

一种双网同步搜索的虚拟网络映射算法

彭利民

(华南理工大学自动化科学与工程学院 广州 510006)

摘要 针对虚拟网络映射中的资源分配问题,通过建立双网同步搜索映射模型,采用普里姆最小生成树算法思想,同步搜索虚拟网络中的待映射虚拟节点和物理网络中的可映射物理节点,将相邻的虚拟节点依次映射到邻接的物理节点上,协调完成节点及其邻接链路的映射操作,使虚拟网络映射具有拓扑一致性。仿真实验表明,提出的 DS-VNM 算法能有效地降低虚拟链路的映射路径长度,提高网络收益与网络代价比、虚拟网络请求接受率,获得了较好的资源分配性能。

关键词 虚拟网络映射,双网搜索,资源分配

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.7.030

Virtual Network Mapping Algorithm of Searching Virtual and Substrate Network Synchronously

PENG Li-min

(School of Automation Science and Engineering, South China University of Technology, Guangzhou 510006, China)

Abstract Aiming at the resource allocation problem in virtual network mapping, a mapping model of searching virtual network and substrate network synchronously was proposed in this paper. By using the principle of the prim minimum spanning tree algorithm, the virtual node waiting for being mapped in the virtual network and the right substrate node satisfying the current virtual node in the substrate network are searched simultaneously, so that adjacent nodes in the virtual network are mapped into adjacent nodes in the substrate network in a proper order, at the same time the virtual nodes and their adjacent links are mapped in a harmony way, resulting in that topological property of the virtual network is also kept in the substrate network after being mapped. Simulation results show that the DS-VNM algorithm can reduce the mapping path length of virtual links effectively, and improve ratio of the network revenue and network cost and the acceptance ratio of the virtual network requests, and achieve good performance while allocating network resource.

Keywords Virtual network mapping, Double networks search, Resource allocation

1 引言

网络虚拟化技术是解决当前互联网僵化问题的有效手段。Internet 服务提供商利用网络虚拟化技术,可以在同一物理网络上创建多个虚拟网络,从而为 Internet 用户提供多样化的网络服务^[1]。虚拟网络映射是指在满足虚拟网络资源需求的条件下,将虚拟网络请求嵌入到底层物理网络之上。当两个图同构时,两个图的顶点之间具有保持相邻关系的一一对应,因此虚拟网络映射本质上即为子图同构搜索问题,但目前判断一个图是另一个图的同构子图仍然是一个 NP 难问题^[2]。由于虚拟网络映射须满足虚拟网络请求的资源约束条件,如虚拟节点的计算资源和虚拟链路的带宽资源等约束,特别是虚拟网络的拓扑结构具有多样性、虚拟网络请求具有动态性等,因此虚拟网络映射成为当前网络虚拟化技术中极具挑战性的研究课题。

近几年,学者们提出了一些启发式算法来求解虚拟网络映射问题,它们可分为两类,第一类算法是在一个阶段内完成节点和链路的映射操作^[3-5],该类算法在映射过程中考虑了虚

拟节点映射和虚拟链路映射之间的耦合关系;另一类是两阶段映射算法,即首先利用启发式算法映射节点,然后利用最短路径或 K 最短路径算法映射虚拟链路^[6-7]。文献[3]利用同构子图搜索算法,在一阶段内完成节点和链路的映射操作,但该算法的时间复杂度太高,不适合处理大规模的虚拟网络映射问题。在文献[3]的基础上,文献[4]提出了节点拓扑属性的资源评价方法,同步实施节点和链路映射操作,提高了虚拟网络映射的资源分配性能,但该算法假设物理网络资源总是可用的,因此不适合处理动态的虚拟网络映射请求。文献[5]将虚拟网络映射中的资源分配简化为整数规划问题,协调解决节点映射和链路映射,但该算法使用的整数规划松弛算法破坏了节点和链路固有的关联性,使该算法的资源分配性能受到影响;文献[6]首先使用贪婪算法将资源需求较大的虚拟节点映射到可用资源较多的物理节点上,然后采用 K 最短路径算法^[7]映射虚拟链路,但该映射算法在虚拟网络映射过程中没有考虑网络拓扑属性,资源分配性能有待提高;文献[8]通过建立节点映射模型,将虚拟网络映射到物理网络中某个局部区域内,虽然该方法可降低虚拟网络的资源消耗,但由于

到稿日期:2014-07-08 返修日期:2014-10-27 本文受广东省自然科学基金项目(S2012040007599)资助。

彭利民(1976-),男,博士后,副教授,主要研究方向为网络虚拟化、分布式计算等,E-mail: penglm86@126.com.

