

# 基于拓扑感知的可重构服务承载网动态重构算法

梁宁宁, 兰巨龙, 张震

(国家数字交换系统工程技术研究中心, 河南 郑州, 450002)

**摘要:** 可重构信息通信基础网络通过构建服务承载网的方式为业务提供自适应的承载服务。针对高效利用有限底层资源的问题, 提出一种基于资源关键度进行动态映射的服务承载网构建算法。算法将通过节点或链路的最短路径数作为资源关键度的衡量指标, 区别对待底层资源; 并实时动态感知关键资源的使用状况, 依据不同业务需求对服务承载网进行自适应调整。仿真结果表明, 算法在构建成功率、收益花费比和资源均衡度等方面均具有良好性能。

**关键词:** 拓扑感知; 关键资源; 动态映射; 可重构服务承载网

中图分类号: TP393

文献标识码: A

## Dynamic topology awareness-based reconfigurable service carrying network reconfiguration

LIANG Ning-ning, LAN Ju-long, ZHANG Zhen

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

**Abstract:** Reconfigurable information communication basal network supports self-adaptive services to applications by constructing reconfigurable service carrying network(RSCN). To effectively utilize the limited substrate network resources, an algorithm of dynamic topology awareness-based RSCN reconfiguration (DTAR) was proposed. The algorithm uses the number of shortest paths as resource critical degree which across the node or link to distinguish substrate resource. And it also dynamically awares the states of critical resources, reoptimises the RSCN according to service request. Experimental results show that comparing with the existing algorithms, the proposed algorithm achieves higher success ratio, gains higher revenue, cost ratio and load equilibrium for substrate network.

**Key words:** topology awareness, critical resource, dynamic embedding, reconfigurable service carrying network

### 1 引言

随着当前网络应用规模的不断扩张、新兴业务不断涌现, 现有的以 IP 为核心的网络基础架构已经不堪重负, 带来了网络结构僵化、核心功能单一、可控性和演变能力低下等问题。加之当前网络的刚性结构, 而现有措施大多是对其进行修补或是简单扩展, 并未从根本上满足泛在互联、融合异构、可信可管可扩等需求<sup>[1]</sup>。

为此, 可重构信息通信基础网络通过为用户构建可重构服务承载网的方式, 实现其对功能可动态重构和扩展的底层物理网络的共享, 从而为不同业务提供其根本需求和可定制的基础网络服务。所谓“可重构服务承载网 (RSCN, reconfigurable service carrying network)”即是将业务需求与底层网络资源状态, 如网络拓扑、资源状态等条件优化考虑, 构建出的多个在底层物理资源上存在交集, 能够提供不同服务能力的承载子网, 如图 1 所示。

收稿日期: 2015-01-03; 修回日期: 2015-10-26

基金项目: 国家重点基础研究发展计划 (“973”计划) 基金资助项目 (No.2012CB315901, No.2013CB329104); 国家自然科学基金资助项目 (No.61372121); 国家高技术研究发展计划 (“863”计划) 基金资助项目 (No.2013AA013505)

**Foundation Items:** The National Basic Research Program of China (973 Program) (No.2012CB315901, No.2013CB329104), The National Natural Science Foundation of China (No.61372121), The National High Technology Research and Development Program of China (863 Program) (No.2013AA013505)

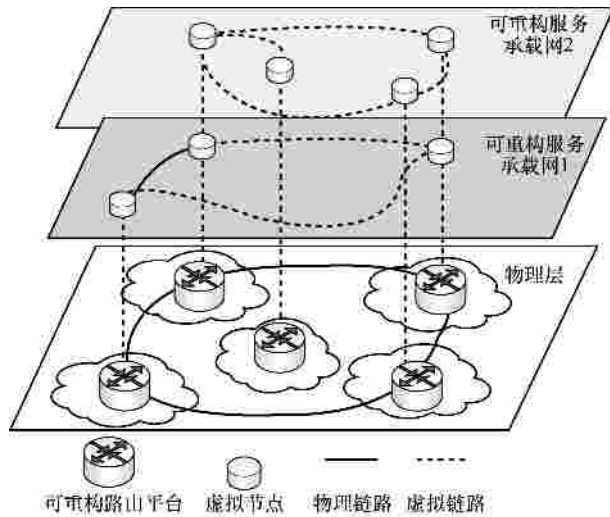


图 1 可重构服务承载网示意

构建服务承载网即是在满足用户业务需求的基础上，根据网络的动态变化优化网络资源配置，充分利用网络资源。然而，在服务承载网构建中主要存在 2 个问题：1) 不关心底层拓扑，没有区分底层节点和链路的重要性；2) 缺少对已映射的服务承载网构建请求的动态优化措施。基于此，本文提出了一种基于拓扑感知的可重构服务承载网动态重构 (DTAR, dynamic topology awareness-based rscn reconfiguration) 算法。算法定义“资源关键度”、“资源紧要度”作为衡量底层资源重要性的指标，在服务承载网构建时，将底层资源区别对待；并动态感知资源的使用状态，发现紧要资源并进行相应迁移处理，从而大大提高了服务承载网请求的构建成功率。

