# 基于广义网络温度GNT的虚拟网络映射方法

# 摘要

本文提出一种基于广义网络温度GNT(Generalized Network Temperature)的虚拟网络映射方法. 该方法首先从底层网络中提取流量特征, 计算底层节点的广义网络温度; 然后将节点GNT值与底层拓扑信息结合, 综合评估底层节点的重要性, 并按重要性对底层节点降序排序, 节点排名越高, 节点映射能力越强; 然后采用贪婪策略进行虚拟节点映射, 虚拟节点映射完毕后利用Dijkstra最短路径算法进行虚拟链路映射, 最终完成整个虚拟网络的映射过程. 仿真结果表明, 本文提出的两阶段启发式虚拟网络映射算法可以提高虚拟网络映射的成功率和底层网络资源的利用率, 同时缓解网络阻塞与资源碎片化问题.

# 引言

虚拟化技术 -> 虚拟网络映射问题 -> NP-hard

随着互联网的快速发展, 传统互联网的僵化问题愈发严重, 为了有效解决互联网的僵化问题, 引进了虚拟化技术. 而作为虚拟化技术的关键问题之一的虚拟网络映射问题, 由于

目前的启发式虚拟网络映射算法在进行节点重要性排序时, 只考虑网络拓扑结构与全局网络资源, 忽略了网络负载变化对于节点重要性排名的影响. 针对这个问题, 本文基于王昌达等提出的广义网络温度(Generalized Network Temperature, GNT), 提出了基于广义网络温度的虚拟网络映射算法. 该算法的优点是: 1) 将广义网络温度与网络拓扑特征结合, 可以更加全面的评估节点的重要性, 节点排序结果更加准确; 2)

第一个优点: 本算法采用了新型的节点排序方法, 同时考虑网络的静态和动态属性. 大部分基于节点重要性的两阶段启发式算法在进行节点排序时, 仅考虑网络拓扑特征(比如, 节点度属性, 节点强度属性)和网络资源(比如, 节点cpu资源与链路带宽和的乘积), 在映射虚拟网络时会出现节点cpu资源满足约束但链路资源不满足需求的情况, 导致虚拟网络映射失败.

第二个优点: 同时考虑网络的静态和动态属性, 节点的重要性会随着网络流量的变化而变化, 会影响虚拟网络映射的成功率.

第三个优点: 本文建立了一个仿真实验平台来验证所提出的算法的可行性, 并选取几个同类型且具有代表性的经典算法作为对比算法, 在相同的实验条件下进行仿真实验, 在虚拟网络映射成功率与长期收益支出比等评价指标上进行性能比较.

本文的创新点包括以下几个方面:

1. 本文在GNT的基础上, 选择

# 相关工作

随着互联网的蓬勃发展, 网络设备数量, 网络流量以及终端用户的数量呈指数倍增长, 互联网的僵化问题也日趋严重. 而虚拟化技术允许将多个虚拟网络部署在同一个底层网络上, 彼此之间互不干扰, 并且实现底层网络资源的共享, 最大化网络资源的利用率. 因此, 虚拟化技术被认为是解决互联网僵化问题, 推动互联网进一步发展的有效方案.

在虚拟化技术中, 虚拟网络由一系列虚拟节点以及连接虚拟节点的虚拟链路组成. 而底层网络是由许多底层物理节点与物理链路组成. 根据终端用户提出的虚拟网络请求, 将虚拟网络映射到共享的底层物理网络上的过程被成为虚拟网络映射(VNE, Virtual Network Embedding).