其在映射过程中没有考虑网络节点间的位置关系,相邻的虚拟节点被映射后在物理网络中仍然无法保持邻接关系,从而增加了虚拟网络映射的资源消耗。

综上所述,现有的虚拟网络映射算法主要存在以下问题:

(1)虚拟网络中相邻的节点被映射后在物理网络中无法保持邻接关系,增加了虚拟网络映射的资源开销;(2)基于同构子图搜索或整数规划的一阶段虚拟网络映射算法的时间复杂度较高,不适合处理在线的虚拟网络映射请求。本文基于网络拓扑一致性原则,建立双网同步搜索映射模型,在虚拟网络和底层物理网络中同步搜索虚拟节点和物理映射节点,将相邻的虚拟节点映射到邻接的物理顶点上,并协调映射节点及其邻接链路,使虚拟网络被映射后尽可能地保持原来的网络拓扑结构,降低虚拟链路的映射路径长度。仿真实验表明,DS-VNM算法获得了较优的资源分配性能。

2 虚拟网络映射模型

2.1 问题描述

虚拟网络映射是指将虚拟网络请求部署到底层的物理网络之上,并为虚拟节点和虚拟链路分配满足其资源约束的网络资源。

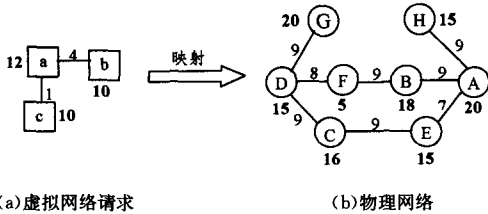


图1 虚拟网络映射示例

定义1(物理网络, Substrate Networks) 物理网络采用无向带权图 $G_s = (N_s, L_s, A_m, A_b)$ 表示,其中 N_s 表示物理节点集合, L_s 表示物理链路集合, A_m 表示节点的计算、存储等属性, A_b 表示链路的带宽、传输时延等属性。不失一般性,文中考虑节点的CPU属性和链路的带宽属性。图1(b)中节点旁的粗体数字表示节点的CPU资源量,链路旁的数字表示链路的带宽资源量。

定义2(虚拟网络, Virtual Networks) 类似地,采用无向带权图 $G_v = (N_v, L_v, A_m, A_b)$ 表示虚拟网络,其中 N_v 是虚拟节点集合, L_v 是虚拟链路集合, A_m 表示虚拟节点的资源约束条件, A_b 表示虚拟链路的资源约束条件。图1(a)中节点旁的粗体数字为节点的CPU资源需求量,链路旁的数字为链路的带宽资源需求量。

定义3(虚拟网络映射, Virtual Network Mapping) 虚拟网络映射是指将虚拟网络中的节点和链路映射到物理网络上,并满足虚拟节点和虚拟链路的资源约束条件。它包含两个基本操作:(1)将虚拟节点映射到满足节点资源约束的物理节点上,且同一个虚拟网络中的节点不能映射到两个不同的物理节点上;(2)将虚拟链路映射到满足链路带宽约束的物理链路或路径上,且该路径不能构成环路。图1中,虚拟节点 a, b 和 c 分别映射到物理节点 A, B 和 C 上,虚拟链路 (a, b) 和 (a, c) 映射到物理链路 (A, B) 和物理路径 (A, E, C) 上,此时虚拟链路 (a, b) 的映射路径长度为1,虚拟链路 (a, c) 的映射路径长度为2。

2.2 优化目标

虚拟网络映射算法的主要目标是在满足虚拟网络资源约束的条件下,使物理网络中的网络资源得到有效地利用。

定义4(网络收益与网络代价比) 在时刻 t ,物理网络接受一个虚拟网络请求的网络收益为虚拟网络中的节点CPU需求量和链路的带宽需求量总和。

$$R(G_v, t) = \sum_{n_v \in N_v} CPU(n_v) + \sum_{l_v \in L_v} BW(l_v) \quad (1)$$

