

分布式环境中基于协商的虚拟网映射算法

江逸茗¹, 兰巨龙¹, 程东年¹, 吴方明²

(1. 国家数字交换系统工程技术研究中心, 河南 郑州 450002; 2. 吉林大学 计算机科学与技术学院, 吉林 长春 130022)

摘要: 网络虚拟化技术可以在共享的底层物理网络上为用户同时提供多种可定制的服务网络。目前的虚拟网映射算法比较依赖于集中式的管理节点, 使其在可靠性和适用范围等方面存在诸多问题。为此, 提出了一种分布式环境下的虚拟网映射算法, 该算法通过多个节点之间的相互协商来完成虚拟网的映射, 并且在降低通信开销和缩短虚拟链路的路径长度方面进行了相应改进。实验结果表明, 该算法与同类型算法相比, 在资源利用率和通信开销方面具有一定的优越性。

关键词: 虚拟网; 分布式映射算法; 映射协议; 通信开销

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2014)12-0062-08

Virtual network embedding algorithm based on negotiation in distributed environment

JIANG Yi-ming¹, LAN Ju-long¹, CHENG Dong-nian¹, WU Fang-ming²

(1. National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450002, China;

2. College of Computer Science and Technology, Jilin University, Changchun 130022, China)

Abstract: The network virtualization is able to offer various customized end-to-end services networks on a shared substrate network. However, the existing virtual network embedding algorithms present many problems in reliability and applicability due to their dependence on a centralized manager node. Hence, a virtual network embedding algorithm in distributed environment is proposed, which implements embedding process by the negotiation of several nodes. Moreover, this algorithm makes improvement in reducing communication cost and length of virtual links. Simulation experiments demonstrate that the proposed algorithm can achieve better performance on efficiency of resources utilization and communication cost.

Key words: virtual network; distributed embedding algorithm; embedding protocol; communication cost

1 引言

随着规模的不断扩大, 现有互联网结构僵化、可扩展性差的缺点日益突出, 为不同网络体系之间的融合以及网络的更新换代增加了一定困难。网络虚拟化技术^[1,2]成为了解决上述问题的重要技术手段。通过网络虚拟化技术, 服务提供商可以在共享

的底层物理网络上创建多个相互隔离的虚拟网(VN, virtual network), 从而为用户提供多样化的可定制端到端服务^[3], 例如为网络视频会议或IPTV^[4]等服务建立专门的虚拟网, 以实现具有特定传输需求的高质量服务。在虚拟网的运行环境中, 基础设施提供商(InP, infrastructure provider)负责管理和运营底层网络, 服务提供商(SP, service provider)以虚拟网

收稿日期: 2013-06-18; 修回日期: 2013-08-02

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(2012CB315901, 2013CB329104); 国家自然科学基金资助项目(61309019, 61372121); 国家高技术研究发展计划(“863”计划)基金资助项目(2013AA013505, 2011AA01A103, 2011AA01A101)

Foundation Items: The National Basic Research Program of China (973 Program) (2012CB315901, 2013CB329104); The National Natural Science Foundation of China (61309019, 61372121); The National High Technology Research and Development Program of China (863 Program) (2013AA013505, 2011AA01A103, 2011AA01A101)

请求的方式向 InP 申请网络资源并建立服务。虚拟网的实现基础是解决虚拟网请求的映射问题，也就是通过为虚拟网请求分配相应的底层网络资源，来实现虚拟网到底层网络的映射。

目前，针对虚拟网映射问题的研究大部分采用了集中式的映射方法^[5-12]，即由一个管理节点完成网络状态的收集维护和映射方案的决策。如果该管理节点出现故障，则整个网络将面临瘫痪的风险，这种单点失效(single point of failure)现象将会给虚拟网的可靠性和稳定性带来较大的负面影响。

分布式的映射方法由于不依赖这种集中式的管理节点，使其在管理节点出现故障时仍然能够正常的进行虚拟网的映射和维护，从而有效地避免了单点失效现象，但现有的分布式算法^[13]仍然存在 3 点不足：1) 算法要求网络的全部节点周期性地相互交换资源状态信息，这会带来大量的通信开销；2) 在映射各个星形子拓扑时仅考虑了节点的负载压力，导致逻辑上相邻的星形子拓扑可能分别被映射到相距较远的底层节点上，使部分虚拟链路占用了过多的底层链路资源；3) 该算法未考虑底层节点和链路的最大资源能力限制，也未考虑并行映射时的资源冲突问题。

针对现有的分布式算法存在的问题，本文设计了一种基于节点协商的分布式虚拟网映射算法，该算法不依赖集中式的管理节点，不在全网范围内进行大规模的状态信息交换，仅通过节点之间的协商和状态查询来实现虚拟网的映射，从而降低了映射过程中产生的通信开销。同时为了提升映射请求的响应速度，部分子算法可在多个节点上并行执行，并设计了相应的机制来避免并行执行过程中产生冲突。

