

Overlay 网络上的服务部署问题的研究（二）¹

余镇危, 冯烟利, 潘耘
中国矿业大学(北京), (1000083)
zwyu@cumtb.edu.cn

摘要: 本文提出了 Overlay 网络上的服务部署问题, 给出形式化描述, 并给出了一对多单因子服务部署问题的数学模型和求解算法, 通过数值模拟表明了算法的有效性。

关键字: Overlay 网络, 服务部署, 单因子服务

1 引言

把 Overlay 网络上的能提供的服务分成两类: 一对一的服务和一对多的服务。为了支持 Overlay 网络能提供多种服务, 并能组合出多种服务的能力, 引入了服务因子的概念。文章《Overlay 网络上的服务部署问题的研究(一)》中研究了一对一单因子的服务部署问题部署, 本文继续从一对多的单因子服务部署问题的角度进行研究。

2 一对多的单因子服务部署问题

目前几乎所有的一对多的服务都是建立在组播服务的基础之上的, 所以我们在研究一对多类型的服务部署问题时, 重点研究组播服务节点的部署问题。在 Overlay 网络上建立组播路由, 需要部署组播节点到基础网络之上, 这是进行组播路由的基础。我们把组播看作一种服务, 选择合适的主动节点让其承担组播服务, 即不但存储转发, 而且要复制数据包, 如图 9 所示。我们从如何部署一组组播会话问题中的组播服务节点入手, 研究了一对多类型的服务部署问题。

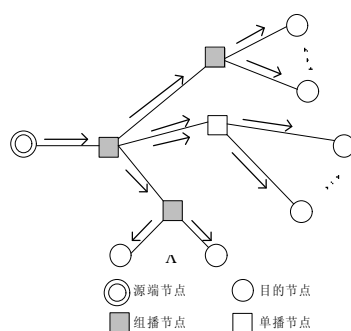


图 9 组播服务的部署问题

Fig. 9 The placement problem in multicast service

2.1 对问题的形式化

首先把问题抽象, 网络模型可表示为带权无向图 $G=(V, E)$, V 是 Overlay 网络节点的集合, E 是边(表示节点之间的连接)的集合, 边权 (edge weight) 表示节点之间的权值, 可

¹基金项目: 高等学校博士学科点专项科研基金资助课题 (20030290003)

解释为延迟、代价等等，定义为 $e:V \times V \rightarrow (0, \infty), e(i, j) = e(j, i)$ ；组播会话的集合 S 由多元组 $s_i = (v_{i_1}, v_{i_2}, \Lambda) \quad v_{i_j} \in V$ 组成，每个多元组 s_i 代表会话 s_i 中端节点 v_{i_j} 的集合。网络 G 为会话集合 S 最多提供 K 个具有组播功能的节点，我们的目的是：如何定位这 K 个组播节点，使会话 S 消耗的总费用达到最小。为了方便形式化描述，给出了如下定义：

定义 5 组播节点的标志函数

$$x(i) = \begin{cases} 0 & \text{节点 } i \text{ 为普通节点} \\ 1 & \text{节点 } i \text{ 为组播节点} \end{cases} \quad (5)$$

假定源点为组播节点即 $x(s)=1$ 。

定义 6 上行组播节点：在给定组播树 T 上与节点 i 距离最近的具有组播功能的祖先节点。用 $p(i)=k$ 来表示，其中 $x(k)=1$ 。

定义 7 定义节点 j 的上行组播节点 $p(j)$ 到 j 的路径上的链路

$$y(j, u, v) = \begin{cases} 1 & \text{从 } p(j) \text{ 到 } j \text{ 的路径上有边 } (u, v) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

利用以上的定义，我们可以计算每个节点 i 到其上行组播节点的代价

$$c(i) = \sum_{u \in V} \sum_{v \in V} e(u, v) \cdot y(i, u, v) \quad (7)$$

定义给会话 i 分配的组播节点

$$z(i, j) = \begin{cases} 1 & \text{组播节点 } j \text{ 分配给了会话 } i \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

多组会话中组播节点部署问题可抽象成：

$$\left\{ \begin{array}{l} \min_{x \in \{0,1\}^n} C_M = \sum_{s_i \in S} \left(\sum_{j \in V} c(j) \cdot z(i, j) + \sum_{j \in s_i} c(j) \right) \quad (a) \\ s.t. \quad \forall s_i \in S, \sum_{j \in V} z(i, j) \leq k \quad (b) \\ \sum_{j \in V} x(j) \leq k \quad (c) \\ \forall s_i \in S, \forall j \in V, z(i, j) \leq x(j) \quad (d) \end{array} \right. \quad (9)$$