文献[15, 16, 17]是关于网络虚拟化与虚拟网络映射问题的研究综述, 并证明VNE问题是一个NP-hard问题. 在文献[曹的综述]中, 作者根据采用的优化策略将虚拟网络映射算法分为三类: 精确解算法类, 启发式算法类和元启发式算法类. 精确解算法和元启发式算法可以得到最佳或接近最佳的映射方案, 但算法复杂度太大, 无法应用到实际的网络环境中. 而上述两类算法不同的是, 启发式算法虽然

启发式算法一般将虚拟网络映射分成两个阶段, 即虚拟节点映射阶段和虚拟链路映射阶段. 同时, 在进行节点映射之前, 一般会根据网络拓扑属性和网络资源属性对底层物理节点和虚拟节点进行排序. 本小节重点介绍具有代表性的启发式虚拟网络映射算法:

在文献[36曹]中, Yu等人基于节点乘积值提出启发式映射算法: 首先对所有物理节点和虚拟节点进行排序, 然后使用贪婪策略进行节点映射, 最后采用Dijkstra方法[曹文献81]和多商品流(Multi-Community Flow, MCF)方法[曹文献81]来解决虚拟链路映射. Yu在进行节点排序时, 排序值由两部分组成: 节点自身的CPU资源和节点所有直连链路的带宽和, 排序值代表该节点在局部网络的映射能力, 排序值越大, 映射能力越强. 该节点排序方法简单方便, 但忽略了节点在整个网络中的影响力, 导致映射能力强的底层物理节点被过早浪费.

在文献[曹39]中, Cheng等人提出了一种基于拓扑感知的启发式虚拟网络映射算法. 与文献[曹36]不同的是, 在进行节点排序时, Cheng等人利用马尔可夫随机游走模型(Markov Random Walk model)来迭代地求解节点最终稳定的排序值. 但Cheng等人仍然忽略了节点在整个网络中的影响力.

在文献[曹41]中, Feng等人引入新的节点拓扑属性提出了三种不同的节点排序方法, 但其本质仍然是根据节点CPU, 直连链路带宽和以及物理路径的中间节点数来确定节点的排序值.

而在文献[曹43]中, Zhang等人将节点度属性和聚类系数作为节点排序的依据, 其余步骤与之前的算法相类似. 但节点度属性与聚类系数信息仍然属于局部拓扑属性, 忽略了全局拓扑属性.

其他文献:

综上所述, 现阶段大多数启发式虚拟网络映射算法采用的节点排序方法还是基于静态的网络拓扑与网络资源, 忽略了网络负载变化对节点重要性的影响.

# 相关知识

本节由两部分内容组成, 第一部分介绍VNE虚拟网络映射问题模型的建立, 虚拟网络映射过程以及相关评价指标; 第二部分详细介绍广义网络温度的相关内容, 包括: 流量矩阵的定义, 三种网络熵的定义与性质, 最后介绍广义网络的定义与计算方法.

## 3.2 虚拟网络映射问题

本节介绍虚拟网络映射问题, 包括: 网络模型的建立, 虚拟网络映射流程以及相关评价指标.

### 3.1.1 底层网络

底层网络可以用一个带权无向全连通图，其中，下标”S”是”Substrate”的首字母, 和分别代表底层网络物理节点集合与物理链路集合，和分别代表的是底层网络中的节点和链路数量. 和分别表示底层网络节点和链路的相关属性, 节点属性包括: 计算能力, 内存大小, 地理位置等, 链路属性则包括: 带宽容量, 传输延迟等. 与大多数先前研究相同, 文本将CPU计算能力和地理位置限制作为节点属性, 将带宽容量作为链路属性. 另外, 表示底层网络中无环路径的集合.

如图2.2(a)所示，底层网络具有8个节点和13条链路，节点和链路旁的数字分别代表节点和链路所具有的计算和带宽资源.

### 3.1.2虚拟网络模型

与底层网络类似，虚拟网络也是由虚拟化后的虚拟节点通过虚拟链路连接组成，同样可以用带权无向全连通图表示: . 其中，和分别代表虚拟网络中的虚拟节点和虚拟链路，而和则分别代表虚拟节点和虚拟链路的数量. 虚拟节点和虚拟链路分别附带有以表示的CPU需求, 地理位置需求和以表示的带宽容量需求. 除此之外, 用表示虚拟网络请求的到达时间, 用表示虚拟网络请求的持续时间. 当收到一个虚拟网络请求时, 需要为其分配底层网络资源. 如果当前底层网络能够满足该虚拟网络请求的各项需求和约束条件, 就接受该请求, 反之, 拒绝该请求. 最后, 用表示虚拟节点可以接受的对应底层节点实际地理位置与期望地理位置的最大距离.

