

网络虚拟化环境中面向服务聚合的映射算法*

江逸茗, 兰巨龙, 程东年, 王志明

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

通讯作者: 江逸茗, E-mail: j8403@163.com

摘要: 可重构信息通信基础网络通过构建宏电路实现针对特定服务的传输质量优化. 由于该网络架构加入了对网络虚拟化技术的支持, 因此在对服务请求进行映射时, 若将类型相同的服务映射到同一组底层设备上, 则能够有效提升宏电路的优化效果. 针对该需求, 提出了一种面向服务聚合的虚拟网映射算法, 该算法综合考虑了底层设备的剩余资源和服务承载情况, 使得服务可被优先映射到承载同类型服务较多的底层设备上. 此外, 为了对算法的运行时间进行优化, 还提出了一种基于跳数约束的候选节点选取策略. 实验结果表明, 该算法不但在请求接收率和资源占用率等评价指标上有着较好表现, 而且还能有效提高映射后服务的聚合程度.

关键词: 网络虚拟化; 宏电路; 服务聚合; 映射算法

中图分类号: TP393

中文引用格式: 江逸茗, 兰巨龙, 程东年, 王志明. 网络虚拟化环境中面向服务聚合的映射算法. 软件学报, 2014, 25(6): 1328-1338. <http://www.jos.org.cn/1000-9825/4442.htm>

英文引用格式: Jiang YM, Lan JL, Cheng DN, Wang ZM. Mapping algorithm for service aggregation in network virtualization. Ruan Jian Xue Bao/Journal of Software, 2014, 25(6): 1328-1338 (in Chinese). <http://www.jos.org.cn/1000-9825/4442.htm>

Mapping Algorithm for Service Aggregation in Network Virtualization

JIANG Yi-Ming, LAN Ju-Long, CHENG Dong-Nian, WANG Zhi-Ming

(National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450002, China)

Corresponding author: JIANG Yi-Ming, E-mail: j8403@163.com

Abstract: Reconfigurable Fundamental Information Communication Network achieves optimum transmission quality for special services by establishing Omnibus Circuits. Due to the support of network virtualization in this architecture, optimum efficiency of Omnibus Circuits can be promoted by mapping services of the same type to one group of substrate devices. For this requirement, this paper designs a virtual network mapping algorithm for service aggregation. Because this algorithm has a comprehensive consideration of surplus resources and situation of mapped services, the services tend to be mapped on the substrate devices which host numerous services of the same type. Furthermore, a node selection strategy with hop constraint is proposed for reducing the running time of the algorithm. Simulation experiments show that this algorithm not only has a better performance in the acceptance ratio and the resource occupancy rate, but also substantially increases the degree of service aggregation.

Key words: network virtualization; omnibus circuit; service aggregation; mapping algorithm

近年来,随着互联网规模的不断增长和新型网络服务的不断涌现,现有网络中“尽力而为”的信息传递模式已经无法满足网络服务在移动性、服务质量保障等方面的实际需求.另一方面,互联网本身也存在着结构僵化、可扩展性差的缺点,为互联网的更新换代增加了一定困难.在此背景下,网络虚拟化成为了新一代互联网的发展方向^[1].利用网络虚拟化技术,服务提供商可以在共享的底层物理网络上创建多个虚拟网络(virtual network,简

* 基金项目: 国家重点基础研究发展计划(973)(2012CB315901, 2013CB329104); 国家自然科学基金(61309019, 61372121); 国家高技术研究发展计划(863)(2013AA013505, 2011AA01A103)

收稿时间: 2012-09-03; 修改时间: 2013-03-06; 定稿时间: 2013-06-21

称 VN),其意义包括 4 点:

- 1) 为底层网络的运营商提供了一种新的盈利模式;
- 2) 使多种不同架构的网络能够同时在一个底层网络上运行,提高了资源的利用率;
- 3) 为新型网络架构的实验和部署提供了可行途径;
- 4) 可为用户提供多种满足特定需求的专有服务网络,如 IPTV 或网络视频会议等等.

为了解决服务需求与网络能力之间的差距日益扩大的问题,可重构信息通信基础网络(reconfigurable fundamental information communication network,简称 RFICN)通过认知机制,将网络传输的基本要素抽象为元能力,并将网络服务的特定需求描述为多种元能力的集合.以元能力理论为基础,通过可重构路由交换平台实现网络资源的优化调度,以达到对服务按其类型进行相应优化的基本传输理念.有别于以往的分组传递通信模式,RFICN 提出了一种基于元能力理论的全新数据传递模式——宏电路(omnibus circuit).宏电路是为一组具有共同传输路径和相似传输需求的服务建立的自适应型虚电路,通过对元能力的智能调度,宏电路可以在链路层面为特定类型的服务提供传输质量的优化,从而实现基础网络在服务质量、安全、组播、移动性等方面的功能扩展与能力自适应,最终实现网络可重构的设计目标.

为了顺应下一代网络的发展趋势,RFICN 在其体系架构中加入了网络虚拟化技术的支持.在该技术框架内,通过分析虚拟网的拓扑信息和资源需求信息,可以获知网络的整体流量分布和需求分布情况,从而为构建宏电路提供必要数据.网络虚拟化技术支持在同一网络链路上承载多种不同类型的服务,但从宏电路构建的角度上来说,只有当某种类型的服务在一条链路上的流量远远大于该链路上其他服务的流量时,才有使用宏电路对该链路的进行传输质量优化的需要.因此在为服务请求分配资源时,可将具有相似传输需求的服务尽量映射到同一网络设备上,使得各个设备上的流量传输需求能够更加集中,从而提升宏电路的生成概率和对服务质量的改进效果,最终实现网络对服务的自适应承载,改善网络的整体传输质量.所以,针对 RFICN 的特殊需求,有必要提出一套新的虚拟网映射方法.