## 2 相关研究

面对用户数量和业务种类爆炸式地增长，可重构服务承载网的构建目标愈来愈明确，即在满足用户业务需求的基础上，尽可能多地接受服务承载网构建请求，最大化利用有限的底层资源。以此为出发点，如何高效地利用底层资源进行服务承载网构建，是服务承载网构建中的核心问题。现有评估底层资源的方式主要包括两方面。

### 1) 底层节点和链路资源

文献[2]提出了一种节点映射与链路映射相分离的两阶段承载网映射算法。首先，算法以贪婪方式进行节点映射，将所有虚拟节点映射到具有更充足资源的底层节点，而后在链路映射阶段分别使用了最短路径和多商品流算法进行映射。文献[3]使用

混合整数规划方法，提出节点与链路协调映射的承载网构建算法。文献[4]提出了同时映射虚拟节点和虚拟链路的分布式算法，然而在性能和最优化方面都与集中式算法存在差距，且算法并未考虑构建请求的生存时间。文献[5]提出了周期性地检测底层资源负载情况，并据此对超负荷资源上的承载网进行重映射的算法。然而，这类算法在映射过程中仅考虑了节点 CPU 和链路带宽的约束，忽略了节点和链路本身对映射造成的影响，片面地刻画了底层资源的属性，最终将导致底层资源利用率的降低。

### 2) 拓扑属性

拓扑属性作为网络资源重要参数之一，在现有承载网构建中备受关注。文献[6]提出了一种一阶段回溯算法子图同构检测的动态映射方式。文献[7]提出了一种能够适应网络环境变化的承载网拓扑控制算法。该算法基于生物系统中的吸引子模型，根据节点故障所引起的网络环境变化，动态地重新配置吸引子结构，从而实现承载网拓扑的重建。但算法的复杂度较高，不易实现。利用小拓扑的灵活性，文献[8]将每个虚拟网分裂成多个星型子网，并将星型子网中具有最高度的中心节点映射到具有最低 CPU 和可用带宽的底层节点。一旦确定中心节点，其他具有较高度的节点将一个接一个地映射到具有较高资源占用率且距中心节点较近的底层节点之上。然而，算法假设底层网络资源无限，且并不适用于许多特殊拓扑。文献[9]提出了一种基于流量约束的虚拟网络映射算法，然而，该算法也仅适用于拓扑结构为星型的构建请求。文献[10]将 Google PageRank 算法引入 Web 搜索域中，提出了一种拓扑感知的底层资源度量方法，通过马尔可夫随机游动模型，基于当前资源及其拓扑属性对节点排序，并依据其排序值进行贪婪映射。文献[11]以资源连接度度量底层资源，这一参数的引入虽然凸显了底层资源的差异性，但该指标并不能准确定义一个节点或是链路的重要性。例如，虽然一个节点的连接度很高，但与其相连的节点并不重要，那么该节点也并不一定重要；反之，若一个节点的连接度不高，但与它相连的节点大多非常重要，那么该节点在网络中也可能非常重要。虽然，这类算法已或多或少的考虑了网络拓扑对承载网映射的影响，但仍然存在 2 方面的问题：1) 没有使用较为准确、合理的方式区分底层节点和链路的重要性；2) 缺少对已映射的

服务承载网构建请求的动态优化机制。

本文认为,不同的节点和链路在底层拓扑中的重要性大相径庭,放弃使用非关键底层资源,而将业务需求过量映射到关键的底层节点和链路上,易造成资源瓶颈,引起底层网络失效,从而严重影响服务承载网的构建成功率。因此,综合地度量底层资源属性,动态感知底层资源状态,对于高效构建可重构服务承载网尤为重要。

基于此,本文提出了一种能够将底层资源合理区别对待的服务承载网动态构建算法,该方法具有以下特点:1)将通过底层节点和链路的最短路径数作为衡量关键资源的指标,能够较全面地反映该资源在全网中的重要性;2)依据资源关键度对底层节点排序,将虚拟节点优先映射到满足业务需求的非关键底层资源之上,合理高效地使用底层网络资源;3)通过对资源使用程度的动态感知,发现紧要资源并进行相应地迁移处理,有效避免了瓶颈资源的出现,增强了基础物理网络的可靠性。