如图2.2(b)所示，该虚拟网络包含3个虚拟节点和3条虚拟链路，节点和链路旁的数字分别代表虚拟节点和虚拟链路所需要的计算和带宽资源量.

### 3.1.3虚拟网络映射过程

在满足虚拟网络的节点和链路各项需求与约束条件的前提下，将虚拟网络映射到底层网络的过程称为虚拟网络映射(Virtual Network Embedding，VNE)，该过程通常包含两个子过程: 节点映射子过程和链路映射子过程.

1. **虚拟节点映射子过程**: 将虚拟网络所有的虚拟节点映射到互不相同的底层物理节点上, 并且节点的CPU计算需求与地理位置需求必须得到满足. 对于所有的节点

Subject to:

其中, 是虚拟节点映射函数.

公式某某和某某保证每个虚拟节点的CPU计算需求和地理位置需求都能得到满足, 而公式某某某保证了来自同一个虚拟网络请求的不同虚拟节点不会映射到同一个底层物理节点上. 但来自不同虚拟网络的多个虚拟节点可以共享同一个底层物理节点.

1. **虚拟链路映射子过程**: 将虚拟网络所有的虚拟链路映射到底层物理网络的不同物理路径上, 并且虚拟链路的带宽容量需求都必须得到满足. 对于所有的虚拟链路:

Subject to:

其中, 是虚拟链路映射函数.

公式某某保证每条虚拟链路的带宽容量需求都能得到满足.

只有当虚拟节点映射子过程和虚拟链路映射子过程都成功完成, 虚拟网路的映射才算成功, 否则虚拟网络映射失败. 当虚拟网络请求被成功映射之后, 在其持续时间内都将占有所分配的网络资源, 生命周期结束之后, 才会将其占有的网络资源释放掉, 用以重新分配给后续的虚拟网络请求.

### 3.1.4评价指标

在进行虚拟网络映射时，需要利用评价指标对算法性能进行量化与评估，有利于与其他算法进行比较. 本节介绍VNE算法研究中常用的评价指标:

1. **虚拟网络映射接受率**: 又称虚拟网络映射成功率，是VNE算法研究中最重要的评价指标. 虚拟网络映射接受率由映射成功的虚拟网络数量占全部虚拟网络请求的比值所决定，直接反应算法的映射性能，即接受率越高，算法的映射能力越强. 虚拟网络映射接受率的计算公式如下所示:

其中，分子表示从零时刻开始经过T时间成功映射的虚拟网请求的数量，而分母表示这段时间内的经过算法处理的虚拟网络请求的总数量.

1. **虚拟网络映射收益**: 虚拟网络服务提供商通过将虚拟资源以虚拟网络的形式租给终端用户，并向终端用户收取费用，这部分费用即虚拟网络的映射收益. 对于虚拟网络服务提供商来说，需要尽可能最大化虚拟网络的映射收益. 虚拟网映射收益的计算公式如下:
2. **虚拟网络映射成本**: 又称虚拟网络映射支出. 除了最大化虚拟网络映射收益之外，虚拟网络服务提供商还需要付出虚拟网络映射成本, 并尽可能最小化该成本. 虚拟网络映射成本的计算公式如下:

符号表示底层链路是否被虚拟网络占用，若占用，取值为1，否则取值为0.

1. **虚拟网络收益支出比**:在VNE算法研究中，单独使用虚拟网络收益和支出不能全面地反映算法的性能, 为了避免该缺陷, 通常使用虚拟网络收益支出比来讨论算法的性能，计算公式如下:

更一般的，使用长期收益支出比来评价算法的性能，计算方式如下:

### 3.1.4 网络状态更新

1. 主动更新: 每隔固定的时间间隔根据当前的网络负载状态更新网络流量矩阵.
2. 被动更新: 只有在收到虚拟网络请求时, 才更新网络流量矩阵.

