

基于灵活栅格光网络的节点聚合虚拟映射算法

郑冬,刘逢清*

(南京邮电大学 电子与光学工程学院,南京 210023)

摘要:针对光交换节点处配有大量计算和通信资源数据中心的特点,提出一种虚拟节点聚合的虚拟网络映射算法。通过把同一虚拟网络的多个虚拟节点聚合映射到一个光节点上,聚合节点之间的通信由光节点配置数据中心的内部实现,非聚合节点之间的通信由光节点之间的通信带宽实现,以减少虚拟网络映射对光网络带宽资源的需求,从而降低业务阻塞率。借助 MyEclips 和 JAVA 语言进行编程仿真分析,仿真结果表明:所提算法可以有效降低虚拟网络请求的阻塞率。

关键词:灵活栅格光网络;数据中心;节点聚合;虚拟网络映射

中图分类号:TP301.6 文献标识码:A 文章编号:1002-5561(2019)06-0054-05

DOI:10.13921/j.cnki.issn1002-5561.2019.06.013

开放科学(资源服务)标识码(OSID):



Node aggregation virtual mapping algorithm based on elastic grid optical network

ZHENG Dong, LIU Fengqing*

(College of Electronic and Optical Engineering, Nanjing University of Posts & Telecommunications, Nanjing 210023, China)

Abstract: A virtual network mapping algorithm based on virtual node aggregation is proposed for optical switching nodes equipped with a large number of computing and communication resources data centers. By aggregating and mapping multiple virtual nodes of the same virtual network onto an optical node, the communication between aggregating nodes is realized by the optical node configuring the data center internally, and the communication between non-aggregating nodes is realized by the communication bandwidth between optical nodes, so as to reduce the requirement of virtual network mapping for optical network bandwidth resources and reduce the traffic blocking rate. With the help of MyEclips and JAVA language, the simulation results show that the proposed algorithm can effectively reduce the blocking rate of virtual network requests.

Key words: elastic grid optical network; data center; node aggregation; virtual network mapping

0 引言

随着各种互联网业务的普及,网络带宽的需求以摩尔定律的速度迅速增长,传统的波分复用(WDM)系统渐渐无法满足新业务带宽的需求和灵活性的要求^[1,2]。与之相比,灵活栅格光网络具有更小的频谱分配粒度和灵活的分配方案,能有效解决未来光传送网的带宽问题^[3,4]。

作为解决传统网络僵化问题的网络虚拟化技术日渐成熟。虚拟子网的映射问题,即如何高效地将虚

拟节点(或虚拟链路)映射到物理节点(或物理链路),是网络虚拟化的关键技术之一^[5,6]。目前,底层网络(基于 2/3 层网络)的研究已经相对成熟,基于传统光网络的研究工作也比较完善。但是,基于灵活栅格光网络中的虚拟网络映射研究还比较缺乏。因此,研究灵活栅格光网络下的虚拟网络映射问题具有一定的现实意义。在灵活栅格光网络与数据中心聚合的底层网络中进行虚拟子网映射时,与光交换节点直连的数据中心不仅可以提供虚拟网络请求中虚拟节点需要的计算资源,还可以使用内部通信的方式提供带宽资源。此外,基于灵活栅格光网络的虚拟映射算法,也需要考虑频谱一致性和频谱连续性的限制条件^[7]。

因此,本文在底层网络是灵活栅格光网络的前提下,提出一种可以将虚拟网络请求中不同节点映射到同一个光交换的节点聚合映射算法,有效降低虚拟网络请求的阻塞率。

收稿日期:2019-01-20。

作者简介:郑冬(1992-),男,安徽省池州人,南京邮电大学电子与光学工程学院、微电子学院硕士研究生。获得研究生入学二等奖学金。主要研究方向为灵活栅格光网络中的网络虚拟化。