2 虚拟网映射问题模型

2.1 网络映射模型

虚拟网映射问题的模型描述如下。

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 CPU 处理能力和最大带宽。

虚拟网请求。一个虚拟网请求包括虚拟网拓扑 G^v 、映射请求到达时间 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需求约束。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。

虚拟网映射问题。虚拟网映射问题可以描述为将虚拟网请求的拓扑 G^v 映射到底层网络拓扑 G^s 上，并且该映射要满足 R_N^v 和 R_L^v 的约束

$$M : G^v \rightarrow (N^s, L^s, R_N^s, R_L^s) \quad (1)$$

其中， $N^v \subset N^s$ 且 $L^v \subset L^s$ ， R_N^s 和 R_L^s 是底层网络为虚拟网请求分配的节点和链路资源。虚拟网映射问题可以细分为节点映射和链路映射。图 1 给出了虚拟网映射实例。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，该映射请求将被拒绝。

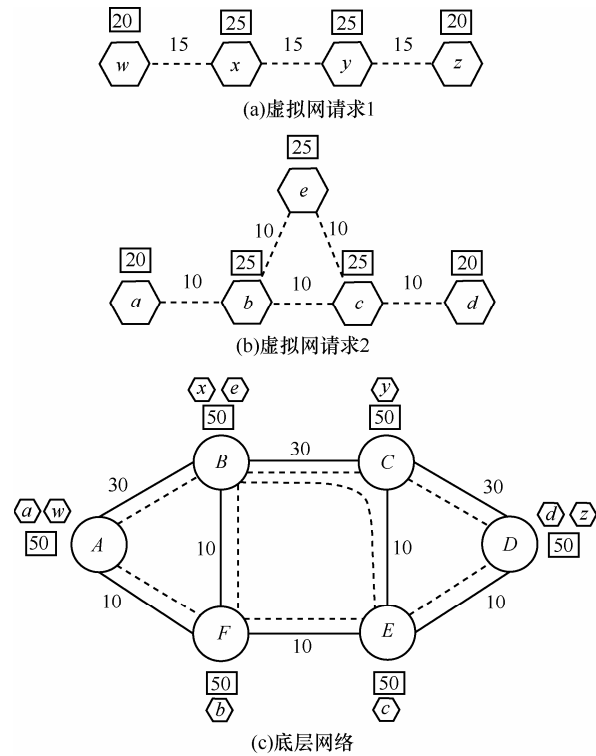


图 1 虚拟网映射实例

2.2 映射目标

虚拟网的映射目标主要是增加的 InP 的收益，同时降低其映射的开销。InP 的收益是由请求接收率(AR, acceptance ratio)来决定的，其定义为

$$AR = \frac{R_s}{R_T} \quad (2)$$

其中, R_S 是成功映射的虚拟网数量, R_T 是虚拟网请求总数量。要想提高请求接收率就需要对资源进行合理分配, 资源的合理利用性体现在 2 方面: 1) 负载均衡, 也就是避免出现底层网络的部分节点或链路负载过高而其他节点或链路却利用率较低的现象; 2) 最短路径优先, 一条虚拟链路的路径可能由多条底层链路组成, 若路径的长度越长则意味着虚拟链路占用的带宽资源也越多。因此, 采用最短路径优先原则一方面能够节约带宽资源, 提高请求接收率, 另一方面能够降低 InP 的映射开销。

2.3 映射环境

虚拟网的映射环境可以分为集中式映射环境和分布式映射环境。

集中式映射环境指的是在底层网络中存在至少一个中心管理节点, 该节点负责收集和维护整个底层网络的状态信息, 同时还负责虚拟网的映射、撤销和正常运行维护。由于中心管理节点掌握了网络的全局信息, 因此可以根据底层网络的资源利用状态来制定虚拟网的映射策略, 从而更加合理地分配底层网络资源。但集中式映射环境也存在一些局限性, 具体如下。

1) 若中心管理节点出现故障, 将使底层网络在故障期间无法处理虚拟网请求, 并且对正在运行的虚拟网也将产生较大的负面影响, 降低了整个网络的可靠性和稳定性。

2) 虚拟网有可能被映射在多个底层网络域上。如果各个底层网络域是由不同 InP 负责运营, 则出于利益因素的考虑, 各个网络域可能不对外公布自身的拓扑状态信息, 导致无法在多个域之间建立一个中心管理节点。这就限制了集中式映射方法的应用范围。

3) 如果单个底层网络域的规模较大, 则维护网络的状态信息将需要大量的通信开销, 并加重中心管理节点的信息处理压力。

由于分布式映射算法不需要底层网络的全局状态信息, 因此在分布式映射环境中可以不用设置中心管理节点, 也就避免了集中式管理带来的问题。相对于集中式环境来说, 分布式映射环境存在以下不同。