## 3.2 广义网络温度

### 3.2.1 广义网络温度

广义网络温度(Generalized Network Temperature, GNT)是王昌达等人提出的一种兼顾网络拓扑与网络流量的度量指标, 具有时空融合特性. 该指标在网络攻击检测, 网络安全态势评估等多个领域具有广泛的应用场景. 文献(韦尧)使用GNT作为检测指标对DDoS攻击进行检测, 取得了较好的检测结果. 在文献(陈前)中, 作者提出了基于GNT与网络拓扑结构的节点重要性排序方法.

在热力学中, 温度T定义为一个绝热系统中热量的引进或流失额度与由此导致的热力学熵的变化量的微商. 在文献(老板)中, 作者类比热力学温度, 用网络的Shannon熵取代热力学熵, 用网络中”在传输数据包数量的变化”取代热力学中”热量的引进或流失额度”, 最终得到新的度量指标, 即广义网络温度GNT, 具体定义如下:

其中, 表示网络中传输数据包数量的变化, 而表示由数据包数量变化导致的Shannon熵的变化.

与热力学温度类似, GNT数值一般不会出现较大波动, 当GNT发生突变时, 表示网络中数据传输出现异常, 可能收到攻击.

### 3.2.2 网络熵

本小节介绍GNT中使用的网络熵, 包括: 香农Shannon熵, Rényi 熵和Tsallis熵.

#### 3.2.2.1 香农Shannon熵:

信息论之父香农将统计物理中熵的概念引申到信道通信的过程中, 并将其定义为信息熵(也称香农熵), 从而实现对信息的度量, 奠定了信息论和数字通信的基础. 对于一个随机变量, 其香农熵定义为:

其中, 为变量的可能取值, 其对应的概率为. 底数对应香农熵的量纲, 时量纲为bit(比特), (自然对数底)时量纲为nat(纳特), 本论文中, 如无特殊说明, 后面的讨论中将省略底数, 统一取(即量纲为纳特), 并用表示.

香农熵的性质: 1) 连续性: 变量的香农熵会随着概率发生连续性变化; 2) 单调性: 变量的香农熵随着变量取值的增多而单调增加; 3) 可加性: 不同来源的信息可以叠加, 具体如下式:

在文献(老板)中, 作者使用Shannon熵作为GNT的网络熵, 但在某些特定场景下, Shannon熵的效果略差于其他类型的熵, 因此为了更好地提高算法的普适性, 将引入更多类型的网络熵.

#### 3.2.2.2 Rényi 熵

Rényi 熵是香农熵的广义形式, 由香农熵稍微放宽可加性条件得到, 具体定义如下:

相比香农熵, Rényi 熵引入可调节参数, 对信息量的度量更具一般性与灵活性. 特别地, 当时, Rényi 熵退化为香农熵, 即.

#### 3.2.2.3 Tsallis熵

而Tsallis熵则是由巴西物理学家Constantino Tsallis于1998年提出, 其具体定义如下:

Tsallis 具有伪可加性, 具体体现为:

其中, 同Rényi 熵一致, 是可调节参数. 当时, 值越小, 的值越接近于0, 此时高概率事件的分布相比于低概率事件更明显, 因此Tsallis熵在时主要体现高概率事件的分布状态; 相反, 在时, 主要体现低概率事件的分布状态.

综上所诉, 香农熵更具广泛和普遍意义, 而Rényi 熵和Tsallis熵则通过引入可调节参数对香农熵进行进一步的推广, 有所侧重地对信息进行度量.

# 基于GNT的虚拟网络映射方法

本节兼顾网络流量变化和网络拓扑与网络资源信息提出了一种启发式虚拟网络映射方法. 本方法首先从底层网络提取流量信息, 得出流量矩阵, 然后计算节点的GNT, 之后与网络拓扑特征结合, 综合评价节点的重要性, 进行节点排序. 然后对虚拟网络进行两阶段映射, 首先基于节点重要性, 利用贪心策略进行节点映射, 在节点映射完成后, 利用Dijkstra最短路径算法进行虚拟链路的映射, 最终完成整个虚拟网络的映射.