其中：约束(b)表示每组会话能获得的具有组播功能的节点，不超过 k 个；这里我们假定由于某种现实条件的影响最多只能部署 k 个组播服务；约束(c)表示网络中能提供组播功能的节点为 k 个；约束(d)保证无论组播节点 j 给一个会话 i 定位在那里，总有一个是能提供组播服务的，我们称该模型为：**MPP(MultiService Proxy Placement problem)**。MPP 问题是一个 0-1 整数规划问题，是 NP-hard 的，当 k 确定后，通常可以用松弛变量法来解决该问题。尽管 k 值的确定仍然可以用聚类分析的层次法，象 UPP 模型中的那样通过反复计算求得，但计算代价较高。鉴于遗传算法在求解许多 NP-hard 的优化问题方面的出色表现，这里我们采用遗传算法和启发式算法相结合的方法，来同时对 k 值和服务部署策略进行寻优。先给介绍一下

遗传算法的基本原理，再给出求解 MPP 问题的具体算法。

2.2 MPP 问题的遗传算法求解方案

该模型的解空间为 $x \in \{0,1\}^n$ ， $x(j) = 0$ 表示节点 j 为普通节点；否则，为 1 表示组播节点。先确定组播树（使用启发式算法），再部署组播服务节点：第一步，基于原网络拓扑结构，用最短路径树算法（SPT）针对每个组播会话 S_i ，求得一棵组播树 t_i ；第二步，随机给定一个部署解 x ；第三步，根据 t_i ，就可以确定 $m(j)$ ，进而确定 $y(j,u,v)$ ，进而确定 $c(j)$ ；第四步，根据 t_i 是否包含节点 j 以及 $x(j)$ 的值，确定 $z(1,j)$ ；第五步， $C_1 = \sum_{j \in V} c(j) \cdot z(1,j) + \sum_{j \in S_i} c(j)$ ；第六步，类似第三步到第五步，计算 C_2, C_3, \dots 。最后把它们加起来求得 C_M ；第七步，判断约束，计算惩罚量 $p(x)$ ，并根据 C_M 确定 x 的适应值 $f(x) = C_M + p(x)$ ；第八步，由第二步到第七步构成了完整的个体评价环境 $x \leftrightarrow f(x)$ ，根据此评价环境编写遗传算法，选择最优得 x 值。

2.3 算法实现

以下给出这个算法的类语言描述，先确定组播树（使用启发式算法），再部署组播服务节点（使用遗传算法）：

Algorithm 2 *MtP* (G, S) //Multicast tree-then-Placement

Input: G is the active overlay network description; S is the set of multicast sessions

Output: opt_x is optimum placemat; opt_f is optimum cost

Begin

$T_i = SPT(G, S_i)$; // Use short path tree algorithm to build multicast tree T_i for each of session S_i on the graph G

//employ genetic algorithm to place multicast service nodes//

set population size pop_x ; set maximum generation $maxG$

Initial population x ; // chromosome x is a 0-1 n dimension random vector//

While (not termination condition) do

Begin

$f(x) = fitnessI(G, S, x)$ //compute the fitness value of every individual x //

get optimum individual opt_x ; get optimum solution opt_f ;

Select //adopt roulette wheel selection operation

Crossover, Mutate

Generate the offspring

End

Return opt_x and opt_f

End

Algorithm3 $fitnessI$ (G, S, x) //computer placement policy x 's fitness value

Input: G is the active overlay network description; S is the set of multicast sessions, x is a placement policy

Output: x 's fitness value

```

Begin
   $C_M := 0;$ 

   $p_1 = \text{penalty}\left(k - \sum_{j \in V} x(j)\right), p_2 = p_3 = 0;$ 

  For each of  $T_i$  do
    Begin
      computer  $m(j)$  and  $y(j, u, v)$ , ( $j = 1, 2, \dots, n$ );
      computer  $c(j)$  and  $z(i, j)$ , ( $j = 1, 2, \dots, n$ );

       $C = \sum_{j \in V} c(j) \cdot z(i, j) + \sum_{j \in s_i} c(j);$ 

       $p_2 = p_2 + \text{penalty}\left(k - \sum_{j \in V} z(i, j)\right);$ 

       $p_3 = p_3 + \sum_{j \in V} \text{penalty}(x(j) - z(i, j));$ 

       $C_M = C_M + C + M_1 p_1 + M_2 p_2 + M_3 p_3;$ 
    End

  return  $x$ 's fitness value  $f = \frac{1}{1 + C_M}$ .