* 通信作者:刘逢清(E-mail:6571667@qq.com)。



1 算法思想及设计

基于灵活栅格光网络与数据中心聚合的底层网络,当虚拟子网在映射时,分为节点映射(在实际底层网络中寻找满足条件的物理节点来承载虚拟网络请求的虚拟节点)与链路映射(在实际底层网络中寻找满足虚拟网络请求中虚拟链路资源条件的物理链路)2个子过程。前人的映射算法是在映射的第一个阶段(即节点映射阶段)时,考虑将虚拟网络请求中每一个虚拟节点都映射到不同的物理节点上。而对于灵活栅格光网络与数据中心融合的底层网络,其网络拓扑结构如图1所示,则可以考虑将部分有连接的虚拟节点映射到同一个物理节点(即同一个光交换节点)上,这是由于数据中心其内部结构(如图2所示)拥有通信所需的带宽资源(即可以通过内部通信的方式提供相应虚拟节点所需的带宽资源),并且对虚拟网络的用户来说,他们只关心自己的功能能否被实现,而不关心它在物理网络中是如何实现的(即不关心是多个虚拟节点映射到一个还是多个物理节点上)。实际上,一个虚拟节点一般有多个可映射的物理节点,它可选择其中一个;多个虚拟节点之间会有共同的可映射物理节点。

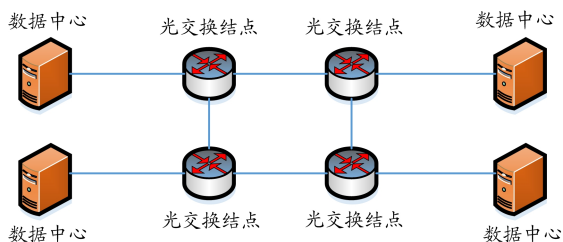


图1 底层网络拓扑结构

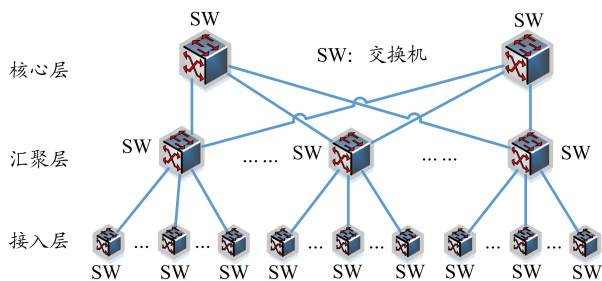


图2 数据中心内部结构

底层网络的抽象表示如图3(a)所示,编号1~4代表光交换节点,每一个节点附近数字表示其数据中心能够提供的CPU计算资源大小,每一条通信线路上的数组代表可使用的栅格资源情况,1表示占用,0表示未分配。根据频谱一致性和频谱连续性的原则,在分配带宽资源时,一条路径上的所有链路都必须分配连

续且相同起始点的栅格资源。虚拟网络请求逻辑图如图3(b)所示,1~3代表请求的虚拟节点,每一个虚拟节点附近数字表示该虚拟节点请求的CPU计算资源大小,通信线路上的数字表示请求的带宽资源。本文不考虑调制方式的影响,统一用所需的栅格个数表示,这不会对算法结果产生影响。

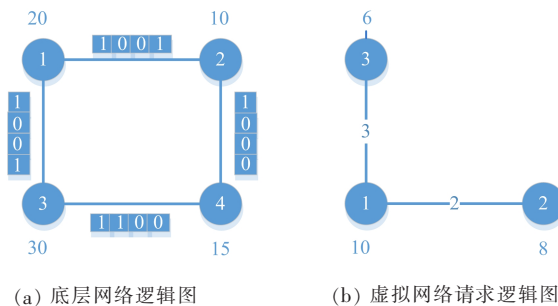


图3 底层网络和虚拟网络的逻辑示意图

一个虚拟网络请求进行节点聚合映射的结果如图4所示。可以看出,虚拟节点2、3在聚合映射时,聚合形成了一个虚拟节点(即虚拟节点2、3映射到同一个物理节点上),物理节点的数据中心以内部通信方式提供虚拟节点2、3间的通信资源要求,不占外部带宽资源(即外部带宽资源被节省下来)。