在网络虚拟化环境中,基础设施提供商(infrastructure provider,简称 InP)负责管理和运营底层网络,服务提供商(service provider,简称 SP)以虚拟网请求(virtual network request)的方式向 InP 申请网络资源并建立服务^[2,3].虚拟网的映射就是通过分配相应的底层网络资源来实现对虚拟网请求的承载,解决虚拟网映射问题是实现网络虚拟化技术的基础,但同时,虚拟网映射问题是 NP 难问题^[4].文献[5]利用混合整数规划,针对不同的应用场景,分别提出了确定型虚拟网映射算法(D-ViNE)以及随机虚拟网映射算法(R-ViNE).文献[5]以底层网络负载均衡为目标,设计了一种两阶段映射算法,该算法首先将虚拟节点映射到负载较轻且距离已被映射的虚拟节点较近的底层节点上,然后再使用最短路径算法映射虚拟链路.文献[7]提出了一种基于链路负载均衡度和节点负载均衡度的虚拟网构建算法.文献[8]提出底层物理网络中的一条虚拟链路可以由多个分割的物理路径组成,算法通过解决多商品流问题来实现支持路径分裂的链路映射.文献[9]通过构建一跳中继路由节点来为虚拟链路提供高质量的后备路由路径,从而可以处理带有高级别 QoS 需求的虚拟网请求.文献[10]首次利用蚁群算法来解决虚拟网映射问题.文献[11,12]讨论了网络设备出现变更或故障的情况下如何进行容错映射的问题.文献[13,14]对虚拟网的动态重映射问题进行了讨论.

上述虚拟网映射算法的主要设计目标是提高虚拟网请求映射的成功率,但是却并没有对服务的特定需求进行考虑.本文从 RFICN 中构建宏电路的实际需求出发,按照将同类型服务进行汇聚的思想,设计了一种面向服务聚合的虚拟网映射算法(virtual network mapping algorithm for service aggregation,简称 VMSA),并通过实验仿真对该算法进行了验证与分析.

本文第 1 节描述虚拟网映射的模型和原则.第 2 节详细描述 VMSA 算法的原理、实现过程和优化策略.第 3 节通过仿真验证 VMSA 算法的映射效果.第 4 节为结束语.

1 问题描述

1.1 虚拟网映射模型

虚拟网映射问题可由以下模型描述:

- 底层网络

底层网络的拓扑可以用带权无向图 $G^s = (N^s, L^s, C_N^s, C_L^s)$ 表示,其中: N^s 和 L^s 是底层网络的节点集合和链路集合; C_N^s 和 C_L^s 分别为底层网络的节点和链路所能提供的最大传输处理能力,比如交换能力、计算能力、链路带宽、QoS 能力等.

- 虚拟网请求

一个虚拟网请求包括虚拟网拓扑 G^v 、承载的服务类型 s 、请求到达时间 t_a 和请求持续时间 t_d .虚拟网拓扑可以用带权无向图 $G^v = (N^v, L^v, R_N^v, R_L^v)$ 表示,其中: N^v 为虚拟节点的集合, L^v 为虚拟链路的集合, R_N^v 和 R_L^v 分别表示虚拟节点和虚拟链路的资源约束.虚拟网请求在 t_a 时刻到达后,底层网络为其分配满足 R_N^v 和 R_L^v 约束的网络资源.虚拟网在运行 t_d 时刻后,其所占用的资源将被底层网络回收.如果虚拟网请求的服务类型 s 不同,则该请求对传输质量的需求也不一样.

- 虚拟网映射

虚拟网的映射问题可以描述为虚拟网请求拓扑 G^v 到底层网络拓扑 G^s 的一个满足 R_N^v 和 R_L^v 约束的映射:

$$M:G^v \mapsto (N^v, L^v, R_N^v, R_L^v),$$

其中, $N^v \subset N^s$ 且 $L^v \subset L^s, R_N^v$ 和 R_L^v 是底层网络为虚拟网请求分配的节点和链路资源.虚拟网映射可以分解为节点映射和链路映射:

节点映射: $M^N : (N^v, R_N^v) \mapsto (N^s, R_N^s)$;

链路映射: $M^L : (L^v, R_L^v) \mapsto (L^s, R_L^s)$.

图 1 给出了一个虚拟网映射实例.假如,在虚拟网映射的过程中无法在底层网络里为某一个虚拟节点或虚拟链路找到拥有足够剩余资源的映射目标,则代表虚拟网映射失败,该虚拟网请求将被拒绝.

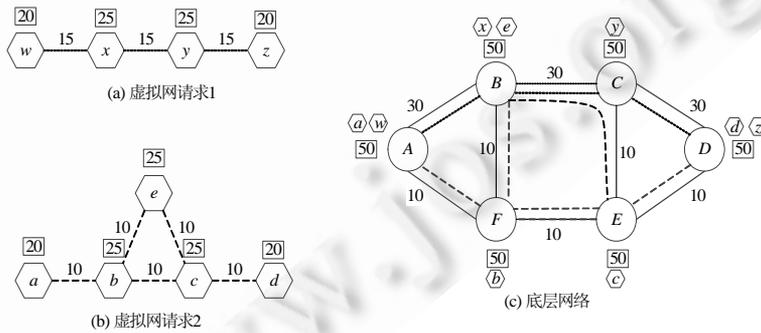


Fig.1 Example of virtual network mapping

图 1 虚拟网映射实例

1.2 虚拟网映射原则

虚拟网映射的首要目标是尽可能满足虚拟网请求中链路和节点的资源需求.在此基础上,传统的虚拟网映射算法一般是按照负载均衡和最短路径优先两个原则来构建虚拟网.