类似地,物理网络接受一个虚拟网络请求的网络代价指为该虚拟网络请求分配的资源量总和。

$$C(G_v, t) = \sum_{n_v \in N_v} CPU(n_v) + \sum_{l_v \in L_v} \sum_{l_s \in L_s} BW(f_{l_s}^{l_v}, l_s) \quad (2)$$

式中,当虚拟链路 l_v 映射到物理链路 l_s 上且为 l_v 分配所需带宽资源时, $f_{l_s}^{l_v}$ 为1;否则 $f_{l_s}^{l_v}$ 为0。 $BW(f_{l_s}^{l_v}, l_s)$ 是指为虚拟链路 l_v 分配的带宽资源量。时间 T 内,虚拟网络映射算法的网络收益与网络代价比定义为:

$$R/C = \frac{\sum_{t=0}^T R(G_v, t)}{\sum_{t=0}^T C(G_v, t)} \quad (3)$$

定义5(虚拟网络请求接受率) 虚拟网络请求接受率是指时间 T 内已成功映射的虚拟网络个数与虚拟网络请求总数之比,它可以定义为:

$$RVQ = \frac{\sum_{t=0}^T VN_{recv}}{\sum_{t=0}^T VN_{requ}} \quad (4)$$

定义6(虚拟链路扩张因子) 虚拟链路的映射路径长度越长,虚拟链路占用的链路带宽资源越多,因此虚拟链路的映射路径长度可以反映虚拟网络映射算法的资源分配性能。虚拟链路扩张因子定义为

$$VLSF = \frac{1}{|L_v|} \sum_{n_i, n_j \in N_v} hops(M(n_i), M(n_j)) \cdot vlink(n_i, n_j) \quad (5)$$

式中, $|L_v|$ 为虚拟网络中链路的个数, $M(n_i)$ 和 $M(n_j)$ 分别为虚拟节点 n_i 和 n_j 在物理网络上的映射物理节点, $hops(M(n_i), M(n_j))$ 表示物理网络中节点 $M(n_i)$ 和 $M(n_j)$ 之间的路径长度,文中使用节点间跳步数表示节点之间的路径长度; $vlink(n_i, n_j)$ 为二元变量,当虚拟节点 n_i 和 n_j 相邻时, $vlink(n_i, n_j)$ 为1,否则为0。

3 虚拟网络映射算法

从Internet服务提供商的角度来看,他们总希望在满足虚拟网络资源约束条件下,尽可能地降低虚拟网络映射的资源消耗,以接受更多的虚拟网络请求,从而获得更大的网络收益。

3.1 双网同步搜索映射模型

文中通过综合考虑虚拟网络和物理网络的拓扑属性,建立双网同步搜索映射模型,使虚拟网络映射具有拓扑一致性特征。双网同步搜索映射模型可定义为:

$$NMIF(v_k, v_s) = \frac{ebw(M(v_i), v_s) - ebw(v_i, v_k)}{hops(M(v_i), v_s) \cdot vlink(v_i, v_k)} \quad (6)$$

式中, v_k 和 v_s 分别为虚拟网络和物理网络中的网络节点, v_i 为虚拟网络中已映射虚拟节点, $M(v_i)$ 表示 v_i 在物理网络中的映射物理节点, $ebw(M(v_i), v_s)$ 表示物理网络中节点 $M(v_i)$ 与 v_s 之间最短路径上的链路带宽量, $ebw(v_i, v_k)$ 表示虚拟链

路 $vlink(v_i, v_k)$ 的带宽需求量; $hop_s(M(v_i), v_s)$ 表示物理网络中节点 $M(v_i)$ 与 v_s 之间的路径长度, $vlink(v_i, v_k)$ 为二元变量, 当节点 v_i 和 v_k 在虚拟网络上相邻时, 则 $vlink(v_i, v_k)$ 为 1, 否则为 0;