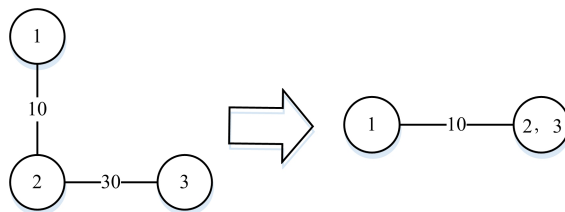


图4 结点聚合映射

根据上述理论基础,可以得到约束条件:

◆节点映射约束:

$$\sum_{v_r} M_{v_r}^{v_n} = 1, \forall v_n \quad (1)$$

$$\sum_{v_n} M_{v_r}^{v_n} \geq 0, \forall v_r \quad (2)$$

由式(1)可知,在进行虚拟子网映射时,对于虚拟网络请求的任何一个虚拟节点 v_n ,都必须映射到一个物理节点 v_r 上,若找不到物理节点承载该虚拟节点,则映射失败。

由式(2)可知,对于底层网络中任何一个物理节点 v_r 而言,可以不承载或者承载多个虚拟节点。每一个物理节点能承载最多的虚拟节点数为该物理节点计算资源,同时满足虚拟节点计算资源请求的个数。

◆链路映射约束:

$$\sum_{g_v} P_g^g = 1, \forall g \quad (3)$$

$$\sum_{e_v} U_{e_v}^g \geq 1, \forall g_v \quad (4)$$

$$\sum_{e_v} U_{e_v}^g \leq N, \forall g_v \quad (5)$$

式(3)中, $P_g^g=1$ 表示虚拟链路 g_v 使用物理路径上起始点为 g 的栅格, $P_g^g=0$ 表示虚拟链路 g_v 不使用物理路径上起始点为 g 的栅格。因此,在链路映射时,一个虚拟链路映射的物理路径只占用起始点为 g 的栅格,不占用其它起始点的栅格资源。

式(4)中, $U_{e_v}^g=1$ 表示虚拟链路 g_v 映射到物理链路 e_v (e_v 指 2 个直连物理节点间的链路), $U_{e_v}^g=0$ 表示虚拟链路 g_v 没有映射到物理链路 e_v 。因此,在链路映射时,虚拟链路映射的物理路径长度必须大于 1,即虚拟链路 g_v 映射的物理路径必须最少包含 2 个物理节点。

式(5)中, N 表示物理网络中所有物理链路条数。因此,在链路映射时,虚拟链路映射的物理路径长度必须小于物理网络链路条数,即虚拟链路 g_v 映射的物理路径最多包含所有物理节点。

1.1 节点聚合映射算法

根据上述的模型约束和理论基础,本文提出一种节点聚合启发式映射算法。由于该算法不是整数线性规划(ILP)算法,因而上述约束条件式(1)~式(5)不参与运算,只是作为启发式算法逻辑上的参考,保证结果的正确性。伪代码如下:

```

1: for 每一个虚拟网络 do
2:   根据虚拟请求中所有节点的度,降序排序形成虚拟节点集合  $V_c$ 
3: end for
4: for 每一个虚拟节点  $v_n \in V_c$  do
5:   找出底层网络中满足虚拟节点  $v_n$  度的所有物理节点,并根据物理节点当前可以使用的计算资源降序排序形成待选物理节点集合  $Select(v_n)$ 
6:   在集合  $Select(v_n)$  中删去在本次虚拟网络请求中已经使用过的物理节点集合  $Used$  中的所有节点
7:   for 每一个物理节点  $v_r \in Select(v_n)$  do
8:     计算映射选择因子  $SIF(v_r)$ , 根据计算值,降序排序存入集合  $Mapping(v_n)$  中
9:   end for