1) 集中式环境中, 只有中心管理节点能够接收和处理虚拟网请求。而在分布式环境中, 底层网络的任意一个节点都可以接收并处理虚拟网请求。

2) 集中式环境中, 中心管理节点掌握了网络域内所有节点和链路的资源状态信息。在分布式环境

中, 每个节点只掌握本节点以及连接在本节点上所有链路的资源状态信息。若某个节点要想知道其他节点或链路的状态信息, 需要通过发送状态查询消息来实现。

3) 集中式环境中, 虚拟网的映射策略是中心管理节点根据其维护的全网资源状态信息来制定的。而分布式环境中, 虚拟网的映射策略是通过各个节点之间的相互协商来制定的。

4) 集中式环境中, 虚拟网请求是依次被处理的。但在分布式环境中, 可能有多个虚拟网请求同时被映射, 在各个并行的映射进程之间可能会产生冲突。

3 分布式映射算法

在虚拟网的分布式映射环境中, 由于没有节点来维护底层网络的全局信息, 因此虚拟网的分布式映射需要通过底层节点之间的协商和信息交换来完成。但是节点之间过多的通信不但会加重网络的负担, 还会影响映射请求的响应速度。因此, 对于分布式映射算法来说, 不但要实现底层资源的合理利用, 还要尽量降低映射时的通信开销。

针对单个域内的分布式映射问题, 本文提出一种基于协商的分布式虚拟网映射算法(N-DVE, distributed VN embedding algorithm based on negotiation), 该算法考虑了底层节点和链路的最大能力限制, 并尽量减少了分布式映射时状态信息的交换次数。映射时的状态信息获取和控制命令下发等操作都是通过分布式映射协议实现的。

3.1 分布式映射协议

分布式映射协议主要用于节点间的协商, 该协议由资源状态查询和控制命令这 2 大类消息组成。资源状态查询消息是用于底层节点之间的资源状态信息交换, 具体包括以下几方面。

1) Query(n^s): 向一个底层节点 n^s 查询其资源状态, 包括 n^s 的可用 CPU 资源、连接在 n^s 上的所有底层链路的可用带宽资源。

2) PathQuery(p): 该消息可用来查询某个底层路径 p 的最小可用带宽。

控制命令类消息主要用于在虚拟网映射过程中向节点下发各种控制命令, 以实现各个节点之间的协同映射。

3) Authorize(n^s, n^v, G^v): 将虚拟节点 n^v 的映射决策权授予底层节点 n^s 。此外, 该消息还将为映射决策提供必要的状态信息, 如映射请求的拓扑 G^v 、已

映射的节点和链路信息等。

4) $Failure(n^s, n^v)$: 当底层节点 n^s 向另一个底层节点 n^v 授权进行虚拟节点 n^v 的映射决策后, 若 n^v 无法为 n^s 确定一个满足约束条件的映射目标时, 则 n^v 要向 n^s 发送一个映射失败的反馈消息。

5) $Embed(n^s, n^v, l^v, p, backup)$: 用于下发映射执行命令, 也就是将虚拟节点 n^v 映射到底层节点 n^s 上, 或将虚拟链路 l^v 映射到底层路径 p 上, $backup$ 是冲突避免机制的备选映射方案。

6) $ASK(n^s, n^v)$: 当被授权的底层节点 n^s 成功映射虚拟节点 n^v 后, 返回一个确认消息。

7) $Start(Req, N')$: 当映射请求 Req 的所有虚拟节点和链路都已经被映射后, 由某个特定节点向被 Req 映射的底层节点集合 N' 下达虚拟网开始运行的命令。

8) $Stop(Req, N')$: 在映射请求 Req 的映射过程中, 如果无法为某个虚拟节点或链路找到合适的映射目标, 则向已被 Req 映射的底层节点集合 N' 下达撤销映射的命令, 该虚拟网请求将被拒绝。

3.2 算法流程

为了降低通信开销, 本算法将不会在全网范围内进行周期性的状态信息交换, 在虚拟网映射的过程中仅根据需要来对部分节点进行状态信息的查询, 从而最大程度地减少无用的状态信息传递。此外, 由于算法在计算多条路径时需要全网拓扑, 因此, 要求底层网络使用的路由协议能够支持每个节点都维护一个全网拓扑结构图, 如 OSPF 协议。

在映射时将被映射的第一个底层节点设为虚拟网的中心节点, 该节点将与其他节点协同完成后续的映射操作, 在映射完成后由中心节点下达虚拟网的 $Start$ 和 $Stop$ 命令。在映射时, N-DVE 算法按照与中心节点的距离将虚拟节点分为多个层次, 同一层次中的虚拟节点将会由多个底层节点并行的进行映射。