- 负载均衡原则

在虚拟网映射时,要尽可能地将多个虚拟网请求映射到不同的底层资源上,避免出现底层网络的部分节点

和链路负载过高而其他节点却利用率较低的现象.这种负载不均衡现象不但会引起链路拥塞或节点过载,导致网络性能下降,同时还会提高虚拟网请求被拒绝的概率;

- 最短路径优先原则

虚拟网请求中,一条虚拟链路有可能映射到多条底层链路上,若虚拟链路占用的底层链路越多,则其消耗的网络资源也越多,同时还会降低该虚拟链路的传输质量.所以,在映射时应尽量选择最短路径,减少虚拟链路占用的底层链路数量.

RFICN 的设计理念是在链路层面支持对特定服务的传输质量优化,因此,为了进一步提升优化效果,有必要对虚拟网映射算法提出新的映射原则:

- 聚合映射原则

聚合映射原则包含两点含义:一是集中映射传输需求相似的虚拟网请求,二是区分映射传输需求相异的虚拟网请求.

RFICN 通过建立宏电路来实现针对特定服务的传输质量优化.对于某一条底层链路来说,只有在某一类服务的流量远大于该链路上其他类型服务的流量时,才会产生宏电路的构建需求;而且占据主导地位的服务流量在链路中占有的比例越大,宏电路对传输质量的优化效果也越明显.因此,为了提高宏电路的生成概率和优化效果,实现区分服务的设计理念,在映射时应尽量使传输需求相似的虚拟网请求映射到同一组链路和节点上.

由于传输需求相似的虚拟网请求将被尽量映射到同一组链路和节点上,所以为避免链路拥塞和节点过载,这些链路和节点应尽可能少地去承载其他类型的虚拟网请求,从而为映射同类型的服务预留出更多资源.因此,出于负载均衡和提升传输质量的综合考虑,应将传输需求相差较大的虚拟网请求映射到不同的节点和链路上.

2 面向服务聚合的虚拟网映射算法

与传统的虚拟网映射算法相比,面向服务聚合的虚拟网映射算法 VMSA 最主要的区别是:在为虚拟节点和链路选取映射目标时,将会把同类型服务在映射目标中占有的比例纳入其评价标准,使该虚拟网请求能够优先映射在承载较多同类型服务的节点和链路上.因此,在介绍映射算法之前,首先给出节点映射和链路映射的评价标准.

2.1 节点评价标准

在为虚拟节点 n' 选取映射目标时,需要建立一种针对候选底层节点 n^s 的评价标准,该标准可以对 n^s 是否适合承载 n' 进行评价.首先定义了节点服务承载强度 ω :

$$\omega(n', n^s) = \left[C_N^s(n^s) - \sum_{n_i^v \in N(n^s)} R_N(n_i^v) \right] \cdot \frac{\sum_{n_i^v \in N(n^s), s(n')=s(n_i^v)} R_N(n_i^v) + \varepsilon_1}{\sum_{n_i^v \in N(n^s)} R_N(n_i^v) + \varepsilon_2} \quad (1)$$

其中, $C_N^s(n^s)$ 表示底层节点 n^s 的最大处理能力, $N(n^s)$ 表示映射在 n^s 上的虚拟节点集合, $R_N(n_i^v)$ 为虚拟节点 n_i^v 的处理能力需求, $s(n')$ 表示虚拟节点 n' 所属虚拟网请求的服务类型. ε_1 和 ε_2 为两个相近的极小正数,且 $\varepsilon_1 < \varepsilon_2 \ll 1$, ε_2 用来避免 n^s 是空闲节点时(n^s 没有承载任何虚拟网请求)除数为 0 的情况, ε_1 则是用来避免空闲节点的 ω 为 0 的情况, ε_1 和 ε_2 的比值决定了对空闲节点的评价值.

可以看出, ω 不仅仅考虑了底层节点的剩余处理能力,还考虑了 n' 所属的服务类型在 n^s 承载的所有服务中占有的比例.

单点链路服务承载强度 φ 描述了所有与 n^s 相连的底层链路 l^s 的服务承载情况和资源剩余情况,从另一个角度评价了 n^s 是否适合承载虚拟节点 n' . φ 的定义如下:

$$\varphi(n', n^s) = \sum_{l^s \in NL(n^s)} \left\{ \left[C_L^s(l^s) - \sum_{l_i^v \in L(l^s)} R_L(l_i^v) \right] \cdot \frac{\sum_{l_i^v \in L(l^s), s(n')=s(l_i^v)} R_L(l_i^v) + \varepsilon_1}{\sum_{l_i^v \in L(l^s)} R_L(l_i^v) + \varepsilon_2} \right\} \quad (2)$$

其中, $NL(n^s)$ 表示与 n^s 相连的底层链路集合, $C_L^s(l^s)$ 表示底层链路 l^s 的最大传输能力, $L(l^s)$ 表示映射在 l^s 上的虚拟链路集合, $R_L(l_i^s)$ 为虚拟链路 l_i^s 的传输能力需求, $s(l)$ 表示虚拟链路 l 所属虚拟网请求的服务类型. ε_1 和 ε_2 为两个相近的极小正数, 且 $\varepsilon_1 < \varepsilon_2 \ll 1$, ε_2 用来避免 l^s 是空闲链路时 (l^s 没有承载任何虚拟网请求) 除数为 0 的情况, ε_1 则是用来避免空闲链路的 θ 值为 0 的情况, ε_1 和 ε_2 的比值决定了对空闲链路的评价值.