```

End

其中： M_1 ， M_2 和 M_3 是三个较大的常数因子，而惩罚函数 $\text{penalty}(x)$ 定义为：

$$\text{penalty}(x) = \begin{cases} (1+a)^x, & x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

2.4 算法复杂性分析：

设 C_{select} , C_{cross} , C_{mutate} , C_{shift} , $C_{inverse}$ 分别表示上述各个遗传算法中染色体适应值评价复杂度、赌盘选择算子复杂度、交叉算子复杂度、变异算子复杂度、基因片断迁移算子复杂度、基因片断翻转算子复杂度；算法的复杂度为 C_I ：

$$C_1 = \sum_{s_i \in S} C_6(|G|, |S_i|) + \max G \cdot (\text{pop}_x \cdot (C_2 + \frac{1}{2} C_{cross} + C_{mutate} + C_{shift} + C_{inverse}) + C_{select})$$

$$C_2 = O(|S| \cdot |V|)$$

$$C_3 \leq \max G \cdot (\text{pop}_x \cdot (C_4 + \frac{1}{2} C_{cross} + C_{mutate} + C_{shift} + C_{inverse}) + C_{select}) + \sum_{s_i \in S} C_5(|G|, k, |S_i|)$$

$$C_4 \leq \sum_{s_i \in S} C_5(|G|, k, |S_i|) + C_2 \quad C_5(n, k, m) \leq C_7(n, k) + C_8(nk, k, m) + C_6(nk, km) + C_9(nk, k, n)$$

$$C_6(n, m) = O(n^2); \quad C_7(n, k) = O(nk); \quad C_8(nk, k, m) = O(km); \quad C_9(nk, k, n) = O(nk)$$

在这里，该遗传算法均采用改进的赌盘选择算子、一致交叉算子、普通的基因位变异算子和基因片迁移及翻转算子，故： C_{cross} , C_{mutate} , C_{shift} 和 $C_{inverse}$ 的复杂度均以 $O(L)$ 为界，并且 $C_{select} = O(P)$ ，这里 L 表示染色体的长度，而 P 表示种群规模。

2.5 仿真模拟

在部署一对多类型的服务节点时,可采用最简单的随机部署策略,随机选取若干个节点来承担组播服务。这里采用随机部署策略进行对比实验分析,我们给出了节点数为 8 的网络,假定只能部署少于 5 个的组播节点, MPP 算法获得的代价为 25, 而用随机部署策略部署的算法平均值为 37.2, 远高于 MPP 算法得到的值, 如图 10。

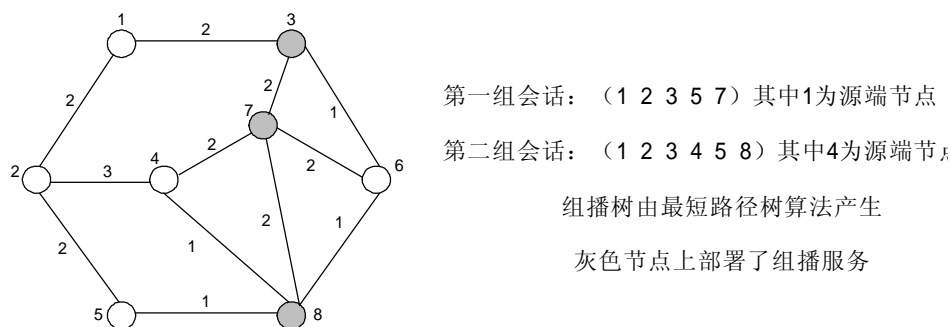


图 10 节点数为 8 的网络

Fig. 10 a simulation network with 8 nodes

我们仍采用 **BRITE** 工具^[1]来生成网络的拓扑图, 拓扑生成基于 **Waxman** 模型^[2]的拓扑生成算法。在我们的实验中, 网络有如下属性: a. 边的费用在 [2,10]上均匀分布。b. 平均每个节点连接边数 $m=2$, $\alpha=0.15$, $\beta=0.2$ 。

在求解问题时, 我们取最大进化代数为 100, 种群规模为 30。早熟控制对于提供遗传算法的全局寻优能力有重要作用, 而交叉概率和变异概率的设定也是每个遗传算法实现时必须考虑的问题, 为此, 我们设计实现了自适应的早熟控制算法。自适应的交叉概率 $p_c=1-md$ 和变异概率 $p_m=md/20$, 基因片翻转概率 p_r 和基因片迁移概率 p_s 均设为 0.005。

图 11 给出了算法的收敛性, 是运行 10 次的平均值, 可以看出收敛性较好。图 12 给出了在网络节点变化时, 与随机部署算法的运行结果的对比, 可以看出随着节点的增多, **UPP** 算法得出了代价值的变化小于随机部署算法, 可部署服务节点的个数不大于总节点的 50%。图 13 给出了在网络节点为 20 时, 组播会话的个数与代价之间的关系, 实验表明会话数的增加会引起代价的增加。图 14 给出了网络节点为 20 时, 组播会话的增加对服务节点的需求。