### 3 可重构服务承载网构建模型

#### 3.1 基础物理网络

为了方便数学建模,将基础物理网络抽象为无向图  $G^s = (N^s, L^s, C^s)$ ,其中,  $N^s$  和  $L^s$  分别表示基础物理网络中的节点集合和链路集合,  $C^s$  表示基础物理网络能够提供的能力,可分解为节点资源和链路资源。节点资源包括节点内存容量  $C_{\text{bm}}^s$  和 CPU 能力  $CP_{\text{cpu}}^s$ ; 链路资源指链路带宽  $Bw^s$ 。

#### 3.2 业务需求模型

RSCN 业务需求模型表示为无向图  $G^r = (N^r, L^r, R^r)$ ,其中,  $N^r$  和  $L^r$  分别表示 RSCN 中的虚拟节点集合和虚拟链路集合,且  $N^r \subset N^s$ ,  $L^r \subset L^s$ ,  $R^r$  表示 RSCN 构建请求中,对虚拟节点和虚拟链路的限制条件,包括虚拟节点的计算能力需求  $CP_{\text{cpu}}^r$  和存储资源需求  $C_{\text{bm}}^r$ , 虚拟链路的带宽资源需求  $Bw^r$  以及时延需求  $D^r$ 。

#### 3.3 服务承载网映射

RSCN 构建问题可以表述为满足无向图  $G^r$  中约束条件  $R^r$  的  $G^r$  到  $G^s$  子集的映射,表示如下

$$M: G^r \rightarrow G^s, G^s = (N^s, L^s, C^s) \quad (1)$$

其中,  $N^v$  和  $L^v$  分别是  $N^s$  和  $L^s$  的子集,  $G^v$  表示构建 RSCN 的基础物理网络的服务能力。RSCN 构建

可以分割成节点映射和链路映射。

$$\text{节点映射 } M^N: (N^r, R_N^r) \rightarrow (N^v, C_N^v) \quad (2)$$

$$\text{链路映射 } M^L: (L^r, R_L^r) \rightarrow (L^v, C_L^v) \quad (3)$$

其中,  $R_N^r$  和  $R_L^r$  分别表示 RSCN 构建中对虚拟节点和虚拟链路的约束条件,  $C_N^v$  和  $C_L^v$  分别表示构建 RSCN 的基础物理节点和链路所能提供的服务承载能力。

#### 3.4 服务承载网目标

##### 1) 高效利用底层资源

服务承载网是在共享的底层资源上构建出能够提供不同服务能力的承载子网,其核心问题即是如何在有限的底层资源之上,构建尽可能多的服务承载网,高效地使用底层网络资源。

定义 1 收益  $R(G^r)$ 。成功为业务构建服务承载网所带来收益的总和。

$$R(G^r) = \sum_{n^r \in N^r} CP_{\text{cpu}}^r(n^r) + \sum_{n^r \in N^r} C_{\text{bm}}^r(n^r) + \sum_{l^r \in L^r} Bw^r(l^r) \quad (4)$$

其中,  $CP_{\text{cpu}}^r(n^r)$  和  $C_{\text{bm}}^r(n^r)$  分别表示成功构建服务承载网的业务对底层节点 CPU 计算能力和内存容量的需求;  $Bw^r(l^r)$  表示业务对链路带宽的需求。

定义 2 花费  $C(G^s)$ 。为成功构建服务承载网所消耗的底层资源的总和。

$$C(G^s) = \sum_{n^s \in N^s} CP_{\text{cpu}}^s(n^s) + \sum_{n^s \in N^s} C_{\text{bm}}^s(n^s) + \sum_{l^s \in L^s} Bw^s(l^s) \quad (5)$$

其中,  $CP_{\text{cpu}}^s(n^s)$  和  $C_{\text{bm}}^s(n^s)$  分别表示构建的服务承载网为业务需求所分配的底层节点 CPU 计算能力和内存容量;  $Bw^s(l^s)$  表示构建的服务承载网为其分配的链路带宽资源。

有效的服务承载网构建即是利用有限的底层资源构建尽可能多的服务承载网,满足尽可能多的业务需求,换言之,即在构建相同数量的服务承载网时,最小化底层资源的使用。因此,相应给出以下 2 个定义。

定义 3 服务承载网构建成功率  $C_{\text{accepto}}$ 。服务承载网成功构建的个数与构建请求总数之比,即

$$C_{\text{accepto}} = \frac{C_{\text{success}}}{C_{\text{all}}} \quad (6)$$

定义 4 长期收益花费比  $RC$ 。构建服务承载网的收益与花费之比,即

$$RC = \lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T R(G^r, t)}{\sum_{t=0}^T C(G^s, t)} \quad (7)$$

## 2) 基于业务需求进行资源迁移

除了高效使用底层资源外，服务承载网构建的另一目标就是根据业务需求及底层资源运行状态，对已构建的服务承载网进行动态调整。当感知到底层资源超负荷工作时，及时根据业务需求将紧要资源上的业务相应地进行迁移。