最后, 通过定义节点适应度 η 来评价 n^s 是否适合承载虚拟节点 n^s :

$$\eta(n^s, n^s) = \alpha(n^s, n^s) + \alpha \cdot \varphi(n^s, n^s) \quad (3)$$

其中, α 为权重因子.

由公式(3)可以看出, η 从底层节点的剩余资源、节点承载的服务类型、与该节点相连的底层链路的剩余资源以及链路承载的服务类型这 4 个方面对底层节点 n^s 做出综合评价. η 值较大, 说明 n^s 不但拥有较多的剩余资源, 而且表明了 n^s 和其周边的链路承载了较多与虚拟节点 n^s 同类型的服务.

2.2 链路评价标准

在链路映射时, 一条虚拟链路可能映射到多条相互连接的底层链路上, 虚拟链路占用的底层链路个数称为该虚拟链路的长度. 虚拟链路的长度越长, 其消耗的底层链路带宽资源就越多, 所以从资源利用率的角度上来看, 应采用最短路径优先原则. 但根据聚合映射原则, 最短路径不一定是最优路径. 因此在路径选择时, 要兼顾虚拟链路长度和底层链路上的服务承载情况.

本文通过定义服务路径长度 θ 来对虚拟链路 l^s 的底层传输路径 p 进行评价, θ 值越小, 代表 p 的评价越好.

$$\theta(l^s, p) = \sum_{l^s \in p} \left[1 - \frac{\sum_{l_i^s \in L(l^s), s(l^s)=s(l_i^s)} R_L(l_i^s) + \varepsilon_1}{\sum_{l_i^s \in L(l^s)} R_L(l_i^s) + \varepsilon_2} \right] \quad (4)$$

其中, p 是承载 l^s 的底层链路集合, 代表了 l^s 在底层网络的实际传输路径; $L(l^s)$ 表示映射在 l^s 上的虚拟链路集合; $R_L(l_i^s)$ 为虚拟链路 l_i^s 的传输能力需求; $s(l)$ 表示虚拟链路 l 所属的虚拟网请求的服务类型. ε_1 和 ε_2 为两个相近的极小正数, 且 $\varepsilon_1 < \varepsilon_2 \ll 1$, ε_2 用来避免 l^s 是空闲链路时除数为 0 的情况, ε_1 和 ε_2 的比值决定了对空闲链路的评价值.

2.3 映射算法

VMSA 映射算法的设计思想是: 当 InP 收到一个虚拟网请求 R 以后, 首先从 R 的网络拓扑 G^v 中选择一个连接度数最高的虚拟节点 n_{1st} , 将其映射到 η 值最高的底层节点上; 然后, 以 n_{1st} 为起点, 以广度优先遍历的次序对 G^v 中的其他虚拟节点进行映射, 每映射一个虚拟节点的邻接节点后还要将两个节点之间的虚拟链路进行映射.

假设在某一个虚拟节点 n_i^v 被映射到底层节点 n_i^s 以后, 下一个将要进行映射的对象是通过虚拟链路 l 与 n_i^v 相连的虚拟节点 n_j^v , 映射的方法是: 将所有未被 R 映射过的底层节点设为候选集 C , 对于 C 中的每一个节点 n_j^s , 首先计算 $\eta(n_i^v, n_j^s)$ 的值; 然后利用 K 短路算法^[15] 求出前 k 条连结 n_i^s 和 n_j^s 的最短路径, 并分别计算每条路径的 θ 值, 选取 θ 值最小的一条路径记为 $p_{\min}(n_i^s, n_j^s)$. 接下来就可以计算 n_j^s 的最短映射适应度 σ :

$$\sigma(n_i^v, n_j^s) = \frac{\eta(n_i^v, n_j^s)}{\theta(l^s, p_{\min})} \quad (5)$$

最后, 将 C 中的底层节点按 σ 值的大小排序, σ 值最大的底层节点 n_{\max} 就可以作为 n_j^v 的映射目标节点, 而 $p_{\min}(n_i^s, n_{\max})$ 可以作为 l 的映射目标路径.

算法 1. VMSA(virtual network mapping algorithm for service aggregation)

输入: 虚拟网请求拓扑 G^v , 底层网络拓扑图 G^s , 参数: $k, \varepsilon_1, \varepsilon_2, \alpha$;

输出: 映射结果 M .