3.2 虚拟网络映射算法

DS-VNM 算法主要解决当前虚拟网络映射过程中虚拟链路的映射路径长度过大以及节点、链路映射操作不协调等问题。DS-VNM 算法的核心思想可归纳为: (1) 利用双网同步搜索映射模型以及采用普里姆最小生成树算法思想, 在虚拟网络和物理网络上同步搜索最合适的待映射虚拟节点和可映射物理节点, 以便将相邻的虚拟节点映射到邻接的物理顶点之上, 使虚拟网络被映射后在物理网络上仍然保持原有的网络拓扑结构, 降低虚拟网络映射的资源消耗; (2) 将虚拟网络视为一个整体, 同步映射节点及其邻接链路, 协调完成节点和链路的映射操作, 提高物理网络资源的利用率。

算法 1 虚拟网络映射算法 (DS-VNM)

Procedure of DS-VNM algorithm

1. $G_v = \phi, G_s = \phi$;
2. map virtual node v_j with maximum degree and CPU request to the substrate node v_l which meets virtual node v_j CPU requirement;
3. add v_j and v_l to the set G_v, G_s respectively;
4. while ($|G_v| \leq n$)
5. compute each nodes' NMIF in neighbor nodes of these mapped nodes in the set G_v ;
6. choose the max NMIF(v_k, v_s);
7. If ($cpu(v_s) > cpu(v_k)$ & $bw(M(v_i), v_s) > bw(v_i, v_k)$)
8. map virtual node v_k to substrate node v_s ;
9. map the virtual link (v_i, v_k) on the substrate link ($M(v_i), v_s$);
10. add v_k and v_s to the set G_v, G_s respectively;
11. end if
12. If there exists virtual links between node v_k and these nodes v_c in set G_v and virtual link (v_k, v_c) is not mapped, then map the virtual link (v_k, v_c) using the k-shortest path algorithm;
13. update nodes and links resource information in the substrate network;
14. end if
15. end while
16. end procedure

DS-VNM 算法中, G_v 表示已完成映射的虚拟节点集合, G_s 表示参与了本次映射的物理节点集合。DS-VNM 算法逐个计算已映射节点集 G_v 的每个邻居节点的 NMIF 值, 并选择 $NMIF(v_k, v_s)$ 值最大的节点对 (v_k, v_s) 参与映射, 当节点 v_s 的 CPU 资源量大于虚拟节点 v_k 的资源需求量, 且物理链路 ($M(v_i), v_s$) 的带宽量大于虚拟链路 (v_i, v_k) 的带宽需求时, 则将节点 v_k 和链路 (v_i, v_k) 分别到映射到 v_s 和物理链路 ($M(v_i), v_s$) 上, 并将节点 v_k, v_s 分别加入集合 G_v, G_s 中。当虚拟节点 v_k 与 G_v 中的节点存在虚拟链路且还未被映射时, 则利用 K-最短路径算法^[7] 映射这些虚拟链路。DS-VNM 算法的时间复杂度主要表现在计算 $NMIF(v_k, v_s)$ 值上, 当虚拟网络中节点个数为 n , 物理网络中节点个数为 m 时, NMIF 的时间复杂度为 $O(n^2 \cdot m)$, DS-VNM 算法的时间复杂度为 $O(n^3 \cdot m)$ 。

4 仿真实验

为了验证 DS-VNM 算法的资源分配性能, 建立仿真实验

环境, 采用 2.2 节给出的优化指标对 DS-VNM 算法进行了性能测试。为了便于比较, 文中同时实现了文献[6]的贪婪节点映射算法和最短路径链路映射算法, 以及文献[8]的虚拟网络映射算法。将文献[6, 8]中的映射算法分别表示为 G-VNM 和 TA-VNM。

4.1 实验设置

实验采用 GT-ITM 工具^[9] 生成物理网络和虚拟网络拓扑结构。物理网络由 100 个节点和大约 560 条链路构成, 物理节点的 CPU 资源量和链路带宽量在 [50, 100] 区间内均匀分布。每个虚拟网络的节点个数 n 在 [2, 10] 均匀分布, 虚拟节点之间的连接概率约为 50%。虚拟节点的 CPU 需求量和虚拟链路带宽需求量均匀分布在 [2, 20] 区间内, 虚拟网络请求的到达过程服从泊松过程, 每秒平均到达 5 个虚拟网络, 虚拟网络的生存期服从指数分布, 其平均生存期为 10s。每次仿真时间为 10min, 共进行了 10 次仿真实验, 实验最终结果为 10 次仿真实验的平均值。