在进行资源迁移时，将根据业务类型选择迁移对象。例如带宽敏感型业务的性能目标是最大化带宽，因此在迁移过程中，算法更倾向为其选择具有充沛资源的底层节点或链路进行映射；而时延敏感型业务的性能目标是最小化平均时延，因此算法将会优先选择低时延路径迁移，即迁移到离原映射节点较近的底层节点之上。

## 4 基于拓扑感知的服务承载网动态重构算法

### 4.1 资源度量

DTAR 算法定义资源关键度和资源紧要度分别作为底层资源属性和度量底层资源使用程度的指标，并作如下定义。

**定义 5** 资源关键度  $CR$ 。资源关键度是指网络中所有最短路径之中经过该资源的个数，包括节点关键度和链路关键度。归一化表示为

$$CR = \begin{cases} \sum_{i,j} \frac{\sum_{p \in P_{ij}^n} j_p^n}{|P_{ij}^n|}, n \in N^S \\ \sum_{i,j} \frac{\sum_{p \in P_{ij}^l} j_p^l}{|P_{ij}^l|}, l \in L^S \end{cases} \quad (8)$$

其中， $P_{ij}$  为节点  $i$  与节点  $j$  之间最短路径的集合， $j_p^n$  和  $j_p^l$  分别表示经过节点  $n$  和链路  $l$  的最短路径的数量。

资源关键度是由拓扑属性确定的定值。资源关键度越高，说明该资源越重要，倘若这些节点或链路失效，对网络的影响也就越大。因此在映射过程中，将优先选择关键度低的资源映射，确保网络均衡可靠使用。

**定义 6** 资源紧要度  $LR$ 。指各底层节点或链路的已用资源与该节点或链路的总资源之比，包括节点紧要度和链路紧要度。表达式如下

$$LR = \begin{cases} \left[ \frac{\sum_i C_{bn}^n(i)}{M}, \frac{\sum_i CP_{cpu}^n(i)}{A} \right], n \in N^S \\ \frac{\sum_i Bw_l(i)}{C}, l \in L^S \end{cases} \quad (9)$$

其中， $Bw_l(i)$  表示服务承载网在底层链路  $l$  上为业务需求  $i$  分配的带宽资源， $C$  为底层链路  $l$  的总带宽； $C_{bn}^n(i)$  和  $CP_{cpu}^n(i)$  分别表示底层节点  $n$  上，业务需求  $i$  被分配得到的缓存容量和 CPU 计算能力的使用情况， $M$  和  $A$  则分别表示节点  $n$  的缓存总量和 CPU 计算能力。

资源紧要度是随着底层资源的使用情况而动态变化的，资源紧要度越大，说明该底层节点或链路上的负载越高，负担越重。

## 4.2 重构算法

### 4.2.1 DTAR 算法思想描述

虽然已有一些将业务需求映射到底层网络的有效算法<sup>[12-15]</sup>，但随着时间的推移，成功构建服务承载网数量的增加将使部分底层资源处于超负荷运转状态，甚至濒临瘫痪，最终致使构建其上的服务承载网无效。针对这一问题，DTAR 算法分为 3 个阶段进行。

**Step1 关键资源感知。**由底层网络拓扑属性区分关键资源，计算得到从每个节点到其他所有节点的最短路径。那么，通过该资源的最短路径数越多，说明该资源越关键。

**Step2 服务承载网构建。**DTAR 算法基于资源关键度对底层资源降序排列，并依此序进行映射。考虑以关键度从小到大的顺序映射的原因如下：1) 为了增强网络的可靠性；2) 有利于网络资源充分使用。

**Step3 紧要资源迁移。**DTAR 算法通过感知底层资源的紧要度，将过载工作的底层节点或链路上的业务依据其需求进行相应迁移，将底层资源失效防范于未然。

图 2 给出了基于拓扑感知的服务承载网动态重构的示意。在图 2(a)中，服务承载网构建请求  $RSCN_1$  中的 3 个虚拟节点分别映射到底层节点  $\{a, d, e\}$ 。假设由底层拓扑属性已计算出节点及链路的资源关键度，其中，节点  $\{b, e, f\}$  为关键节点，设置资源紧要度阈值  $d$  为 80%，其他非关键资源的紧要度阈值  $s$  为 90%，即对具有不同重要性的