- 1 从 G^v 中选取度数最大的节点作为第 1 个待映射节点 n_{1st} ;
- 2 将 n_{1st} 映射到 G^s 中 η 值最高的底层节点, 并标记 n_{1st} 已被映射;

```

3  初始化队列  $q$ , 并将  $n_{1st}$  入队;
4  while  $q$  不为空 do
5    对  $q$  进行出队操作, 获取队头的虚拟节点  $n$ ;
6    获取连接到  $n$  的第 1 条虚拟链路  $l$ , 同时取  $l$  另一端的虚拟节点  $n'$ ;
7    while  $l \neq null$  do
8      if  $n'$  已被映射 then
9        if  $l$  未被映射 then
10         利用  $K$  短路径算法求出前  $k$  条从  $n$  到  $n'$  的最短路径;
11         计算每条路径的  $\theta$  值, 将  $l$  映射  $\theta$  值最小的一条路径上;
12       end if
13     else
14        $C = \{n^s | n^s \in G^s \text{ and } n^s \text{ 未被映射 and } n^s \text{ 的剩余资源足够承载 } n'\}$ ;
15       for all  $n^s \in C$  do
16         计算  $\eta(n', n^s)$ ;
17         利用  $K$  短路径算法求出前  $k$  条从  $n$  的底层节点到  $n^s$  最短路径;
18         计算每条路径的  $\theta$  值, 取  $\theta$  值最小的一条路径, 记为  $p_{\min}(n, n^s)$ ;
19         计算  $\sigma(n', n^s)$ ;
20       end for
21       将  $n'$  映射到  $\sigma$  值最大的  $n^s$  上, 并将  $l$  映射到  $p_{\min}(n, n^s)$  上;
22       将  $n'$  标记为已映射并将其插入到队列  $q$  的队尾;
23     end if
24     将  $l$  设为与  $n$  相连的下一条虚拟链路, 将  $n'$  设为  $l$  另一端的虚拟节点;
25   end while
26 end while

```

若虚拟网管理模式为分布式管理(在一个域内同时存在多个负责处理虚拟网请求的管理节点), 可能出现多个虚拟网请求同时被映射的情况, 这就导致某个管理节点可能受到其他节点的映射行为的影响, 从而使其维护的底层资源状态信息与实际的资源状态不一致. 为了避免这种由并行映射中的冲突现象, VMSA 在为待映射的虚拟节点选择最优映射目标时(算法的第 21 步), 同时还要记录次优映射目标. 当映射命令到达最优目标节点后, 一旦发现该节点的已经被同时期映射的其他虚拟网占用, 从而导致其无法接受新的映射时, 则立即将映射命令转发给次优目标节点, 将待映射的虚拟节点映射在次优目标节点上.

2.4 算法优化策略

从算法 1 可以看出: 为了映射虚拟节点 n' , 在算法的第 14 步中, 底层网络中所有未被当前虚拟网请求映射的底层节点都被选入到 n' 的映射候选集 C 中, 这就使得底层网络规模比较庞大的情况下, 时间消耗较高的 K 短路径算法会被大量调用. 因此, 为了降低 VMSA 算法的时间消耗, 有必要对候选集 C 进行缩减.

本文提出了一种基于跳数约束的候选节点选取策略(node selection strategy with hop constraint, 简称 NSH), 该策略在选取候选节点时, 会将选取的范围限定在距离某个底层节点 h 跳以内且符合映射条件的所有底层节点, 使得候选集 C 的定义变为

$$C = \{n^s | n^s \in G^s, n^s \notin N', Dist(n^s, n_1^s) \leq h\} \quad (6)$$

其中, N' 为已被 R 映射的底层节点集合, n_1^s 为 VMSA 算法的步骤 5 中承载虚拟节点 n 的底层节点, 函数 $Dist$ 用来计算两个底层节点之间的最短路径的跳数.

NSH 的实现算法采用基于递归的深度优先遍历思想, 并将遍历的深度限制在 n_1^s 的 h 跳以内. 初始调用时,

跳数 hop 设为 0.

算法 2. NSH(node selection algorithm with hop constraint).

输入:底层网络拓扑图 G^s ,虚拟节点 n^v ,最大跳数 h ,底层节点 n^s ,跳数 hop ;

输出:映射目标候选集 C .

```

1  if  $hop \neq 0$  and  $n^s$  未被映射且剩余资源足够承载  $n^v$  then
2     $C = n^s \cup C$ ;
3  end if
4  for all  $n' \in \{n^s \text{ 的所有邻接节点} \}$  do
5    if  $n' \notin C$ 
6       $hop = hop + 1$ ;
7      if  $hop \leq h$  then
8         $NSH(G^s, n^v, h, n', hop)$ ;
9      end if
10   end if
11 end for

```

在映射某个虚拟节点的过程中,假设底层链路的总数量为 e ,通过调用 NSH,将候选集 C 中节点的数量由 v 减少为 i ,则 NSH 的时间复杂度为 $O(i+e)$,K 短路算法的时间复杂度为 $O(e+v \log v+k)$.而通过调用一次 NSH 就可以减少 $v-i$ 次 K 短路算法的调用.由此可见,通过采用 NSH,能够有效地降低 VMSA 的运行时间.

3 实验仿真

3.1 实验环境

本实验在 Pentium 4 CPU 3.2GHz,1G 内存的 PC 机上运行.底层网络拓扑和虚拟网请求拓扑由 GT-ITM^[16] 工具生成,底层网络共包括共包含 100 个节点和 570 条链路,底层节点的计算资源和底层链路的带宽资源取值在 [50,100] 内均匀分布.虚拟网请求共计 2 000 个,每个请求的节点个数在 [2,10] 内均匀分布,节点连接概率为 0.5,虚拟节点的资源取值在 [0,20],虚拟链路的带宽取值在 [0,25] 之间均匀分布,请求的到达时间服从平均 100 个单位时间到达 5 个请求的泊松过程,持续时间服从 μ 为 1 000 的指数分布,所有虚拟网请求按照 DiffServ 标准^[17] 被随机划分为 4 类服务,即:控制类服务、多媒体类服务、数据传输类服务、尽力而为类服务.