启发式虚拟网络映射算法主要分为三个步骤, 包括: 网络节点排序, 虚拟节点映射, 虚拟链路映射.

在进行虚拟网络映射时, 本文兼顾网络流量变化和网络拓扑资源信息, 提出了VNE-GNT算法. 其基本思想是: 在底层网络中, 网络节点附近的网络流量负载变化越大, 其附近的GNT广义网络温度也就越大. 因此, 通过广义网络温度GNT衡量网络流量变化对于网络熵的影响, 寻找负载变化明显的节点, 同时结合网络拓扑与网络资源信息, 综合衡量节点的重要性. 节点排序完成之后, 利用贪心策略进行虚拟节点的映射, 然后利用Dijkstra最短路径算法进行虚拟链路的映射, 最终将整个虚拟网络映射到底层网络中.

## 4.1 网络节点排序

在进行虚拟网络映射时, 不同重要性的节点对于映射结果的影响是不同的, 节点越重要, 对映射结果的影响越大.

一般的, 网络节点的重要性不仅与网络的拓扑结构和自身网络资源有关, 也与当前的网络负载状况有关, 节点重要性会随着网络负载变化而变化. 然而, 目前已知的启发式虚拟网络映射算法在进行节点排序时, 大多是基于网络拓扑与网络资源的静态分析的方法. 在网络状态不断变化的情况下, 这些方法普遍不能全面衡量网络节点的重要性, 导致虚拟网络映射成功率和资源利用率的降低.

因此, 本文在进行节点重要性排序时, 将网络拓扑结构和网络资源与网络负载变化相结合, 综合衡量节点的重要性.

首先引入网络流量矩阵, 用以表示一个网络的负载情况. 通过网络流量矩阵可以得到底层网络流量的全局视图, 能够从更宏观的角度观察网络状态, 用于之后计算网络节点的广义网络温度GNT.

定义 1 网络流量矩阵: 由当前网络中所有的OD(Origin-Destination)流量组成的矩阵, 其中OD流定义为从节点(源节点)进入, 最后流出节点(目标节点)的所有流量的集合.

用表示一个拥有个网络节点的网络的流量矩阵, 包含个元素. 如下图所示的网络流量矩阵(单位为kbps)的例子, 行代表源节点, 列代表目的节点, 矩阵第行元素的和代表从源节点流出的总流量, 而矩阵第列元素的和则代表着流入目的节点的总流量.

(此处包含一个网络流量矩阵示例图)

根据网络熵和网络流量矩阵的定义, 可以将一个源节点附近聚集的所有OD流看成是基本状态, 满足上文提到的网络熵概率的条件, 将状态的概率定义为:

其中, 代表从源节点流出的网络流量总和.

根据三种网络熵的定义, 节点附近的网络熵计算公式如下:

进一步得到节点的广义网络温度GNT:

形式上, 广义网路温度是网络流量相对于网络熵的导数, 其物理意义是网络熵随网络流量变化的变化率. 节点附近网络流量变化越快, 网络熵也变化越快, 节点的GNT也就越大. 通过广义网络温度GNT就可以快速找到负载变化明显的节点, 避免虚拟网络映射到这些节点上, 造成阻塞.

网络流量矩阵中的元素只体现源节点和目的节点之间的OD流信息, 不包括OD流的路径信息, 因此还需要结合网络拓扑信息与节点自身网络资源信息, 综合评价节点的重要性, 提高节点排序的准确性.

小标题: 网络拓扑信息

网络节点的度中心性(DC)定义为与该节点直接相连的节点数量, 体现了该节点与其他节点的连接关系, 具体定义如下:

其中, 代表网络中节点的数量.

(性质)与节点直接相连的节点数量越多, 节点的度中心性也就越大, 就越靠近网络的中心, 对于虚拟网络映射结果的影响也就越大. 但, 节点的度中心性只表示节点的在网络拓扑中的局部信息, 无法衡量节点在整个底层网络中的影响力, 降低节点重要性排序的准确性.