N-DVE 算法由 4 个子算法组成, 分别是: 中心节点选择算法、主流程控制算法、映射目标搜索算法和路径搜索算法。中心节点选择算法是由接收虚拟网请求的节点来执行, 主流程控制算法是由中心节点来执行, 而映射目标搜索算法和路径搜索算法则是并行地运行在多个被授权的底层节点上。

3.2.1 中心节点选择算法

在分布式映射环境中, 接收虚拟网请求的可能是某一特定节点, 也可能是底层网络中的任意一个

节点。当某节点接收到虚拟网请求后, 将调用中心节点选择算法来从底层网络中选取一个可用资源较多的底层节点作为中心节点。底层节点的可用资源既包括其自身的可用 CPU 资源, 也包括与其相连的底层链路的可用带宽, 底层节点的可用资源状态的定义为

$$\chi(n^s, n^v) = C_R(n^s) + \beta d(n^v) \sum_{l^s \in NL(n^s)} C_R(l^s) \quad (3)$$

其中, C_R 表示底层节点 n^s 或底层链路 l^s 的剩余可用资源, NL 表示连接在某个底层节点上的底层链路集合, β 为权重因子, $d(n^v)$ 表示虚拟节点 n^v 的度数。算法流程如算法 1 所示。

算法 1 中心节点选择算法

- 1) 接收虚拟网请求, 选择度数最大的虚拟节点 n^v 作为初始映射节点;
- 2) 向所有底层节点发送 Query 消息;
- 3) 将按时反馈状态信息并有足够资源承载 n^v 的底层节点加入候选集 C ;
- 4) 对每个 $n^s \in C$ 计算 $\chi(n^s, n^v)$ 值, 并按照该值对 C 中的节点排序;
- 5) 选择 χ 值最高的底层节点 n_{\min} 作为中心节点;
- 6) 发送 Authorize 消息将虚拟网的映射控制权交给 n_{\min} 。

3.2.2 主流程控制算法

当中心节点选择算法运行完以后, 虚拟网的映射控制权就移交给运行在中心节点上的主流程控制算法。该算法主要负责分发虚拟节点的映射授权。首先, 将 G^v 中度数最大的虚拟节点 n^v 映射到中心节点上, 并按照与 n^v 的最短路径长度将剩余的虚拟节点划分为多个层次 (如与 n^v 距离为 i 跳的虚拟节点设为第 i 层, n^v 设为第 0 层), 然后, 将按照层次由低到高的次序依次对虚拟节点的映射控制权进行分发。当第 i 层的虚拟节点全部映射完毕后, 中心节点将会把第 $i+1$ 层的虚拟节点 n^v 的控制权交给与其相连的第 i 层虚拟节点, 如果 n^v 与多个第 i 层节点相连, 则依次检查这些节点对之间的虚拟链路, 并将控制权交给带宽需求最大的虚拟链路的端节点, 这样就能使占用带宽较多的虚拟链路更有可能被映射到较短的底层路径上, 从而减少虚拟链路占用的带宽资源。算法流程如算法 2 所示。

算法 2 主流程控制算法

输入: 映射请求的拓扑 G^v ;

- 1) 将 G^v 中度数最大的 n^v 映射在中心节点上;

- 2) 以 n^v 为起点, 按广度优先的次序遍历 G^v , 根据与 n^v 的最短路径长度将虚拟节点划分为多层;
- 3) for $i=1$ to z do // z 为最大层数
- 4) for all $n_j^v \in N_i$ do // N_i 为第 i 层的虚拟节点集合
- 5) 在 $i-1$ 层中查找与 n_j^v 相连的虚拟节点;
- 6) if 存在多个与 n_j^v 相连的虚拟节点 then
- 7) 检查各个虚拟节点与 n_j^v 之间虚拟链路的带宽需求量, 将带宽需求最大的虚拟链路的端节点作为授权节点;
- 8) else
- 9) 将唯一与 n_j^v 相连的节点作为授权节点;
- 10) end if
- 11) end for
- 12) 向被授权的底层节点发送 Authorize 消息;
- 13) if 收到授权对象节点返回的 Failure 消息 then
- 14) 向所有已映射节点发送 Stop 命令, 映射失败;
- 15) end if
- 16) end for
- 17) 向所有已映射节点发送 Start 命令, 虚拟网开始运行

3.2.3 映射目标搜索算法

收到中心节点映射授权的底层节点称为执行节点, 执行节点将会运行映射目标搜索算法。该算法从授权消息中获取待映射的虚拟节点, 并为这些虚拟节点搜索合适的映射目标, 同时还要完成部分虚拟链路的映射。