实验将选取两种比较典型的启发式映射算法作为 VMSA 算法的比较对象,即:基于负载均衡的算法 VNA^[6] 和基于贪心映射的算法 SP^[8],VMSA 分为采用 NSH 策略和未采用 NSH 策略这两种情况.VMSA 算法的参数设置为 $\alpha=0.15, \varepsilon_1=0.09, \varepsilon_2=0.1, k=5$.每映射 100 个虚拟网请求记录一次实验数据.

3.2 NSH 优化效果

首先,通过实验来验证 NSH 对 VMSA 算法在执行时间上的优化效果,并观察在不同的跳数限制下 NSH 的优化效果.NSH 的实现原理是缩减节点映射的候选集 C ,其中,最大跳数 h 越小, C 中的候选节点也就越少.图 2 给出了未采用 NSH 时算法的执行时间数据以及采用 NSH 并且 h 分别为 2,3,4 时的算法执行时间数据.

由实验结果可知: h 的值选取的越小,候选集 C 中的节点数量就越少,运行时间的优化效果就越明显.因此,本实验在测试采用 NSH 策略的 VMSA 算法时,参数 h 的值设为 2.

3.3 请求接收率

当底层网络无法为虚拟网请求分配足够的资源时,该虚拟网请求将被拒绝,而合理的映射策略能够使底层网络接收更多的虚拟网请求.虚拟网请求的接收率(acceptance ratio,简称 AR)是衡量映射算法的重要指标,其定义是成功映射的虚拟网请求数量 R_s 与请求总数量 R_T 的百分比,即:

$$AR = R_s / R_T \tag{7}$$

AR 越高,表示映射算法对资源的利用就越好.图 3 展示了 VNA,SP,VMSA 这 3 种算法的运行过程中,AR 的值随着请求的不断到来而发生变化的情况.由实验数据可知,VMSA 算法的请求接收率略微优于 SP 和 VNA.

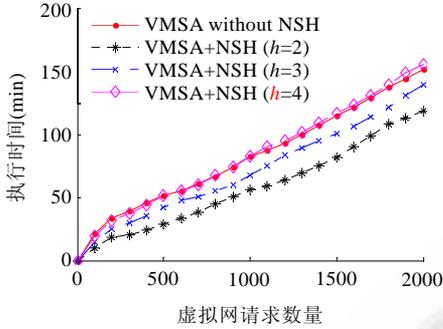


Fig.2 Comparisons of optimizing effect of NSH
图 2 NSH 优化效果比较

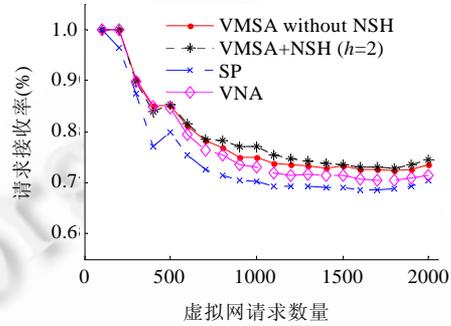


Fig.3 Acceptance ratio
图 3 请求接收率

3.4 服务聚合度

服务聚合度(service aggregation degree,简称 SAD)是用来衡量同类型服务的映射聚合程度.在底层链路和节点上,居于主导地位的服务占有的比例越大,则服务聚合度越高,宏电路的生成概率和优化效果也就越好.服务聚合度可分为节点服务聚合度和链路服务聚合度.

- 节点服务聚合度定义如下:

$$SAD_{node} = \frac{1}{n_{node}} \cdot \sum_{n^s \in N^s} \frac{\sum_{n_i^v \in N(n^s), s(n_{max})=s(n_i^v)} R_N(n_i^v)}{\sum_{n_i^v \in N(n^s)} R_N(n_i^v) + \varepsilon} \tag{8}$$

其中, n_{node} 表示非空闲的底层节点个数; $N(n^s)$ 表示映射在 n^s 上的虚拟节点集合; $R_N(n_i^v)$ 为虚拟节点 n_i^v 的处理能力需求; $S(n_{max})$ 表示底层链路 n_s 中占用计算资源最多的服务的类型; ε 为极小正数,用来避免除数为 0 的情况.

- 链路服务聚合度定义如下:

$$SAD_{link} = \frac{1}{n_{link}} \cdot \sum_{l^s \in L^s} \frac{\sum_{l_i^v \in L(l^s), s(l_{max})=s(l_i^v)} R_L(l_i^v)}{\sum_{l_i^v \in L(l^s)} R_L(l_i^v) + \varepsilon} \tag{9}$$

其中, n_{link} 表示非空闲的底层链路个数; $L(l^s)$ 表示映射在 l^s 上的虚拟节点集合; $R_L(l_i^v)$ 为虚拟节点 l_i^v 的处理能力需求; $S(l_{max})$ 表示底层链路 l_s 中占用带宽最多的服务类型; ε 为极小正数,用来避免除数为 0 的情况.

由图 4 可见:在承载的虚拟网请求数量较少时,会优先选择空闲节点和空闲链路作为映射的目标;随着请求的不断到来,只承载单一类型服务的节点和链路的数量将会逐渐减少,各种算法的 SAD 值也会出现不同程度的下降.在部分虚拟网请求由于生存期结束而释放资源以后,整个底层网络的资源占用率也将达到稳定,而 SAD 的值也将趋于稳定.由于 VMSA 算法采用了聚合映射相同类型服务的思想,使其在节点服务聚合度和链路服务聚合度上都远远优于 SP 和 VNA.