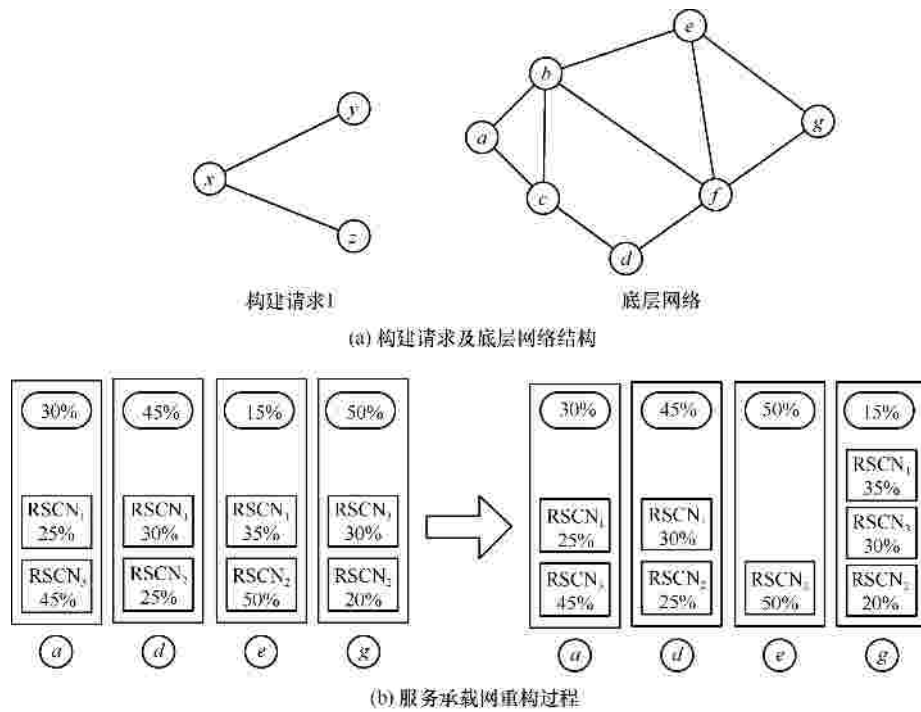


图 2 基于拓扑感知的服务承载网动态重构示意

节点或链路设置不同的资源紧要度阈值，资源越重要，其阈值设置越低。此时由于 RSCN<sub>1</sub> 的构建，感知到关键节点 e 的资源紧要度已超过阈值 80%，处于超负荷阶段。

图 2(b)中，将映射到关键资源 e 的服务承载网 RSCN<sub>1</sub> 的虚拟节点迁移到能够满足其业务需求的底层节点 g，并同时释放节点 e 上的相应资源。

#### 4.2.2 构建目标函数

高效构建服务承载网即是在构建过程中，最小化构建花费。

那么，最小化构建花费表示为

$$\min \sum_{n^s \in N^s} CP_{cpu}^s(n^s)CR_n + \sum_{n^s \in N^s} C_{bm}^s(n^s)CR_n + \sum_{l^s \in L^s} Bw^s(l^s)CR_l \quad (10)$$

$$\text{s.t.} \quad d_p < D_i \quad (11)$$

$$\sum_{n=1}^N C_{bm_n}^s < C_{bm}^s \quad (12)$$

$$\sum_{n=1}^N CP_{cpu_n}^s < CP_{cpu}^s \quad (13)$$

$$\sum_{n=1}^N Bw_n^s < Q \quad (14)$$

其中，式(11)表示构建服务承载网所选路径的时延小于业务构建请求的时延需求；式(12)表示当前承载业务的虚拟节点消耗的缓存资源之和小于节点

最大缓存容量；式(13)表示虚拟节点消耗的 CPU 计算资源之和小于底层节点最大 CPU 计算能力；式(14)表示虚拟链路消耗的带宽资源之和小于底层链路的最大带宽。

#### 4.2.3 关键资源感知映射

DTAR 算法为了更加合理地使用底层资源，由底层网络拓扑结构计算资源关键度 CR，确定关键资源，并为关键资源设置紧要度阈值 d，为非关键资源设置紧要度阈值 s，且 d < s，即确保关键资源不能被过度使用；动态感知底层资源状态，计算资源紧要度 LR；将底层资源按照资源关键度递减排序；进行节点映射时，选择能满足业务需求的资源关键度最小的资源依序进行映射，均衡充分地使用底层网络资源；而后以最短路径进行链路映射。算法 1 和算法 2 分别给出了关键资源感知以及基于关键资源的服务承载网构建的详细步骤。

##### 算法 1 关键资源感知算法

输入：G<sup>s</sup>

过程：

1) 对 G<sup>s</sup> 中任意节点 v 计算最短路径，得到从节点 v 到 G<sup>s</sup> 中其他所有节点的最短路径，形成一棵从 v 出发，叶节点为其他所有节点的树 SPT(v)。

2) 计算经过任意中间节点 w 最短路径的条数

```

sum(w)
  if w 在 SPT(v)中出现
    sum(w)+1 ;
  else w 未在 SPT(v)中出现
    sum(w) ;
3) 计算经过任意链路 e 最短路径的条数 sum(e)
  if e 在 SPT(v)中出现
    sum(e)+1 ;
  else e 未在 SPT(v)中出现
    sum(e) ;
4) 遍历 Gs 中每个节点和链路, 得到 sum(w)和
sum(e)。