网络节点的接近中心性(CC): 衡量节点居于网络中心的程度, 越靠近网络中心, 对于整个网络的影响越大, 越有可能成功映射.

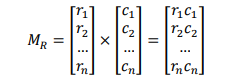
节点排序的主要步骤:

(1) 根据底层网络拓扑结构和网络流量信息构建网络邻接矩阵和网络流量矩阵

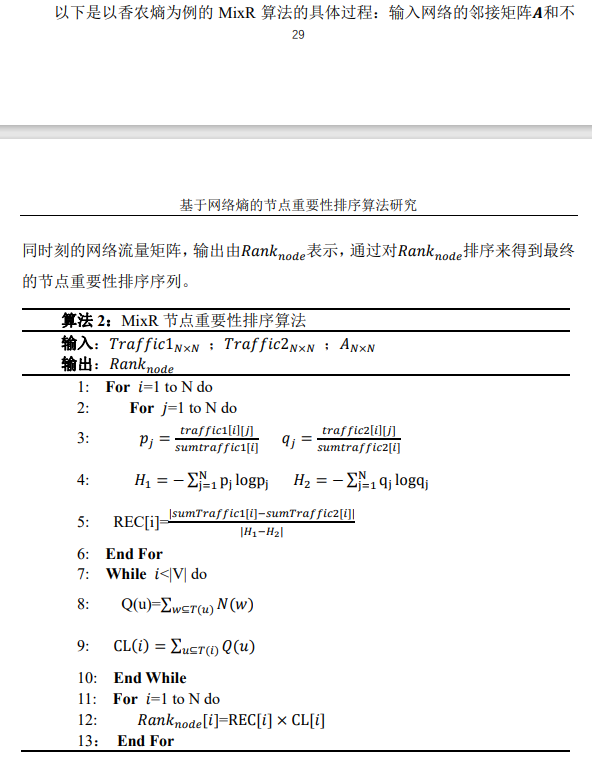
(2) 基于网络拓扑信息求出各个网络节点的度特征

(3) 根据不同时刻的网络流量矩阵求出节点的GNT广义网络温度: .

(4) 将三者结合, 得到对应节点最终的重要性值, 计算公式如下:



(5) 最后, 按重要性值降序排序, 值越大, 对应的节点越重要.



算法复杂度分析:

实例分析:

本节结合一个简单的网络来说明节点排序的主要步骤, 如图所示, 是一个含有10个节点的底层网络, 该网络某时刻的网络流量矩阵如下, 单位是kbps.

网络节点自身的cpu与带宽资源放在网络节点旁边的方框中,

时刻的网络流量矩阵:

时刻的网络流量矩阵:

网络的邻接矩阵如下: 用于计算节点度属性等等

然后就是节点排序每一个步骤的小结果:

1. 根据公式某某某, 计算得出节点在时刻的香农熵为: , 以及时刻节点的香农熵为: ;
2. 根据网络流量矩阵, 求出不同时刻节点总的流量: dskjfrvdlsfvhloisuhfvc
3. 根据公式某某某, 求出节点基于香农熵的广义网络温度:
4. 根据网络的邻接矩阵求得节点的度属性: iadsluhfvclashbfnvcl
5. 根据节点自身的网络资源信息求得节点的资源重要性: lesfhvsdjhnvfldsjnfv
6. 最终, 节点基于香农熵的重要性值为: lsadfhvlasjfvljadsbnl. 并按照重要性值降序排序, 结果如下:

这里是一个表格, 展示节点的重要性排名结果

结果分析: 排序结果表明, 节点n2的重要性值为, 在所有节点中是最大的, 因此排名第一, 是最优可能映射成功的节点; 而节点某某某的重要性值为某某, 在所有的节点中排名最后.

