

生成 Internet As 拓扑图 CWT 层次算法研究

陶翼, 王东

(湖南大学计算机与通信学院, 长沙 410082)

摘要: Internet 拓扑图为大范围开发、利用 Internet 提供了一个有力的工具。Internet As 级拓扑图在自治系统层次上刻画 Internet 特征, 它在当前很多领域有着广泛的应用。该文提出了一种生成 Internet As 级拓扑图的 Complete-Waxman-Tree 算法, 它生成具有三层结构的 As 级拓扑图, 生成图在结点度分布、树大小分布以及树深度分布方面与 Internet 比较吻合。

关键词: As 级; As 级拓扑图; Complete-Waxman-Tree(CWT)算法

Research on Complete-waxman-tree Hierarchy Algorithm of Generating Internet As-level Topology Graph

TAO Yi, WANG Dong

(College of Computer and Communications, Hunan University, Changsha 410082)

【Abstract】 Internet topology graph is a powerful tool to develop and utilize Internet on a large scale. The Internet As level topology graph portrays Internet characteristic on the autonomous system level and has the widespread application on many domains at present. This paper proposes a Complete-Waxman-Tree(CWT) hierarchy algorithm that can generate Internet As-level topology graph with three levels, which are consistent with the Internet very well in the distribution of the node's degree, the tree's size and the tree's depth.

【Key words】 As-level(As-L); As-level topology graph; Complete-Waxman-Tree (CWT) algorithm

Internet 拓扑建模是在高层次上开发、利用 Internet 的基础, 对 Internet 拓扑的研究经历了从随机型到层次型再到无尺度网络的过程。由于 Internet 在不断高速增长, 规模已经变得非常庞大, 其结构与原型 ARPANET 已大相径庭^[1], 而且 Internet 不像组成它自身的网络那样有专门的机构来管理和制定其拓扑结构和扩展方式^[3], 这给网络工作者从整体上认识和研究 Internet 带来巨大的挑战; 这也是为什么直到现在还没有人可以给出一幅完整而又详尽的 Internet 图的原因, 而这样的图是非常重要的, 因为很多新协议的开发(如多播协议)和网络实验(如病毒在大规模网络上的传播模拟)是不可能或者不方便直接在 Internet 上进行的^[4]。所以, 如何构造具有 Internet 某些特征的拓扑图就成为了当前倍受关注的问题。

目前对 Internet 拓扑结构的研究主要集中在两个层次: 路由器级 (Router-Level, RL) 和自治系统级 (Autonomous System-Level, As-L)。RL 拓扑研究主要是针对第 3 层 (即网络层) 设备, 一般而言, 第 3 层的设备不是主机就是路由器, 但是 RL 拓扑图只关注路由器而不关心其它网络设备。这一类拓扑图主要应用于网络层协议的开发, Pansiot 是最早研究 RL 拓扑的研究者之一^[5]。由于 Internet 中路由器数量太大以及某些商业原因, 获得准确的大规模 RL 拓扑图几乎是不可能的^[2,5]。因此大范围内考察 Internet 拓扑结构一般更多考虑 As-L 拓扑图。As 数量比较少 (不会超过 65 535 个^[4]), 因此可以在相对较小的 As-L 拓扑图里研究大范围网络^[6,7]。

1 As-L 拓扑概述

Internet 拓扑通常把 Internet 表示为无向图 G , $G = (V, E)$, 其中 $V = \{v | v \text{ 是 } G \text{ 中的一个结点}\}$; $E = \{(u, v) \in V \text{ 且 } u, v \text{ 邻接}\}$; d_v 表示结点 v 的出度, $d_v = \{|u | (u, v) \in E\}$; f_d 表示节点出度频率,

$$f_d = \{|v | v \in V \text{ 且 } d_v = d\}.$$

最早的 Internet 拓扑建模方法是 Waxman 提出的 Waxman 模型^[8], 最早从 As 角度对网络拓扑结构进行研究的是 Govindan 等人, 现有 As-L 拓扑建模方法一般有几类:

(1) 随机型。这一类模型认为 Internet 拓扑图处于一个完全无序状态, 在大尺度上是均一的。最典型、用得最多的随机模型是 Waxman 于 1988 年提出的 Waxman 模型^[8]。随机模型最大的优点就是算法简单、容易实现, 但是它把整个 Internet 看作是一种松散的、无序的状态而没有刻画出 Internet 应具有的特征。