```

5) 将每个节点的  $sum(w)$  值和每条链路的  $sum(e)$  值代入式(8), 分别计算每个节点和链路的资源关键度  $CR$ 。

6) 确定关键资源, 并设置资源紧要度阈值  $d$ ; 对非关键资源设置阈值  $s$ 。

7) 动态感知  $G^s$  中每个底层节点和链路的已占用资源情况。

8) 将已用资源值代入式(9), 得到每个底层节点和链路的资源紧要度  $LR$ 。

输出:  $CR$ 、 $LR$

算法 2 基于关键资源的服务承载网构建算法

输入:  $G^s$ 、 $G^r$ 、 $CR$ 、 $LR$

过程:

1) 根据  $G^s$ 、 $G^r$  的约束, 由式(10)得到满足基础物理资源与业务需求之间映射关系的路径集合  $G^m$ 。

if  $G^m$  非空

执行 2);

else  $G^m$  为空

无法构建满足业务需求的服务承载网络。

2) 得到  $G^m$  中节点集合  $S=\{S_1, S_2, \dots, S_n\}$ 。

3) 将  $G^m$  中底层资源按照资源关键度降序排列。

4) 节点映射, 选择  $G^m$  中资源关键度最小的节点映射。

for each  $n^v \subset N^r$  do

$S_{cand} \leftarrow$  满足节点 CPU 和内存约束, 且未被  $G^r$  映射的底层节点

完成节点映射:  $M^N(n^v) = \min_{n^s \in S_{cand}} CR(n^s)$

end

5) 链路映射, 以最短路径映射。

for each  $l^v = uv \subset L^r$  do

对于  $l^v$  的 2 个端点  $u$  和  $v$ , 节点映射结果分别为  $s$  和  $t$

计算  $s$  和  $t$  之间的最短路径  $p$

完成链路映射:  $M^L(l^v) = p$

end

输出:  $G^v$

#### 4.2.4 紧要资源迁移

DTAR 算法动态感知底层资源的资源紧要度, 对于超过资源紧要度阈值的底层资源, 将已映射其上的服务承载网按照业务需求进行迁移。对于时延敏感型业务, 优先选择将其迁移到离原映射节点近的底层节点, 确保迁移后的服务承载网仍具有较小时延; 对于带宽敏感型业务, 将在满足时延需求的前提下, 优先选择节点或链路资源充沛的底层资源进行映射。算法 3 为紧要资源迁移的详细步骤。

算法 3 紧要资源迁移算法

输入:  $G^s$ 、 $G^r$ 、 $G^v$ 、 $LR$

过程:

1) 动态感知  $G^v$  映射的底层资源紧要度  $LR$

if  $LR <$  资源紧要度阈值

维持现有服务承载网构建状态;

else  $LR$  资源紧要度阈值

执行 2)。

2) 确定  $G^v$  中过载资源类型。

if 节点过载

执行 3);

else 链路过载

执行 5)。

3) 将过载节点上的已映射资源依据业务需求迁移。

if 时延敏感型

迁移资源至离原映射节点距离最近的

底层节点, 且  $\sum_{n=1}^N C_{bm_n}^r < C_{bm}^s$ ,  $\sum_{n=1}^N CP_{cpu_n}^r < CP_{cpu}^s$ ;

else 带宽敏感型

迁移资源至可用资源最大的底层节点, 且  $d_p < D_l$ 。

4) 对成功迁移节点以最短路径进行链路映射。

5) 将过载链路上的已映射资源迁移至该链路所连接的 2 节点间 (除该链路外) 的最短路径。

输出:  $G^v$

### 5 算法仿真与分析

在效能评估中，DTAR 算法与基于底层资源负载状况的 G-SP<sup>[5]</sup>算法和基于连接度的 WD-VNE<sup>[11]</sup>算法进行性能比较，主要从服务承载网构建成功率、构建收益花费比以及资源均衡度 3 个方面进行讨论。

#### 5.1 实验设置

仿真实验中的基础网络拓扑是由 BRITE 工具随机产生的 100 个节点组成的，节点连接率为 0.5，节点与链路资源服从 50~100 的均匀分布。RSCN 节点个数需求满足 2~10 的均匀分布，链路带宽需求、节点 CPU 能力及缓存容量均为 0~30 的均匀分布，节点连接率为 0.5。业务构建 RSCN 请求的到达过程服从时间单位为 100，强度为 10 的泊松过程；每个 RSCN 的生存时间服从期望为 1 000 的指数分布。为了仿真结果的准确性，本文共进行仿真实验 20 次，所取结果为所有实验结果的平均值。