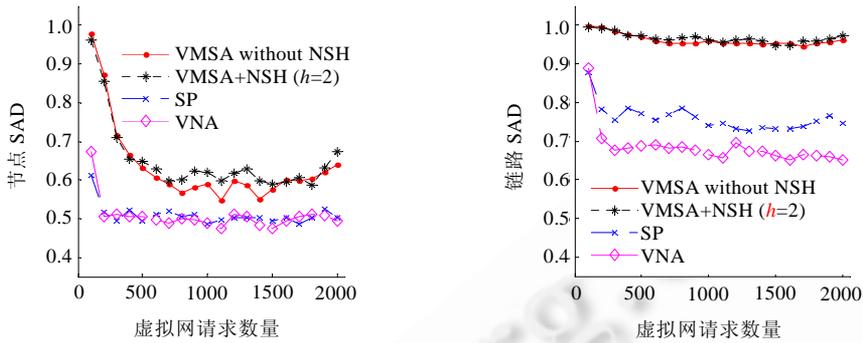


Fig.4 Service aggregation degree

图 4 服务聚合度

3.5 负载均衡度

VMSA 算法在进行节点和链路的映射时,不但考虑了底层节点和链路的服务承载情况,也考虑了其剩余资源的情况,在实现同类型服务集中映射的同时,也最大程度上实现了负载均衡.底层网络负载均衡的程度通过负载均衡度(load balance degree,简称LBD)来评价,LBD包括节点负载均衡度和链路负载均衡度,其定义分别如下:

$$LBD_{node} = \frac{1}{N_{node}} \cdot \sum_{n_i \in N^s} |Load(n_i^s) - Load_{arg}(N^s)| \tag{10}$$

$$LBD_{link} = \frac{1}{N_{link}} \cdot \sum_{l_i \in L^s} |Load(l_i^s) - Load_{arg}(L^s)| \tag{11}$$

其中, N 为底层网络中节点或链路的数量, $Load(n^s)$ 和 $Load(l^s)$ 分别表示单个底层节点或底层链路上的负载, $Load_{arg}(N^s)$ 和 $Load_{arg}(L^s)$ 分别表示底层网络中底层节点的平均负载和底层链路的平均负载.由此可以给出 LBD 的定义:

$$LBD = \beta \cdot LBD_{node} + (1 - \beta) \cdot LBD_{link} \tag{12}$$

其中, β 为权重因子,本实验中取 0.5.

由图 5 可以看出,VMSA 算法的负载均衡度与 SP 算法和 VNA 算法处于同一水平.

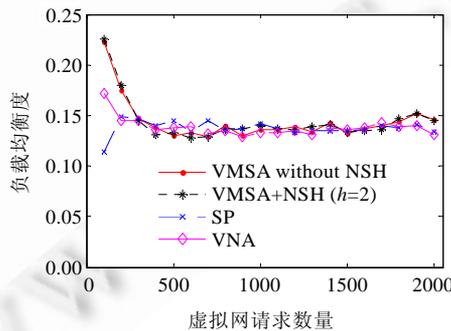


Fig.5 Load balance degree

图 5 负载均衡度

3.6 平均虚拟链路长度

在虚拟网的链路映射过程中,一条虚拟链路可能映射到多条相连的底层链路上.根据最短路径优先原则,一条虚拟链路占用的底层链路的数量越少,其占用的网络带宽资源也就越少.因此,本实验将对所有被成功映射的虚拟网请求进行统计,计算虚拟链路占用的底层链路个数的平均值,即平均虚拟链路长度(average length of

virtual links,简称 VAL),定义如下:

$$VAL = N_{sublink} / N_{vlink} \quad (13)$$

其中, N_{vlink} 表示所有已被映射的虚拟链路数量, $N_{sublink}$ 表示所有已被映射的虚拟链路占用的底层链路数量之和。

由图 6 可见:未采用 NSH 的 VMSA 算法在平均虚拟链路长度方面略优于 SP 算法,与 VNA 算法处于同一水平;而采用了 NSH 的 VMSA 算法由于在节点映射时加入了跳数限制,使得一条虚拟链路占用的底层链路数量也被限制在一定范围内,因此其平均虚拟链路长度要优于未采用 NSH 的 VMSA,SP 与 VNA。

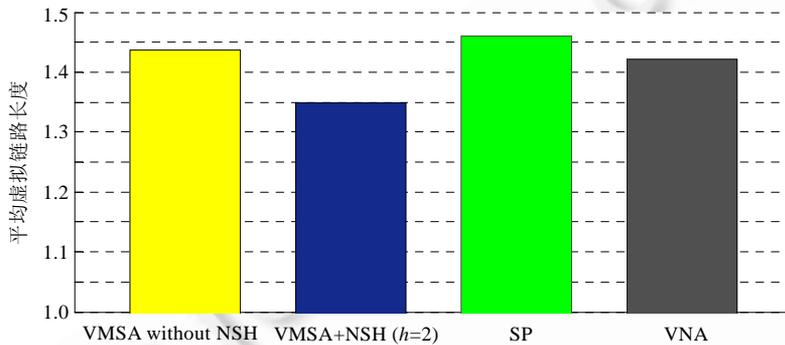


Fig.6 Average length of virtual links

图 6 平均虚拟链路长度

4 结束语

本文根据 RFICN 的实际需要,设计了一种面向服务聚合的虚拟网映射算法,该算法不但在请求接收率、负载均衡以及平均虚拟链路长度等传统的评价标准上有着较好的表现,而且能够将类型相同的服务尽量映射到同一组底层节点和链路上,实现了将服务按照类型进行聚合映射的思想,为通过构建宏电路来提升全网服务质量奠定了基础。