(2) 层次型。这一类模型出于对 Internet 所具有的层次特征的认识, 同一层次结点度接近而不同层次间结点度差别很大, 对同一层次结点一般用 Waxman 模型处理。层次模型的优点是刻画了 Internet 所具有的层次特性, 它比随机模型更符合实际 Internet 特征, 但是当前的层次模型主要是以结点度作为划分层次的依据, 在实际操作中不好把握这一标准。

(3) 幂律型。1999 年, Faloutsos F 等人对 NLANR (National Lab for Applied Network Research) 的 3 份原始的 BGP 数据以及一份 traceroute 测量数据进行分析, 发现 As 拓扑存在幂律 (Power-Law) 关系, 他们总结出 4 条幂律^[9]。幂律型只是从现有的路由表数据 (如 NLANR 提供的 BGP 路由数据) 来分析当前的 Internet 拓扑结构, 并没有真正的产生拓扑图, 而且由于 BGP 路由表数据太大而且在不断地增长、变化, 因此直接

作者简介: 陶翼 (1979 -), 男, 硕士生, 主研方向: 网络拓扑结构与拓扑发现; 王东, 博士、副教授

收稿日期: 2005-12-19 **E-mail:** jt_taycom13@163.com

从BGP路由表数据分析Internet拓扑图是一件工作量巨大而繁琐的工作。

(4)其它模型。在随机型、层次型以及幂律型中,注重的是生成拓扑图的方法而没有考虑实际Internet中的关系。Lixin Gao从商业以及ISP关系的角度把Internet中As的关系划分为3类:提供者和客户(Provider-Customer)关系、端到端(Peer-Peer)关系以及兄弟(Sibling-Sibling)关系并提出了由BGP路由表数据构造这种商业关系As级拓扑图的算法^[10]。为了获得更完整的As-L拓扑图,Chang等人利用了更多的数据源得到了更完整的As-L拓扑图,他们的拓扑图中的链路比单一GBP路由表的多了40%^[11]。

Internet是一些松散网络的集合,它们被分为多个层次又相互连接,具有多层次、自治系统、平均节点度低以及可控制的(每跳)直径等基本性质。在当前比较成熟的As拓扑建模方法中,随机模型不能刻画出Internet的特征;传统层次模型依据节点度来划分层次;幂律型以及其它的方法基本上都是利用BGP路由表数据来分析Internet拓扑结构工作量太大。针对现有模型的不足,本文提出了一种直接生成Internet As级拓扑图的Core-Waxman-Tree(CWT)算法,它可以生成具有三层结构的As级拓扑图,克服了随机模型的松散结构缺陷,具有Internet多层次的特征,同时它不依据度来分层,可以以较小的代价获得Internet As-L拓扑图。

2 CWT 算法

2.1 算法基本思想

CWT算法从自治系统(As)的观点把Internet分为3个层次:第1层为核心网络,它是Internet的核心,由核心网络结点连接组成;第2层为随机网络,由一些松散的相互独立的随机网络子图组成;第3层是拓扑树,由相互独立的树组成,其根结点是高层网络结点,上层网络和拓扑树通过根结点连接构成Internet As级拓扑图。在算法中,首先通过完全图构造核心网络,然后通过现有模型把核心网络结点扩展为一个第2层随机网络子图;最后在高层网络中选择根结点构造拓扑树,这样就得到了Internet As拓扑生成图。

2.2 算法参数

下面给出算法中一些参数的意义:

Num_{Top} :生成图最高层网络总结点数;

$Num_{Top-Core}$:生成图最高层网络骨干结点数;

Num_{Top-K} :生成图最高层网络以K连接加入的结点数;

$Num_{Top-\bar{K}}$:生成图最高层网络介于骨干结点与K连接结点的结点数,满足:

$$Num_{Top-\bar{K}} = Num_{Top} - (Num_{Top-Core} + Num_{Top-K});$$

$Num_{Sec-Ave}$:生成图期望的第二层网络子图平均结点数;

Num_{Tree} :生成图中树的非根结点总数;

$Num_{Tree-Ave}$:期望树平均大小(用结点树衡量,不含根结点)。

2.3 算法描述

Phase 1://生成最高层网络

第1步 在作图平面内随机放置 $Num_{Top-Core}$ 个结点,把它连接成完全图,构成骨干节点;

第2步 在作图平面内随机放置 $Num_{Top-\bar{K}}$ 个结点,每加入一个结点都与第1步图中离自己最近的K个结点连接,与其余的结点以0.2的概率连接;

第3步 在作图平面内随机放置 Num_{Top-K} 个结点,每加入一个结点都与第2步图中离自己最近的K个结点连接而与第K+1个结点以0.2的概率连接;