4.2 实验结果

虚拟链路扩张因子表示虚拟链路的映射路径长度的平均值。从图 2 可以看出, DS-VNM 算法的虚拟链路扩张因子比 TA-VNM 算法降低了 15%, 比 G-VNM 算法降低了 38%。主要原因是 G-VNM 算法在虚拟网络映射过程没有考虑节点之间的位置关系, 虚拟节点被映射后在物理网络中相距较远, 因此虚拟链路的映射路径长度较大。虽然 TA-VNM 算法尽量将虚拟网络映射到物理网络中的局部区域内, 但相邻的虚拟节点被映射后无法保持邻接关系, 因此虚拟链路扩张因子仍然较大。DS-VNM 算法在映射过程中充分考虑了虚拟网络和物理网络中节点间的邻接关系, 相邻的虚拟节点被映射后仍然能保持邻接关系, 因此 DS-VNM 算法的虚拟链路扩张因子最小。

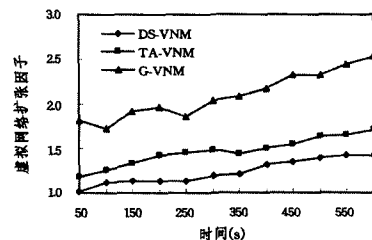


图 2 虚拟链路扩张因子

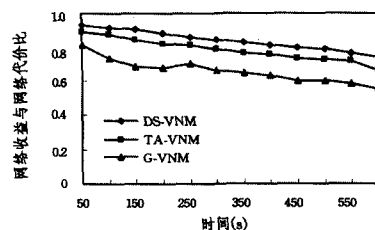


图 3 网络收益与网络代价比

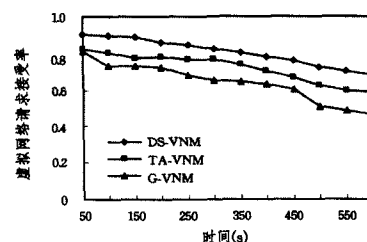


图 4 虚拟网络请求接受率

网络收益与网络代价比表示虚拟网络的资源需求量与虚拟网络映射的资源消耗量比。从图 3 中可以看出,DS-VNM 算法的网络收益与网络代价比相对较大,G-VNM 算法的网络收益与网络代价比相对较小。其主要原因是 G-VNM 算法在虚拟网络映射过程中没有考虑节点间的位置关系,虚拟链路的映射路径长度较大,因此虚拟网络占用的链路带宽资源较多,虚拟网络映射的网络代价较大。虚拟网络请求接受率可以反应物理网络资源的利用率水平,从图 4 可以看出,DS-VNM 算法的虚拟网络请求接受率与 TA-VNM 算法和 G-VNM 算法相比,分别提高了 10% 和 19%。其主要原因是 DS-VNM 算法在虚拟网络映射过程中充分考虑了节点间的位置关系,虚拟网络被映射到物理网络中能很好地保持原有的拓扑关系,降低了虚拟网络映射的资源消耗,从而为后续的虚拟网络请求提供了更多的网络资源量,因此 DS-VNM 算法的虚拟网络请求接受率相对较高。

结束语 在现有研究工作的基础上,文中通过建立双网搜索映射模型,同步搜索虚拟网络和物理网络中映射节点,协调映射节点以及其邻接链路,将虚拟网络中相邻的虚拟节点依次映射到物理网络中邻接的物理节点之上,使虚拟网络被映射后仍然能很好地保持原有的网络拓扑结构。仿真实验表明,DS-VNM 算法有效地降低了虚拟链路的映射路径长度和虚拟网络映射的网络代价,显著地提高了网络收益与网络代价比和虚拟网络请求接受率;文中提出的 DS-VNM 算法是可行的、有效的。