算法首先要为待映射的虚拟节点搜索符合资源约束的映射目标, 为了缩短虚拟链路的路径长度, 并减少信息收集的时间消耗, 可以将搜索的范围限定在执行节点周围 h 跳以内。如果底层网络采用的路由协议支持区域划分 (如 OSPF 协议中的 Area 机制), 还可以把映射目标的搜索范围限定在执行节点所在的区域或是其邻接区域内, 从而进一步降低搜索的时间和虚拟链路的路径长度。在搜索到多个可用映射目标以后, 需要对每个映射目标进行评价并选出最优目标。本文用映射评价系数 μ 来对映射目标进行评价, 其定义为

$$\mu(n^s, n^v, l^v) = \frac{\chi(n^s, n^v)}{R(l^v)L_p(l^v)} \quad (4)$$

其中, n^v 是待映射的虚拟节点, n^s 是 n^v 的可用映射

目标, l^v 是连结 n^v 与执行节点的虚拟链路, $R(l^v)$ 是 l^v 的带宽需求约束, L_p 是 l^v 的底层路径长度。在对各个可用映射目标按 μ 值进行排序后, 就可以选出最优的映射目标, 并将待映射虚拟节点映射到该目标上。算法流程如算法 3 所示。

算法 3 映射目标搜索算法

输入: 映射请求的拓扑 G^v , 待映射的虚拟节点集合 N^v , 最大跳数 h ;

输出: ASK 消息或 Failure 消息

- 1) if 执行节点与已映射虚拟节点之间存在尚未映射的虚拟链路 then
- 2) 调用路径搜索算法, 将虚拟链路映射到该算法计算出的路径上
- 3) end if
- 4) 将执行节点周围 h 跳以内的底层节点加入集合 C , 并向其发送 Query 消息;
- 5) 将 N^v 中的虚拟节点按照资源需求量进行降序排序
- 6) for all $n_i^v \in N^v$ do
- 7) for all $n_j^s \in C$ do
- 8) if n_j^s 有足够的可用资源承载 n_i^v then
- 9) 调用路径搜索算法, 搜索执行节点到 n_j^s 的路径最佳路径 p ;
- 10) if p 不为空 then
- 11) 计算 μ , 将 n_j^s 加入候选集 c_i ;
- 12) end if
- 13) end for
- 14) end for
- 15) if c_i 不为空 then
- 16) 将 c_i 中的节点按 μ 值进行排序, 通过 Embed 命令把 n_i^v 映射到 μ 最大的 n_j^s 上, 并将执行节点与 n_i^v 的虚拟链路映射到 p 上;
- 17) 向中心节点发送 ASK 消息;
- 18) else
- 19) 向中心节点发送 Failure 消息;
- 20) end for

3.2.4 路径搜索算法

路径搜索算法用于在 2 个底层节点之间寻找一条满足带宽约束的路径。其流程是先调用 K 短路径算法^[14]计算出多条路径, 然后通过发送 PathQuery 消息来检查每一条路径的最小可用带宽, 最后挑选长度最短的可用路径作为算法的输出结果。

3.3 冲突避免机制

在分布式的虚拟网映射环境中,有可能在同一时刻存在多个正在被处理的映射请求,且每个映射请求是由不同的中心节点进行处理。而 N-DVE 算法是通过查询响应机制来获取网络的资源状态,从查询到映射命令下达之间会存在一定的时间差,这就使多个虚拟网在并行映射时,可能由于彼此之间缺乏协商而引发冲突。这种冲突会导致部分虚拟网的映射流程出现错误,比如若某个虚拟网通过向一个底层节点 n^s 发送 Query 消息获知 n^s 能够满足自身的映射要求,但在 n^s 收到其发出的 Embed 命令之前,另一个虚拟网通过发送 Embed 命令将自己的虚拟节点映射在 n^s 上,这就可能导致 n^s 在收到前一个虚拟网的 Embed 命令时已经没有足够的资源满足该虚拟网的映射要求,从而使映射流程出现错误。

针对该问题, N-DVE 设计了一种冲突避免机制。在执行节点向虚拟节点 n^v 的映射目标发送 Embed 命令时(映射目标搜索算法的步骤 16)),会选择的一个次优的映射目标作为 n^v 的备选映射方案加入到 Embed 命令里。在映射目标节点收到 Embed 命令后,首先,检查自身的资源状态是否依然能够满足 n^v 的映射要求,若不能满足则表明可能出现了映射冲突现象,这时该节点将会直接把 Embed 命令转发给备选映射方案所指定的映射目标,由备选底层节点完成 n^v 的映射。这样就可以使 2 个映射进程在某个底层节点上发生冲突时,其中一个进程可以通过执行备选映射方案来回避在该节点上的映射冲突,从而避免由此带来的死锁或映射失败等问题。