随着网络流量矩阵与网络资源的消耗, 节点的重要性排序结果也会发生变化, 这体现了方法的动态性

## 4.2 虚拟节点映射

在节点排序完成之后, 采用贪婪策略进行虚拟节点的映射. 具体流程如下: 对于某个虚拟节点来说, 从底层网络的所有节点中挑选节点排序值最高的网络节点. 如果底层节点可以满足虚拟节点的cpu资源约束, 那么该虚拟节点就成功映射到这个底层节点上. 但如果底层节点不能满足约束条件, 就需要从所有底层节点中挑选节点排序值第二高的底层节点, 继续上述过程, 直到找到合适的底层节点. 如果所有底层节点都不能满足该虚拟节点的资源需求, 那么该虚拟节点映射失败.

在虚拟节点映射成功后, 会采用上述策略继续映射剩下的虚拟节点, 同时还要保证满足”同一个虚拟网络的不同虚拟节点不会映射到同一个底层节点”的约束条件.

当且仅当一个虚拟网络请求的所有虚拟节点都成功映射时, 节点映射才算成功, 否则, 如果有任意一个虚拟节点没有被成功映射, 那么该虚拟网络就会被拒绝.

图

## 4.3 虚拟链路映射

在所有虚拟节点都被成功映射之后, 开始进行虚拟链路的映射. 假设虚拟节点和之间存在一条虚拟链路, 而这两个虚拟节点分别映射到底层节点和上. 与传统最短路径方法不同的是, 在进行链路映射之前, 首先将底层节点和之间所有不满足虚拟链路带宽需求的链路删除, 然后再采用Dijkstra最短路径算法计算出底层节点和之间的最短路径, 同时组成该最短路径的所有底层链路都满足虚拟链路的带宽需求, 这样, 虚拟链路映射成功. 对剩余虚拟链路继续采用上述策略, 直至所有虚拟链路都成功映射. 如果有任意一条虚拟链路映射失败, 则整个虚拟链路映射失败.

流程图和伪代码

当且仅当所有虚拟节点和虚拟链路都被成功映射, 才会接受该虚拟网络请求, 否侧, 拒绝该虚拟网络请求.

## 4.4 算法复杂度分析

此外, 本文还讨论VNE-GNT算法的复杂度. 算法复杂度主要由两部分组成: 虚拟节点映射部分和虚拟链路映射部分. 其中, 虚拟节点映射部分的复杂度为, 而虚拟链路映射部分的复杂度为. 因此, 该算法可以在多项式时间内完成虚拟网络请求的映射.

# 实验评估

本节首先设计了基于三种网络熵的虚拟网络映射实验, 然后将本文提出的算法与经典的虚拟网络映射算法进行对比, 以验证本文算法的可行性与优越性.

## 5.1 实验环境和参数设置

为了评估算法的性能，本文进行了仿真实验. 算法的仿真工作所采用的底层物理网络的参数设置如表5.1所示.这些参数设置是VNE算法研究领域具有代表性的参数设置:

表格 5.1 底层物理网络的参数设置

|  |  |
| --- | --- |
| 底层网络生成方法 | Waxman方法, |
| 底层节点个数 | 50 |
| 节点容量 | [50, 100], 整数, 均匀分布 |
| 链路带宽 | [50, 100], 整数, 均匀分布 |
| 底层节点的位置 | [0, 100]区间, 随机分布 |

虚拟网络的参数设置如表5.2所示:

表格 5.2 虚拟网络的参数设置

|  |  |
| --- | --- |
| 虚拟网络生成方法 | Waxman方法, |
| 虚拟网络到达率 | 10个/1000单位时间 |
| 每个虚拟网的生命周期 | 1000单位时间 |
| 虚拟节点个数 | [2, 10], 整数, 随机分布 |
| 节点CPU需求 | [1, 20], 整数, 均匀分布 |
| 链路带宽需求 | [1, 20], 整数, 均匀分布 |
| 虚拟节点位置 | [0, 100]区间, 随机分布 |
| 虚拟节点最大允许偏差 | [3, 8], 整数, 均匀分布 |

本章节的算法仿真的工作时长为100000个时间单位. 在本文中，一个时间单位代表一分钟. 也就是说, 本章节的算法仿真工作将持续100000分钟.