Phase 2://生成第2层网络

第4步 把作图平面分为 Num_{Core} 个区域,在每个区域内,按照下面的方法作图:在区间 $[1, 2Num_{Sec-Ave} - 1]$ 随机取一个整数 Num_{Waxman} ,用Waxman模型作一幅 Num_{Waxman} 个结点的随机图;

第5步 调整第4步的图,对图中每个度为1的结点,以概率0.2连接到其它结点;

第6步 用第5步的图替代第3步中图里面每一个结点;

Phase 3://生成拓扑树

第7步 $Num_{Tree} = \left\lceil \frac{Num_{Tree}}{Num_{Tree-Ave}} \right\rceil$,在第6步的图中选择结点度最大的 Num_{Tree} 个结点为根结点;

第8步 在作图平面内随机放置 Num_{Tree} 个树结点,用下面的方法把所有树结点加入图中:首先,对每一个根结点把它连接到离它最近的树结点,然后对每一个尚未加入到树的树结点,把它加入到树(已经加入树的结点和根结点)。

图1是用CWT算法生成的一幅200个结点的As-L拓扑图。图中实心结点与黑粗线边构成最高层网络;空心结点与细灰直线构成第2层网络;方形结点与细虚线边组成拓扑树。

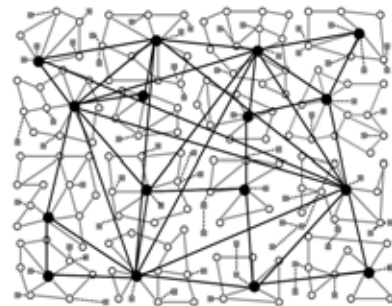


图1 例图

3 算法分析及实验结果

3.1 算法时间复杂度分析

算法分为3个部分:生成最高层核心网络,第2层随机网络和第3层拓扑树。其中生成最高层网络的代价主要是在完全图的生成,生成一幅完全图的代价为 $O(n^2)$,所以生成最高层网络的时间复杂度为 $O(n^2)$;生成第2层随机网络的时间开销主要用在Waxman生成随机图的地方,Waxman生成随机图的复杂度为 $O(n^2)$,共生成 n 幅随机图,因而时间开销为 $O(n^3)$;生成第3层拓扑树的时间开销与树结点相关,为 $O(n^3)$,所以该算法的时间复杂度为 $O(n^3)$ 。

3.2 生成图性质分析

3.2.1 生成图连通性分析

结论 CWT算法生成的As-L拓扑图是连通的。

证明

(1)最高层网络是连通的。最高层网络是由完全图扩展而来的,完全图是连通图,而在添加其他节点的时候也保证了连通性,所以最高层网络是连通的。

(2)第2层网络子图以及最高层网络和第2层网络之间是连通的。该算法用Waxman模型生成第2层网络子图,保证了子图连通性,经过调整以后的子图具有更好的连通性,所以生成的第2层网络子图是连通的;用第2层网络子图最大

的节点取代最高层网络的节点，保证了高层网络的连通性。

(3)拓扑树与高层网络是连通的。树是连通的而同时树的根节点是高层网络节点，所以拓扑树与高层网络是连通的。

综上所述，CWT 算法生成的 As-L 拓扑图是连通的。

3.2.2 生成图健壮性分析

利用 CWT 算法生成的拓扑图高层网络具有很稳定的性能，当某个高层网络节点发生故障时并不会影响整个高层网络的连通性，这一特征是与实际 Internet 十分相符的。如果是树根节点发生故障，那么对那棵树的节点来说是灾难性的，所有树中的节点都会与高层网络断开，但是由于在生成图中树的规模一般都比较小，因此并不会影响对大范围网络的研究；如果仅仅是根节点某条边发生故障就不会存在这样的问题，因为根节点是拓扑图中结点度最大的高层节点，当它的某一条边发生故障时并不会对其连通性造成很大影响。

3.2.3 实验结果分析

我们在CPU为赛扬 1.7GHz，256MB内存和VC6.0的环境下进行实验。在实验中，分别做了 400、600、1 000 个结点的拓扑图，对Waxman参数的取值为： $\alpha = 0.2$ ， $\beta = 0.3$ ，下面是实验结果和分析结果^[4]的比较。

(1)结点度分布

从图 2 可以看出，采用 CWT 算法生成的 As-L 拓扑图结点度的分布与直接从 BGP 路由表数据分析的结果是很相近的，本文的拓扑图里面存在一部分结点度比较大的结点，这与本文算法有关。