VMSA 算法在减少运行时间方面仍有可改进的空间.此外,在虚拟网的映射过程中,如何从服务聚合的角度对已映射的虚拟网进行动态重映射,是有待进一步研究的问题。

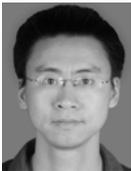
References:

- [1] Cheng X, Zhang Z, Su S, Yang F. Survey of virtual network embedding problem. *Journal on Communications*, 2011,32(10): 143–151 (in Chinese with English abstract).
- [2] Carapinha J, Jiménez J. Network virtualization—A view from the bottom. In: *Proc. of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*. 2009. 73–80. [doi: 10.1145/1592648.1592660]
- [3] Chowdhury M, Boutaba R. A survey of network virtualization. *Computer Networks*, 2010,54(5):862–876. [doi: 10.1016/j.comnet.2009.10.017]
- [4] Cai Z, Liu Q, Lu P, Xiao N, Wang Z. Virtual network mapping model and optimization algorithms. *Ruan Jian Xue Bao/Journal of Software*, 2012,23(4):864–877 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4063.htm> [doi: 10.3724/SP.J.1001.2012.04063]
- [5] Chowdhury M, Rahman MR, Boutaba R. ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping. *IEEE Trans. on Networking*, 2012,20(1):206–219. [doi: 10.1109/TNET.2011.2159308]
- [6] Zhu Y, Ammar M. Algorithms for assigning substrate network resources to virtual network components. In: *Proc. of the IEEE INFOCOM 2006*. 2006. 23–29. [doi: 10.1109/INFOCOM.2006.322]
- [7] Qi N, Wang B, Wang B. Research on balanced construction algorithm of virtual network. *Journal of Electronics & Information Technology*, 2011,33(6):1301–1306 (in Chinese with English abstract). [doi: 10.3724/SP.J.1146.2010.01130]

- [8] Yu M, Yi Y, REXFORD J. Rethinking virtual network embedding: Substrate support for path splitting and migration. ACM SIGCOMM Computer Communication Review, 2008,38(2):17–29. [doi: 10.1145/1355734.1355737]
- [9] Shamsi J, Brockmeyer M. QoSMap: QoS aware mapping of virtual networks for resiliency and efficiency. In: Proc. of the IEEE GLOBECOM Workshop. 2007. 1–6.
- [10] Fajjari I, Aitsaadi N, Pujolle G, Zimmermann H. VNE-AC: Virtual network embedding algorithm based on ant colony metaheuristic. In: Proc. of the ICC 2011. 2011. 1–6. [doi: 10.1109/icc.2011.5963442]
- [11] Rahman MR, Aib I, Boutaba R. Survivable virtual network embedding. In: Proc. of the IFIP Networking 2010. LNCS 6091, Berlin, Heidelberg: Springer-Verlag, 2010. 40–52. [doi: 10.1007/978-3-642-12963-6_4]
- [12] Yu HF, Qiao CM, Anand V, Liu X, Di H, Sun G. Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures. In: Proc. of the IEEE GLOBECOM 2010. 2010. 1–6. [doi: 10.1109/GLOCOM.2010.5683951]
- [13] Cai Z, Liu F, Xiao N, Liu Q, Wang Z. Virtual network embedding for evolving networks. In: Proc. of the IEEE GLOBECOM 2010. 2010. 1–5. [doi: 10.1109/GLOCOM.2010.5683160]
- [14] Fajjari I, Aitsaadi N, Pujolle G, Zimmermann H. VNR algorithm: A greedy approach for virtual networks reconfigurations. In: Proc. of the IEEE GLOBECOM 2011. 2011. 1–5. [doi: 10.1109/GLOCOM.2011.6134006]
- [15] Eppatein D. Finding the k shortest paths. SIAM Journal on Computing, 1998,28(2):652–673. [doi: 10.1137/S0097539795290477]
- [16] Zegura E, Calvert K, Bhattacharjee S. How to model an Internetwork. In: Proc. of the IEEE INFOCOM. 1996. 594–602. [doi: 10.1109/INFOCOM.1996.493353]
- [17] Chan K, Baker F, Babiarz J. Configuration guidelines for diffserv service classes. Internet Engineer TaskForce, RFC 4594, 2006. <http://rfc.arog.net/rfc4594.htm>

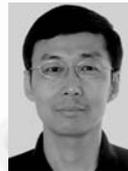
附中中文参考文献:

- [1] 程祥,张忠宝,苏森,杨放春.虚拟网络映射问题研究综述.通信学报,2011,32(10):143–151.
- [4] 蔡志平,刘强,吕品,肖依,王志英.虚拟网络映射模型及其优化算法.软件学报,2012,23(4):864–877. <http://www.jos.org.cn/1000-9825/4063.htm> [doi: 10.3724/SP.J.1001.2012.04063]
- [7] 齐宁,王保进,汪斌强.均衡虚拟网构建算法研究.电子与信息学报,2011,33(6):1301–1306.



江逸茗(1984—),男,江苏南通人,博士生,主要研究领域为网络虚拟化,网络架构.

E-mail: j8403@163.com



程东年(1957—),男,博士,教授,主要研究领域为网络体系结构.

E-mail: cdn@ndsc.com.cn



兰巨龙(1962—),男,博士,教授,博士生导师,主要研究领域为宽带信息网络.

E-mail: ndscljl@163.com



王志明(1986—),男,博士生,主要研究领域为网络虚拟化.

E-mail: wangzm05@gmail.com