4 仿真结果分析

4.1 实验环境

本实验在 Pentium 4 CPU 3.2 GHz、1 GB 内存的 PC 机上运行。底层网络拓扑和虚拟网请求的拓扑由 GT-ITM^[15]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 0.25,虚拟节点的资源取值在[0,20],虚拟链路的带宽取值在[0,35]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,持续时间服从参数为 1 000 的指数分布。

实验将选取一种分布式算法 DVNMA^[13]作为

N-DVE 算法的比较对象,DVNMA 的状态通告周期为 5 个时间单位。DVNMA 与 N-DVE 都属于基于贪婪思想的映射算法,因此为了在虚拟链路的路径长度和接收率等方面进行进一步比较,实验还选取了一种基于贪婪思想的集中式算法 G-SP^[5]作为 N-DVE 算法的比较对象。G-SP 在集中式环境下运行,不启用路径分裂和重映射机制。N-DVE 算法的参数设置为 $k=3$, $\beta=0.1$ 。每处理 100 个虚拟网请求记录一次实验数据。

4.2 参数影响

在映射目标搜索算法中,执行节点在为待映射的节点搜索映射目标时,为了减少通信开销和响应时间,可以通过设置参数 h 来限制映射目标与执行节点之间的距离。 h 值越小则意味着映射目标搜索的范围越小,节点映射失败的概率会相应的增加,但由于降低了虚拟链路的带宽资源占用,所以降低了链路映射失败的概率。因此可以根据节点和链路的资源状态来对参数 h 进行设置,节点资源紧张时 h 可以设置得较大,链路资源紧张时可以将 h 设置得较小。

如图 2 所示,随着 h 值的增加, N-DVE 搜索映射目标的范围也越大,所以映射一个虚拟网所需的平均资源查询消息数量也就越多。同时,搜索范围的增大会导致路径搜索算法的调用次数增加,因此若 h 值增大,则算法的响应时间也会增长(如图 2 所示)。在后续实验中, h 的值设为 3。

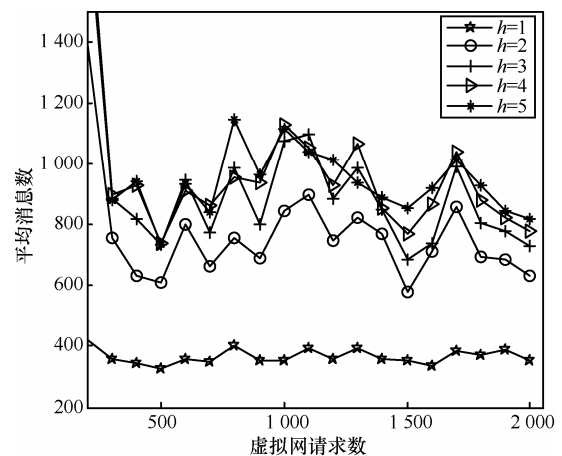


图 2 N-DVE 映射虚拟网时所需的平均消息数

4.3 通信开销

N-DVE 是通过小范围内的查询响应机制来获取网络的状态信息,而 DVNMA 则是通过在全网范围内进行周期性的信息交换来获取网络的状态信

息。因此，与 DVNMA 相比，N-DVE 映射时所花费的通信开销更少。实验统计了 N-DVE 算法和 DVNMA 算法映射一个虚拟网所需的平均消息数量，由图 4 可知，N-DVE 算法的通信开销仅为 DVNMA 算法的 20% 左右。