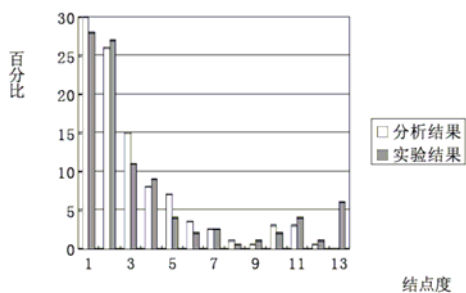


图 2 结点度分布

(2)生成图树特征

1)树大小分布

从图 3 可以看出，CWT 算法的生成图树大小的分布与分析结果也是基本吻合的，由于实验生成图的规模比较小而且对树平均大小参数设置得较小，因此没太大的树生成，结点少于 10 的树分布与分析结果是非常相似的。

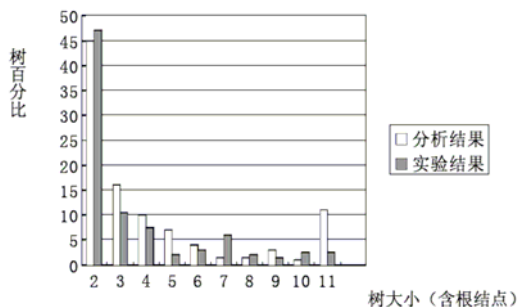


图 3 树大小分布

2)树深度分布

从图 4 可以看出，CWT 算法生成的图在树深度方面与分析结果是基本一致的，但在该算法生成的图中有一小部分深度为 3 的树。

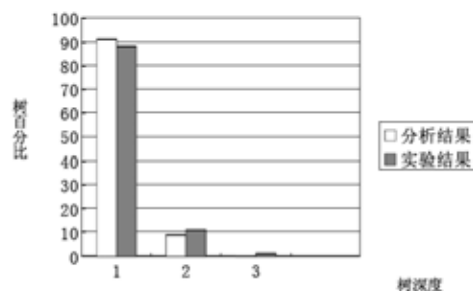


图 4 树深度分布图

4 结论

本文提出了一种直接生成 Internet As-L 拓扑图的 CWT 算法，它克服了随机模型的松散性、当前层次模型使用度划分层次的模糊性以及幂律型和其它模型工作量大的缺点。使用 CWT 算法可以在 $o(n^2)$ 的时间代价内生成具有三层结构的 As-L 拓扑图，它保证了连通性而且生成的拓扑图在结点度分布、树大小分布以及树深度分布等特征与从实际 BGP 路由表分析的结果比较吻合。说明 CWT 算法生成的 As-L 拓扑图是比较符合实际的 Internet 特征的。下一步工作要解决算法生成的拓扑图的局部稳定性问题。

参考文献

- 张宇, 张宏莉, 方滨兴. Internet 拓扑建模综述[J]. 软件学报, 2004, 15(8).
- 姜誉, 方滨兴, 胡铭曾等. 大型 ISP 网络拓扑多点测量及其特性分析实例[J]. 软件学报, 2005, 16(5).
- Magoni D, Hoerd M. Internet Core Topology Mapping and Analysis[J]. Computer Communication, 2005, 28(5): 494-506.
- Magoni D, Punosiot J J. Analysis of the Autonomous System Network Topology[J]. ACM SIGCOMM Computer Communication Review, 2001, 31(3): 26-37.
- Pansiot J J, Grad D. On Routers and Multicast Trees in the Internet[J]. ACM Computer Communication Review, 1998, 28(1): 41-50.
- Zhang Beichuan, Liu Raymond, Massey D, et al. Collecting the Internet AS-level Topology[J]. ACM SIGCOMM Computer Communication Review, 2005, 35(1).
- Dimitropoulos X A, Krioukov D V, Riley G F. Revisiting Internet AS-level Topology Discovery[C]. Proc. of the Passive & Active Measurement Workshop, 2005.
- Waxman B M. Routing of Multipoint Connections[J]. IEEE Journal on Selected Areas in Communications, 1988, 6(9): 1617-1622.
- Faloutsos M, Faloutsos P, Faloutsos C. On Powerlaw Relationships of the Internet Topology[J]. ACM SIGCOMM Computer Communication Review, 1999, 28(4): 251-262.
- Gao Lixin. On Inferring Autonomous System Relationships in the Internet Topology[J]. IEEE/ACM Trans. on Networking, 2001, 9(6).
- Chang Hyunseok, Govindan R, Jamin S, et al. Towards Capturing Representative AS-level Internet Topologies[R]. Tech. Rep: CSE-TR-454-02, EECS Department, University of Michigan, 2002.