```

```

10: 在集合  $Mapping(v_n)$  和  $Select(v_n)$  中,将索引序号之和最小的物理节点  $v_{min}$  找出来,用该物理节点承载虚拟节点  $v_n$ , 并且加入已使用物理节点集合  $Used$ 
11: 获取最大聚合次数  $Maxmerge$ , 初始化本次节点聚合最大可用计算资源  $CPU_{MAX}$  为  $cpu(v_{max})$ , 及当前计算资源消耗总量  $CPUTotal$  为  $cpu(v_n)$ 
12: 将虚拟请求中与虚拟节点  $v_n$  相连且还未做节点映射的虚拟节点按照节点间链路代价进行降序排序形成集合  $V_d$ 
13: for 每一个虚拟节点  $v_{si} \in V_d$  do
14:   if  $Maxmerge > 0$  &&  $(cpu(v_{si}) + CPUTotal) / CPU_{MAX} < \text{合并安全系数 } MSF$  then
15:     将该节点映射到物理节点  $v_{min}$ , 修改  $CPUTotal = CPUTotal + cpu(v_{si})$ ,  $Maxmerge = Maxmerge - 1$ , 在集合  $V_c$  中删除待映射虚拟节点  $v_{si}$ 
   else
     退出结点聚合映射过程,进行下一个虚拟节点映射
   end if
end for
16: end for
17: if 虚拟网络求中所有虚拟节点都映射成功 then
18:   return 映射成功,分配计算资源并记录映射关系
19: else
20:   return 映射失败
21: end if

```

算法 1~6 行:对于每一个虚拟节点 v_n ,在底层网络中找到满足该虚拟节点度和 CPU 资源要求的所有物理节点,并将找出的物理节点按照计算资源降序排序,排序越靠前,说明物理节点资源越丰富,优先考虑将虚拟节点映射到该节点。

算法 7~10 行:映射选择因子 $SIF(v_r)$ 的计算方式为: $SIF(v_r) = \sum_{i=1}^k M_i^{v_n} / \sum_{i=1}^k P_{v_i}^{v_r} \times M_i^{v_n}$, $1 \sim k$ 表示之前已经映射完成的 k 个虚拟节点; $M_i^{v_n}$ 表示在虚拟网络请求中,当前需要映射的虚拟节点 v_n 和已经完成映射的虚拟节点 i 之间的关系,1 表示直连,0 表示无直接连接关系; $P_{v_i}^{v_r}$ 表示虚拟节点 i 映射的物理节点 v_i 与待选物理节点 v_r 的最短路径,这里表示跳数。因此, SIF 值越大表明选择该节点会生成比较集中的网络,在链路映射阶段时,缩短映射的物理链路路径,节省链路带宽资源具

有较大的好处。程序会根据物理节点的 SIF 和计算资源量综合选择最优的节点进行映射。当映射第一个虚拟节点或者当前要映射的虚拟节点没有与之前已映射完的任何一个虚拟节点有直连关系时,无法计算 SIF ,此时选择待选节点中带宽资源最丰富的物理节点承载该虚拟节点,即在待选的物理节点中,找到具有最大的最小栅格资源边的物理节点承载该虚拟节点。

算法 11~15 行:这是节点聚合映射阶段,在已经将虚拟节点 v_n 映射到物理节点的前提下,在虚拟网络请求中找到与虚拟节点 v_n 直连的所有虚拟节点,并按照 2 个虚拟节点间的带宽资源请求大小进行降序排列。当按序取出的虚拟节点满足合并次数限制和资源限制时,可以用同一个物理节点承载该虚拟节点。事先定义合并安全系数 MSF , MSF 越大越容易进行节点聚合,但是考虑到节点聚合会导致一次映射中同一个物理节点计算资源和带宽资源使用过多,所以 $Maxmerge$ 和 MSF 不宜过大。

算法 16~21 行:只有当所有虚拟节点映射成功,节点映射阶段才会成功,否则本次虚拟子网映射失败。

1.2 链路映射算法

```

1: for 每一个虚拟请求 do
2:   按照虚拟链路请求的带宽大小将虚拟链路降序
   排序存入集合  $G_v$ 
3: end for
4: for 每一条虚拟链路  $g_v \in G_v$  do
5:   根据节点映射阶段结果找到底层网络中对应的
   物理节点,采用  $K$  条最短路径算法(KSP)计算
   两端点的  $K$  条最短路径。
6:   for 每一条路径  $i \in K$  do
7:     在物理路径中寻找所有公共可用连续栅格
     块,将所有找到的栅格块按照栅格连续个数
     升序排序形成集合  $Grind(g_v)$ 
8:     if 能在集合  $Grind(g_v)$  找到满足要求的连续
       栅格块
           虚拟链路  $g_v$  映射成功
9:       end if
10:    end for
11:  if  $K$  条路径都找不到满足要求的连续栅格块 then
12:    虚拟链路  $g_v$  映射失败
13:  end if
14: end for
15: if 所有虚拟链路映射成功 then