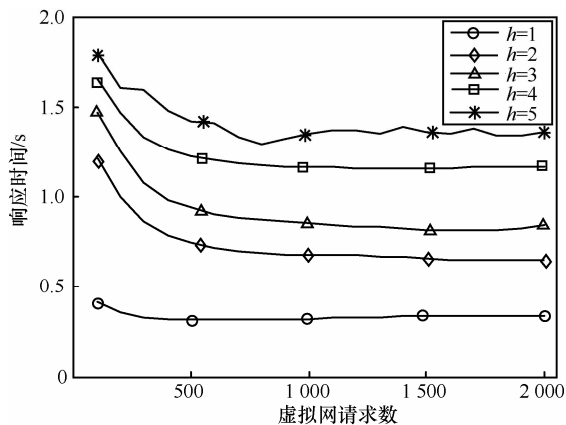


图 3 N-DVE 算法的平均响应时间

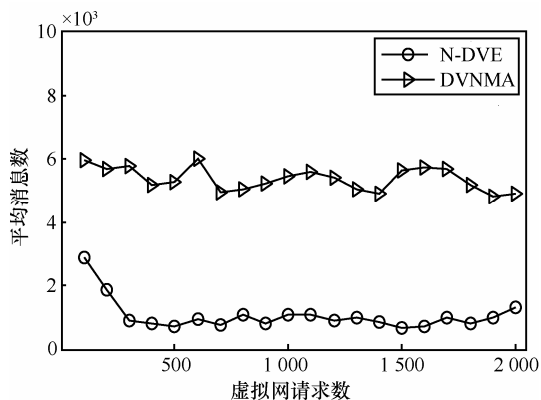


图 4 通信开销

4.4 负载均衡

为了合理利用底层网络的资源，应将负载均衡情况作为虚拟网映射算法的评价标准之一。本实验用平均负载方差来衡量算法的负载均匀程度，定义为

$$D = \frac{1}{2} \frac{1}{|N^s|} \sum_{n_i^s \in N^s} |S(n_i^s) - S_{arg}(N^s)| + \frac{1}{2} \frac{1}{|L^s|} \sum_{l_i^s \in L^s} |S(l_i^s) - S_{arg}(L^s)| \quad (5)$$

其中， N^s 、 L^s 分别是底层节点和链路的集合， S 表示某个底层节点或链路的负载压力，也就是已分配资源在该节点或链路的资源总量中占有的比例， S_{arg} 表示底层网络中全部节点或链路的平均负载压力。由式(5)可知，平均负载方差同时评价了底层网络中的节点和链路的负载均衡情况，该值越小表示评价

越好。如图 5 所示，N-DVE 的平均负载方差优于 DVNMA，与 G-SP 处于同一水平。

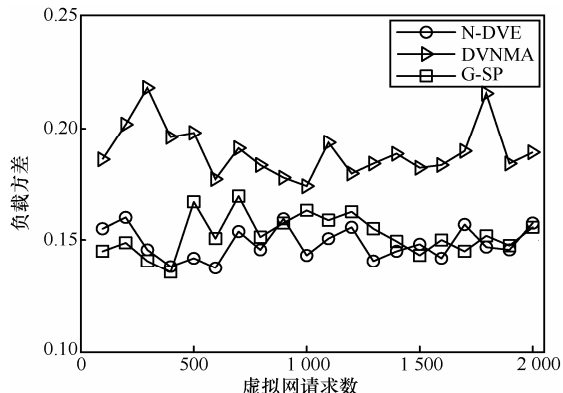


图 5 平均负载方差

4.5 虚拟链路的路径长度

虚拟链路的路径长度主要是用来衡量底层链路带宽资源的利用效率，较短的虚拟链路路径长度意味着虚拟网占用的底层链路带宽资源较少，从而使底层网络能承载更多的虚拟网。在 N-DVE 算法中，当执行节点为每个待映射节点选择映射目标时，都会考虑该目标与执行节点的距离，使其在负载均衡的前提下尽量缩短虚拟链路的路径长度。而 DVNMA 算法在映射时将虚拟网拓扑拆分为多个星形子拓扑，但在映射这些星形子拓扑时只考虑了子拓扑内部节点之间的距离，而没有考虑各个子拓扑之间的距离，使其虚拟链路的平均路径长度要高于 N-DVE (如图 6 所示)。G-SP 算法由于是一种两阶段映射算法，节点映射和链路映射是分开进行的，因此算法在映射虚拟节点时没有考虑缩短虚拟链路的长度，导致其虚拟链路的平均路径长度最长。

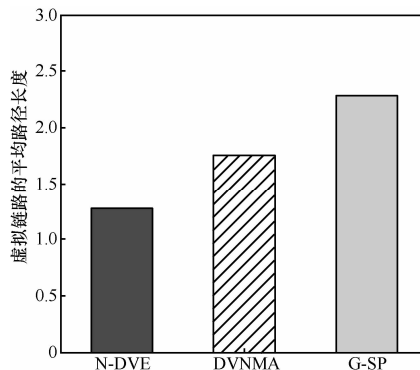


图 6 虚拟链路的平均路径长度

4.6 接收率

接收率决定了 InP 的收益，接收率的高低与负

载均衡程度和虚拟链路长度有很大的关系。由于 N-DVE 算法在上述 2 个评价标准中都有较好的表现, 因此其接收率要高于 DVNMA 算法和 G-SP 算法如图 7 所示。

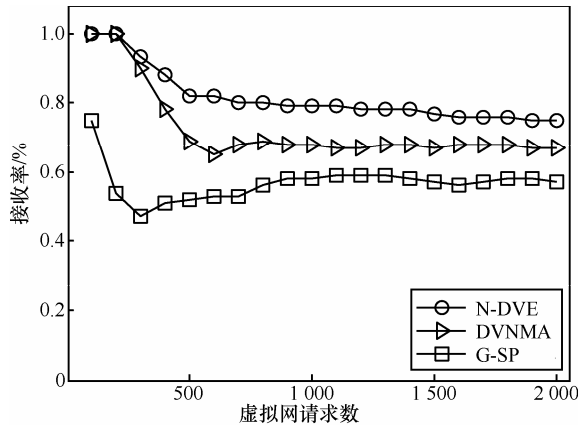


图 7 接收率

5 结束语

本文研究了分布式环境下的虚拟网映射算法。首先, 分析了集中式映射与分布式映射之间的差别, 阐述了分布式映射的特点和应用场景。针对已有的分布式算法通信开销大、虚拟链路占用资源过多的缺点, 设计了一种基于协商的分布式虚拟网映射算法, 该算法仅在小范围内进行状态信息的交换, 并且在降低映射代价方面进行了优化设计。此外, 为了支持并行处理能力, 算法还加入了冲突避免机制。实验证明, 本算法只需要以较小的通信代价就能在各项指标上获得较好的评价。