#### 5.2 服务承载网构建成功率

图 3 为随服务承载网构建请求数增多，不同算法的服务承载网构建成功率。由于 G-SP 算法在映射时，并未将底层资源的拓扑属性考虑其中，仅针对底层负载状况进行处理，因此其服务承载网构建成功率最低；而充分考虑了拓扑属性的 WD-VNE 算法与 DTAR 算法将底层资源区别对待，均表现出较强的服务承载网构建能力。当底层资源充足时，2 种算法的构建成功率相当；但随着服务承载网构建请求数的增多，由于资源连接度并不能准确地描述底层资源的重要性，因此 DTAR 算法的构建成功率将略高于 WD-VNE 算法。

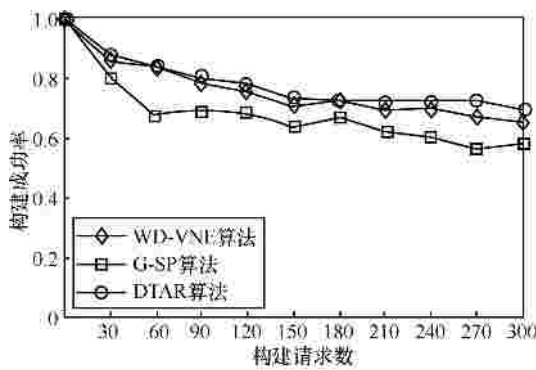


图 3 服务承载网构建成功率

#### 5.3 服务承载网构建收益花费比

图 4 为不同算法的服务承载网构建收益花费比。随着服务承载网构建请求数的增加，3 种算法的收益

花费比日趋均衡。由于 DTAR 算法以最小构建花费为目标进行服务承载网构建，且迁移时，仅针对过载资源上的部分服务承载网进行资源调整，因此其收益花费比率将高于不具备对已映射的服务承载网进行动态优化处理的 WD-VNE 算法，而未考虑底层资源差异性的 G-SP 算法的收益花费比最低。

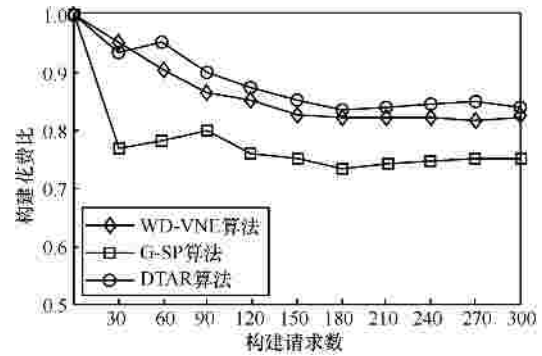


图 4 服务承载网构建收益花费比

#### 5.4 资源均衡度

所谓资源均衡度是指构建的服务承载网对底层网络资源的使用情况。分为网络节点均衡度和网络链路均衡度。

定义 7 节点均衡度。服务承载网构建时，整个底层网络上所用节点的平均处理能力与最大处理能力的比值。

$$N_c = \frac{\sum_1^N N_{c_i}}{N_{c_{max}}} \tag{15}$$

定义 8 链路均衡度。构建服务承载网时，整个底层网络上所用链路的平均利用率与最大链路利用率的比值。

$$LB = \frac{\sum_1^N LRatio_i}{LRatio_{max}} \tag{16}$$

图 5 和图 6 分别给出随服务承载网构建数量的增加，不同算法的资源均衡度情况。构建服务承载网时，DTAR 算法首先选择资源关键度低的节点映射，而避免对关键资源的过度使用，这必定使网络资源得到充分使用，因此其资源均衡度最高。G-SP 算法考虑到负载均衡问题，优先选用负载较低且距离较近的底层节点映射，而 WD-VNE 算法基于资源连接度和资源容量进行映射，对底层资源的使用相对集中，因此其资源均衡度最低。