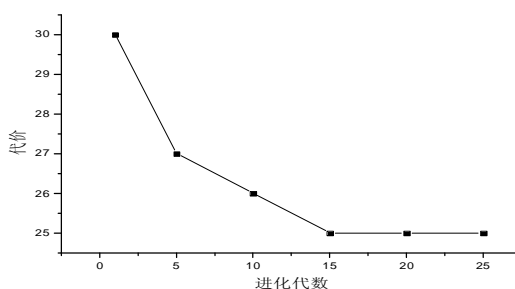


图 11 MPP 算法针对图 11 的收敛性

Fig. 11 The convergence of algorithm MPP on Fig. 11

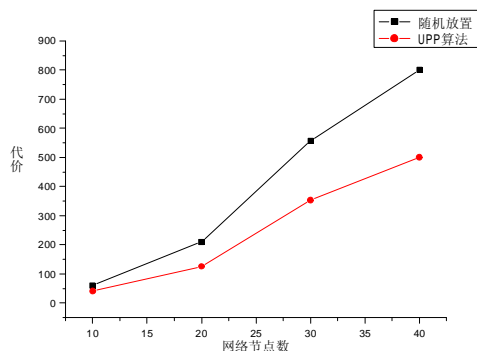


图 12 UPP 算法与随机部署算法的对比

Fig. 12 The comparison of UPP algorithm with random algorithm

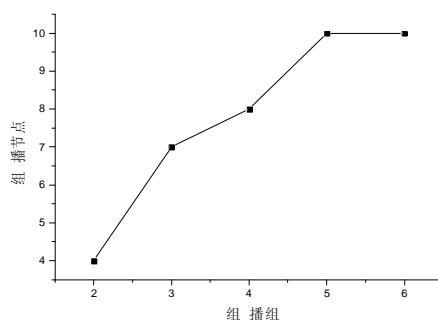


图 13 组播组数与服务节点之间的关系

Fig. 13 The relation of the number of multicast group with the number of service node

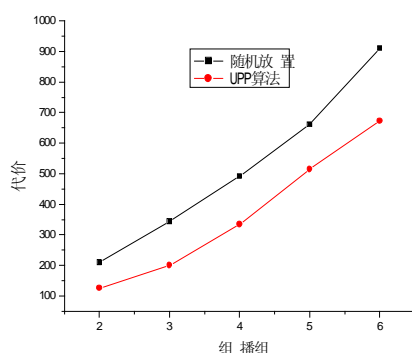


图 14 组播组数与代价之间的关系

Fig. 14 The relation of the number of multicast group with the multicast cost

3 结论

本文研究了Overlay网络上的一对多单因子服务部署问题，从端系统的角度出发研究了这一问题。文献[3]研究了该问题，但是他们的工作是在源端到目的端之间最大延迟的限制下，最小化覆盖整个网络的代理的数目，没有考虑组播的特殊性：组播树包括所有的组播目的节点，也就是说没有考虑优化带宽。文献[4]针对带宽、延迟以及带宽和延迟的混合因素考虑了服务部署的问题，但只考虑了一个组播会话。我们认为应考虑多对组播会话问题中的组播服务节点的部署问题，这有助于提高这个系统的性能。本文中我们研究面对多对组播会话，如何优化组播服务节点的部署位置。以后还需要进一步研究多因子的服务部署问题。

参考文献

- [1] Alberto Medina, Anukool Lakhina, Ibrahim Matta, John Byers. BRITE: An Approach to Universal Topology Generation. In Proceedings of MASCOTS '01, August 2001.
- [2] B.Waxman. Routing of Multipoint Connections. IEEE J.on Selected Areas in Comm., 6:1617-1622, December 1988.
- [3] Placing Servers in Overlay Networks with Jonathan S. Turner, Proc. of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPETS), San Diego, July 2002.
- [4] Min-You Wu, Yan Zhu, and Wei Shu, Proxy Placement for Server-based Multicast Overlays.
www.eece.unm.edu/~wu/papers/mj.pdf.

Service Placement Problem in Overlay Networks (2)

Yu Zhenwei, Feng Yanli , Pan Yun

China University of Mining and Technology-Beijing, Beijing, 100083

zwyu@cumtb.edu.cn

Abstract

The service placement problem in Overlay networks and its formal description are proposed in this paper. The model and the algorithms for one-to-many single factor service placement problem have been given. Numerical experimental simulations have supported the feasibility of the model and the algorithms.

Key words: *Overlay networks, service placement problem, single factor service*