. [C]// IEEE International Symposium on Network Computing & Applications. IEEE Computer Society, 2006.

[18] Ko K, Oh I, Ko D, et al. Improving performance of Reed-Solomon decoder by error/erasure correction[C]// IEEE International Symposium on Signal Processing & Information Technology. IEEE, 2012.

[19] Yi C, Zhang T Q, Hu R, et al. An interleaving approach of enhancing the performance of RS codes in two dimensional space[C]// International Congress on Image & Signal Processing. IEEE, 2013.