```

```

16:   return 虚拟网络映射成功,分配资源并记录
17: else
18:   return 虚拟网络映射失败
19: end if

```

算法 1~3 行:首先找出虚拟网络请求中需要映射的所有虚拟链路,并根据带宽资源请求大小降序存入集合,这是为了优先将带宽资源大的请求进行映射,有助于算法映射的运行效率。

算法 4~5 行:对于每一条虚拟链路,先根据节点映射阶段结果,在底层网络中找到虚拟链路映射的 2 个物理节点,若是同一个物理节点,则使用数据中心内部通信方式提供节点间通信所需的带宽资源,不用再分配栅格资源,并进行下一条虚拟链路映射,否则根据 KSP 算法进行 K 条最短路径查找。

算法 6~14 行:找到 K 条路径后,按照路径长度升序排序。遍历路径,先找出物理路径上所有公共可用连续栅格块,栅格块按照栅格的连续数量进行升序排序,然后找出第一个满足要求的栅格块进行分配,若在所有物理路径上都找不到满足要求的公共可用连续栅格块,则该虚拟链路映射失败,即虚拟网络映射失败。经过上述步骤找到的路径是满足条件的最短路径,找到的栅格块也是最小满足要求的栅格块,这在节约物理链路带宽资源和避免频谱资源碎片化方面起到了一定的作用。

算法 15~19 行:所有虚拟链路映射成功,即虚拟网络映射成功。此时将底层网络资源分配出去并记录,否则映射失败,不对底层网络进行任何的操作。

2 算法验证及分析

本文根据上述伪代码,使用 JAVA 语言进行代码编写,代码定义的底层网络拓扑如图 5 所示。定义每个物理节点能够提供的计算资源量为 60,每条物理链路上的栅格资源数为 40。虚拟网络请求的网络拓扑结构由程序随机产生,每个虚拟网络的节点个数随机取 2~4 个,每个节点的边随机取 1~ $N-1$ 个(N 为总节点个

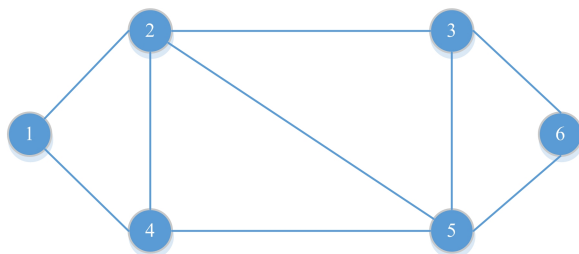


图 5 6 个节点的底层网络拓扑

郑冬,刘逢清:基于灵活栅格光网络的节点聚合虚拟映射算法

数),每个虚拟节点的计算资源随机取 2~4 个计算单元,每条边的带宽资源随机取 1~4 个栅格。虚拟请求动态到达和离去,由程序控制。

定义节点聚合阶段节点最大聚合次数 $Maxmerge$ 为 2,合并安全系数 MSF 为 60%,链路映射阶段 KSP 算法 K 值为 2。本文使用非节点聚合映射算法为对照算法,该算法可使用相同的算法程序,只需要将节点最大聚合次数 $Maxmerge$ 设定为 0 即可。

基于上述条件,本文采用 JAVA 程序仿真得出非节点聚合映射算法和节点聚合映射算法的阻塞率,程序设定虚拟网络请求个数为 5000,其中阻塞率为 10 次实验的平均值,结果如表 1 所示。可以看出,对于同一种映射算法而言,随着全网业务强度的升高,阻塞率在上升,这是因为业务强度升高会导致底层网络资源紧张,从而无法满足后续某些虚拟网络请求的资源请求。而在相同的业务强度下,节点聚合映射算法的阻塞率会低于非节点聚合映射算法阻塞率,这是由于节点聚合映射算法在聚合节点时,节约了链路的栅格资源,其承载能力优于非节点聚合映射算法,进而降低了阻塞率。