此外, 在使用Rényi熵和Tsallis熵进行节点重要性排序时, 可变参数的取值对广义网络温度的计算有影响, 为了得到最佳的性能, 本文与文献保持一致, 即使用Rényi熵时, 而使用Tsallis熵时.

本章节的算法实验仿真工作是在个人工作站上进行的, 操作系统是Windows 11, 用的是AMD Ryzen 5 4600H with Radeon Graphics, 配备16.0GB的RAM.

## 5.2 对比算法介绍

本文提出的两阶段启发式虚拟网络映射方法使用(香农熵, 瑞利熵, T熵)三种网络熵来计算网络节点的广义网络温度指标. 除了本文提出的算法, 还选取几个同类型且具有代表性的经典算法作为对比算法, 包括: G-SP, VNE-DCC, MO-NPSO, RW-BFS, 在相同的实验条件下进行仿真实验, 进行性能比较.

表格 5.3 对比算法介绍

|  |  |
| --- | --- |
| 算法名称 | 算法描述 |
| G-SP | 基于节点cpu与带宽和的乘积进行节点排序, 贪心映射虚拟节点, Dijkstra最短路径算法进行虚拟链路映射 |
| VNE-DCC | 基于节点度与聚类系数进行节点排序, 利用BFS算法进行虚拟节点映射, 最后采用K-Shortest算法映射虚拟链路 |
| M0-NPSO | 利用多目标改进粒子群优化算法进行虚拟网的映射 |
| RW-BFS | 利用随机游走对节点排序, 利用BFS算法同时完成虚拟节点与虚拟链路的映射 |

## 5.3 基于三种网络熵的虚拟网络映射实验

实验未完成, 后续添加内容

## 5.4 经典算法对比试验

实验未完成, 后续添加内容

# 结论与展望

大多数启发式虚拟网络映射算法在进行节点排序时, 只考虑网络拓扑和网络资源等静态信息, 而忽略了网络流量的变化对网络节点重要性的影响. 针对此问题, 设计实现了一种基于广义网络温度的启发式虚拟网络映射方法. 与现有的虚拟网络映射算法相比, 本文兼顾网络流量变化和网络拓扑与资源信息, 综合评估网络节点的排序值, 是典型的静态与动态相结合的节点排序方法.

实验结果表明, 本文提出的基于广义网络温度的虚拟网络映射方法与经典的VNE算法相比, 可以得到更加准确的节点排序结果, 从而提高虚拟网络的映射成功率与底层网络资源的利用率.

但是, 本文也存在一些不足之处. 比如在计算网络节点的广义网络温度时, 使用了三种网络熵: Shannon熵, Rényi熵和Tsallis 熵. 因此下一阶段的主要工作就是进一步研究这些网络熵的性质, 比如阶数对算法映射性能的影响. 同时, 还可以引入更多形式的网络熵, 进一步提升算法的性能.

另外, 在网络节点排序时, 只考虑了节点度作为节点拓扑特征, 而网络的拓扑特征对于网络节点的重要性是必不可少的. 因此可以考虑引入更多拓扑特征, 通过研究单一特征对算法映射性能的影响, 从而选择出对算法性能影响最大的因素, 提升算法性能.

最后, 同大多数VNE算法一样, 虚拟网络在映射到底层网络之后, 资源分配方案在虚拟网络生命周期内都不会发生改变. 但底层网络是动态的, 随着虚拟网请求的不断地到来和离去, 底层网络资源被不断占用与释放, 导致底层网络的资源碎片化愈发严重, 降低后续到来的虚拟网请求的接受率. 因此, 对虚拟网络的映射方案进行动态调整是十分必要的, 比如可以周期地将部署在负载高且资源少的区域的虚拟网络迁移到负载低且资源多的区域, 提高整个网络的资源利用率, 缓解资源碎片化与网络阻塞问题.

参考文献:

[某某] Waxman B M. Routing of Multipoint Connections [J]. IEEE Journal of Selected Areas in Communications, 1988, 6(9): 1617-1622.