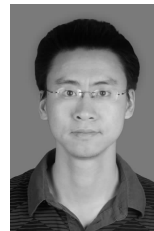
通过路径分裂技术能够提高虚拟链路的映射成功率, 但也为虚拟网的控制和管理增加了一定的难度, 因此在分布式虚拟网映射算法中如何引入路径分裂机制有待进一步研究。此外, 本文算法只讨论了单个域内的映射问题, 而在涉及多个域的虚拟网映射中, 如何协同利用分布式与集中式的映射算法也是有待探索的工作。

参考文献:

- [1] CHOWDHURY M, BOUTABA R. A survey of network virtualization[J]. *Computer Networks*, 2010, 54(5): 862-876.
- [2] JORGE C, JAVIER J. Network virtualization—a view from the bottom[A]. *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*[C]. Barcelona, Spain, 2009.73-80.
- [3] 程祥, 张忠宝, 苏森等. 虚拟网络映射问题研究综述[J]. *通信学报*, 2011, 32(10): 143-151.

- [4] BIAO S, MOHAMMAD H, EUI-NAM H. Delivering IPTV service over a virtual network: a study on virtual network topology[J]. *Journal of Communications and Networks*, 2012, 14(3): 319-335.
- [5] YU M, YI Y, REXFORD J. Rethinking virtual network embedding: substrate support for path splitting and migration[J]. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2): 17-29.
- [6] CHOWDHURY M, RAHMAN MR, BOUTABA R. ViNeyard: virtual network embedding algorithms with coordinated node and link mapping[J]. *IEEE Transactions on Networking*, 2012, 20(1): 206-219.
- [7] ZHANG M, YANG Q, WU C, *et al.* Hierarchical virtual network mapping algorithm for large-scale network virtualisation[J]. *IET communications*, 2012, 6 (13): 1969-1978.
- [8] FAJJARI I, AITSAADI N, PUJOLLE G, *et al.* VNE-AC: Virtual network embedding algorithm based on ant colony metaheuristic[A]. *Proc IEEE ICC 2011*[C]. Kyoto, Japan, 2011.1-6.
- [9] ZHANG Z, CHENG X, SU S, *et al.* A unified enhanced particle swarm optimization-based virtual network embedding algorithm[J]. *International Journal of Communication Systems*, 2012, doi: 10.1002/dac.1399.
- [10] YU H F, QIAO C M, ANAND V, *et al.* Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures[A]. *Proc IEEE GLOBECOM 2010*[C]. Miami, USA, 2010.1-6.
- [11] CAI Z, LIU F, XIAO N, *et al.* Virtual network embedding for evolving networks[A]. *Proc IEEE GLOBECOM 2010*[C]. Miami, USA, 2010.1-5.
- [12] FAJJARI I, AITSAADI N, PUJOLLE G, *et al.* VNR algorithm: a greedy approach for virtual networks reconfigurations[A]. *Proc IEEE GLOBECOM 2011*[C]. Houston, USA, 2011.1-5.
- [13] HOUIDI I, LOUATI W, ZEGHLACHE D. A distributed virtual network mapping algorithm[A]. *Proc IEEE ICC*[C]. Beijing, China, 2008.5634-5640.
- [14] EPPATEIN D. Finding the k shortest paths[A]. *IEEE Symposium on Foundations of Computer Science*[C]. SantaFe, NM, 1994. 154-165.
- [15] ZEGURA E, CALVERT K, BHATTACHARJEE S. How to model an internet[A]. *Proc IEEE INFOCOM*[C]. San Francisco, USA, 1996.594-602.

作者简介:



江逸茗 (1984-), 男, 江苏南通人, 国家数字交换系统工程技术研究中心助理研究员, 主要研究方向为网络虚拟化和宽带信息网络。

兰巨龙 (1962-), 男, 河北张北人, 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。

程东年 (1957-), 男, 河南原阳人, 国家数字交换系统工程技术研究中心教授、硕士生导师, 主要研究方向为新一代信息网络关键理论与技术的研究工作。

吴方明 (1985-), 男, 河南郑州人, 吉林大学硕士生, 主要研究方向为网络与信息安全、密码学、新型网络体系架构。