参 考 文 献

- [1] Fischer A, Botero J F, Beck M T, et al. Virtual network embedding: A survey[J]. IEEE communication surveys and tutorials, 2013, 15(4): 1888-1906
- [2] Cordella L P, Foggia P, Sansone C, et al. A (sub) graph isomorphism algorithm for matching large graphs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(4): 1367-1382
- [3] Lischka J, Karl H. A virtual network mapping algorithm based on subgraph isomorphism detecting[C]// Proceeding of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures, Barcelona, Spain, 2009; 81-88
- [4] 魏晓辉, 邹磊, 李洪亮. 基于优化的同构子图搜索的虚拟网络映射算法[J]. 吉林大学学报, 2013, 43(1): 165-171
Wei Xiao-hui, Zhou Lei, Li Hong-Liang. Virtual network embedding algorithm based on improved sub-graph isomorphism search[J]. Journal of Jinlin University, 2013, 43(1): 165-171
- [5] Chowdhury M, Rahman M R, Boutaba R. ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping[J]. IEEE/ACM Transactions on Networking, 2012, 34(3): 206-219
- [6] Yu Min-lan, Yi Yung, Jennifer R, et al. Rethinking virtual network embedding: substrate support for path splitting and migration[J]. ACM SIGCOMM Computer communication review, 2008, 38(2): 17-29
- [7] Eppstein D. Finding the k shortest paths[J]. SIAM Journal on Computing, 1998, 28(2): 652-763
- [8] Li Xiao-ling, Wang Huai-min, Guo Chang-guo, et al. Topology awareness algorithm for virtual network mapping[J]. Journal of Zhejiang University-Science C(Computers & Electronic), 2012, 13(3): 178-186
- [9] Zegura E, Calvert K, Bhattacharjee S. How to model an internet-work[C]// Proceeding of the 15th Annual Joint Conference of the IEEE Computer and Communications Society. San Francisco CA, USA, 1996: 594-602
- [10] Hallberbach A, Bauer T, Reichert M. Guaranteeing Soundness of Configurable Process Variants in Provop[C]// CEC. IEEE, 2009: 98-105
- [11] Hallberbach A, Bauer T, Reichert M. Capturing Variability in Business Process Models: The Provop Approach[J]. Journal of Software Maintenance and Evolution: Research and Practice, 2010, 22(6/7): 519-546
- [12] Haugen K E, Husa R K, Stolen R K. STAIRS towards formal design with sequence diagrams[J]. Software and Systems Modeling, 2005, 4(4): 355-367
- [13] Rosa M L, Dumas M, Hofstede A T, et al. Configurable Multi-Perspective Business Process Models[J]. Information Systems, 2011, 36(2): 313-340
- [14] Li C, Reichert M, Wombacher A. Discovering Reference Models by Mining Process Variants Using a Heuristic Approach[C]// BPM, 2009: 344-362
- [15] Li C, Reichert M, Wombacher A. The Minadapt Clustering Approach for Discovering Reference Process Models Out of Process Variants[J]. International Journal of Cooperative Information Systems, 2010, 19(3): 159-203
- [16] Schroeter J, Mucha P, Muth M, et al. Dynamic Configuration Management of Cloud-based Applications[C]// SPLC. 2012: 171-178
- [17] Mietzner R, et al. Variability modeling to support customization and deployment of multi-tenant-aware software as a service applications[C]// PESOS. 2009: 18-25
- [18] Reinhartz-Berger I, et al. Organizational Reference Models: Supporting an Adequate Design of Local Business Processes[J]. IJBPM, 2009, 4(2): 134-149
- [19] Reinhartz-Berger I, et al. Extending the Adaptability of Reference Models[J]. IEEE Transactions on Systems, Man and Cybernetics (Part A), 2011, 40(5): 1045-1056
- [20] Rosemann M, Wil. M. P. van der Aalst. A Configurable Reference Modeling Language[J]. Information Systems, 2007, 32(1): 1-23
- [21] Ruehl S T, Andelfinger U. Applying software product lines to create customizable software-as-a-service applications [C] // SPLC Workshops. 2011: 1-16
- [22] Jiang Jian-min, Zhang Shi, Gong Ping, et al. Modeling and analyzing mixed communications in service-oriented trustworthy software[J]. Science China Information Science, 2012, 55(12): 2738-2756
- [23] Jiang Jian-min, Zhang Shi, Gong Ping, et al. Message dependency-based adaptation of services[C]// APSCC. 2011: 442-449
- [24] Jiang Jian-min, Zhang Shi, Gong Ping, et al. Configuring Business Process Models [J]. ACM SEN, Software Engineering Notes, 2013(4)

(上接第 133 页)