表 1 2 种算法阻塞率对比

算法	业务强度	阻塞率
非节点聚合映射	10	0.058
	20	0.079
	30	0.110
	40	0.171
	50	0.215
节点聚合映射	10	0.018
	20	0.031
	30	0.042
	40	0.068
	50	0.109

在不同 K 值情况下,2 种算法的阻塞率对比如图 6 所示。可以看出,同一种业务强度下,节点聚合算法的阻塞率低于非节点聚合映射算法, K 值为 4 的节点聚合映射算法优于 K 值为 2 的节点聚合映射算法。这是由于 K 值的增加,在链路映射阶段时会得到更多的待选物理路径,即增大了链路映射阶段的寻址范围,进而降低了阻塞率。

虽然由于本实验中底层网络资源不够丰富,导致阻塞率较大,但是比较 2 种算法,能够得到节点聚合映射算法的阻塞率低于非节点聚合映射算法的阻塞率。故当底层网络资源丰富时,应用节点聚合映射算法进行映射得到的整体阻塞率能被用户接受。

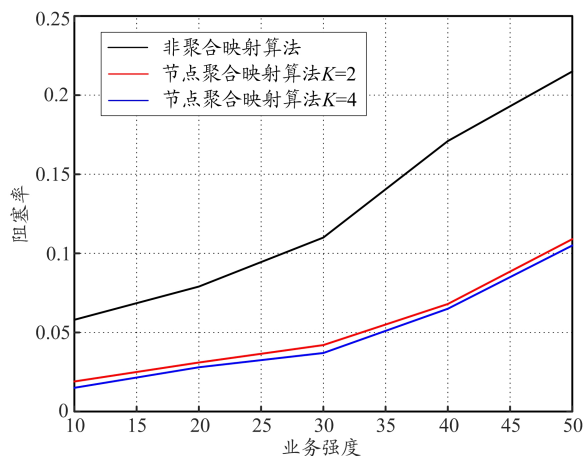


图 6 K 值不同时,2 种算法的阻塞率对比

3 结束语

本文研究了基于灵活栅格光网络与数据中心聚合的底层网络,提出了一种利用数据中心内部通信特点来进行节点聚合映射的虚拟映射算法,并通过仿真得到虚拟网络请求的映射阻塞率,将其与非节点聚合映射算法进行比较。结果表明:基于灵活栅格光网络与数据中心聚合的底层网络应用此节点聚合映射算法时,可有效降低虚拟网络请求的映射阻塞率,提高了底层网络的承载能力。本文提出的算法对未来虚拟光网络映射算法的研究具有一定的借鉴作用。

参考文献:

- [1] 杨焱. 面向云计算的弹性光网络虚拟化技术研究[D]. 北京:北京邮电大学,2015.
- [2] 续艳芳. 灵活栅格光网络中时变业务的频谱分配[D]. 南京:南京邮电大学,2016.
- [3] JINNO M, TAKARA H, KOZICKI B, et al. Spectrum-efficient and scalable elastic optical path network:architecture, benefits, and enabling technologies[J]. Communications Magazine, 2009, 47(1): 66-73.
- [4] JINNO M, TAKARA H, KOZICKI B, et al. Demonstration of novel spectrum-efficient elastic optical pathnetwork with per-channel variable capacity of 40 Gb/s to over400 Gb/s [C]. 34th European Conference on Optical Communication 2008, September 21-25, 2008, Brussels, Belgium. Brussels: IEEE, 2008: 1-2.
- [5] 王守宇. 频谱灵活光网络中的层间映射机制与算法研究[D]. 北京:北京邮电大学,2014.
- [6] 喻玥. 软件定义光网络中资源虚拟化映射算法研究[D]. 北京:北京邮电大学,2015.
- [7] ANKITKUMAR P, PHILIP J, JASON J, et al. Defragmentation of the Transparent Flexible Optical WDM (FWD) Networks [C]. Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2011, March 6-10, 2011, Los Angeles, USA. Los Angeles: Optical Society of America, 2011.