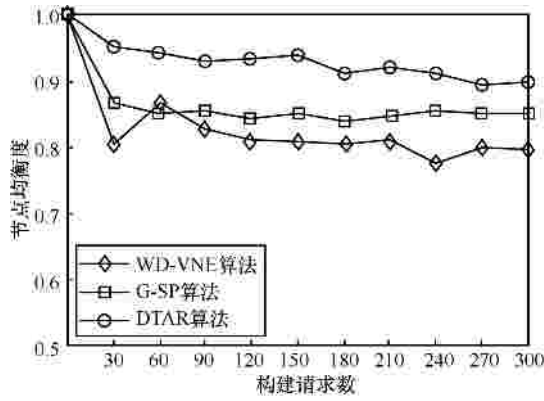


图 5 节点均衡度

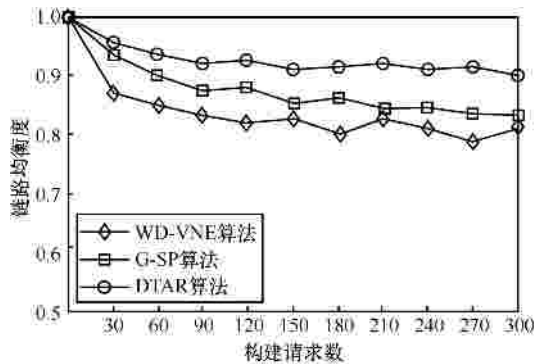


图 6 链路均衡度

## 6 结束语

高效地使用底层网络资源，以期最大化构建成功率，是服务承载网构建的永恒主题。基于此，本文提出了一种基于拓扑感知的服务承载网动态重构算法。DTAR 算法使用通过节点或链路的最短路径数衡量资源重要性，给出了“资源关键度”的定义，并依据资源关键度对资源降序映射；定义了“资源紧要度”指标对底层资源使用状态进行刻画，并将动态感知到的紧要资源进行优化配置。仿真结果表明：在与算法 G-SP 和 WD-VNE 进行的性能对比中，DTAR 算法在提高构建成功率的同时，具有较高的收益花费比和资源均衡度。

### 参考文献：

[1] 兰巨龙, 程东年, 胡宇翔. 可重构信息通信基础网络体系研究[J]. 通信学报, 2014, 35(1):187-198.  
 LAN J L, CHENG D N, HU Y X. Research on reconfigurable information communication basal network architecture[J]. Journal on Communications, 2014, 35(1): 187-198.

[2] YU M, YI Y, REXFORD J, et al. Rethinking virtual network embedding: substrate support for path splitting and migration [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 17-29.

[3] CHOWDHURY N, RAHMAN M, BOUTABA R. Virtual network

embedding with coordinated node and link mapping[C]//IEEE INFOCOM. Rio de Janeiro, c2009: 783-791.

[4] HOUIDI I, LOUATI W, ZEGHLACHE D. A distributed virtual network mapping algorithm[C]//IEEE ICC. c2008:5634-5640.

[5] ZHU Y, AMMAR M. Algorithms for assigning substrate network resources to virtual network components[C]//25th IEEE International Conference on Computer Communications (INFOCOM). Barcelona, c2006:1-12.

[6] LISHKA J, KARL H. A virtual network mapping algorithm based on subgraph isomorphism detection[C]//The 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures. Barcelona, c2009:81-88.

[7] KOIZUMI Y, ARAKAWA S, KAMAMURA S, et al. Adaptability of virtual network topology control based on attractor selection against multiple Node Failures[C]//OptoElectronics and Communications Conference held jointly with 2013 International Conference on Photonics in Switching (OECC/PS). Kyoto, c2013:1-2.

[8] BAVIER A, FEAMSTER N, HUANG M, et al. In VINI veritas: realistic and controlled network experimentation [J]. ACM SIGCOMM Computer Communication Review, 2006, 36 (4): 3-14.

[9] LU J, TURNER J. Efficient mapping of virtual networks onto a shared substrate[R]. Department of Computer Science and Engineering, Washington University, 2006.

[10] CHENG X, SU S, ZHANG Z, et al. Virtual network embedding through topology-aware node ranking [J]. ACM SIGCOMM Computer Communication Review, 2011, 41(2): 39-47.

[11] YUAN Y, WANG C, ZHU N, et al. Virtual network embedding algorithm based connective degree and comprehensive capacity[C]//9th International Conference on ICIC 2013. Nanjing, c2013: 250-258.

[12] ZHANG S, QIAN Z H, WU J, et al. Virtual network embedding with opportunistic resource sharing [J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(3): 816- 827.

[13] HU Q, WANG Y, CAO X J, et al. Virtual network embedding: an optimal decomposition approach[C]//International Conference on Computer Communication and Networks (ICCCN). Shanghai, c2014: 1-6.

[14] DIETRICH D, PAPADIMITRIOU P. Policy-compliant virtual network embedding[C]//International Conference on IFIP. Trondheim, c2014: 1-9.

[15] MELO M, SARGENTO S, KILLAT U, et al. Optimal virtual network embedding: node-link formulation [J]. IEEE Transactions on Network and Service Management, 2013, 10(4): 356- 368.

作者简介：



梁宁宁 (1982-), 女, 天津人, 博士, 国家数字交换系统工程技术研究中心讲师, 主要研究方向为宽带信息网络和下一代网络。

兰巨龙 (1962-), 男, 河北张北人, 博士, 国家数字交换系统工程技术研究中心总工程师、教授、博士生导师, 主要研究方向为新一代信息网络关键理论与技术。

张震 (1985-), 男, 山东济宁人, 博士, 国家数字交换系统工程技术研究中心讲师, 主要研究方向为网络测量技术。