



Internet 拓扑发现的探测点部署方法的研究

北京航空航天大学 计算机学院, 北京 100083

摘 要: 获取骨干网络的拓扑信息对于研究网络内在结构和网络管理等都有着重要的意义。对 Internet 网络进行拓扑发现的主要方法是部署一定的探测点分别进行 traceroute 探测, 最后将各探测点的结果合并。由于客观条件的限制, 在大多数探测系统中, 探测点通常较少且部署代价较大, 使得探测的结果和实际网络有较大的出入。因此, 在探测点数量固定的情况下, 如何更有效的部署探测点, 从而使得拓扑发现结果更丰富, 就成为了一个关键的问题。本文首先研究了探测点的位置与探测结果的关系, 并在此基础上提出了一种部署 traceroute 探测点的方法。最后, 本文通过实验证明了使用此方法部署探测点比随机部署能更有效的发现网络中的节点和链路, 并使用 skitter[2]的原始探测数据对上述方法进行了验证。

关 键 词: 拓扑发现 探测点 traceroute 部署

中图分类号: TP393.07

文献标识码: A

文章编号: AF040802

On the placement of traceroute-like topology discovery instrumentation

School of Computer Science, Beihang University, Beijing 100083, China

Abstract: An accurate map of the Internet is very important for studying the network's internal structure and network management. The main approach to map the Internet is to collect information from a set of sources by using traceroute-like probes. In a typical mapping project, active measurement sources are relatively scarce while traceroute destinations are plentiful, which makes the sampled graphs quite different from the original ones. So, in the case that the number of sources is fixed, it becomes very important to determine how to place these sources such that the sampled graphs can be closer to the original ones. In this paper, we investigate the relationship between the placement of traceroute sources and their sampled result. Based on the relationship, we propose a method on how to place the traceroute sources. We show that the graph sampled from sources selected using our method is more accurate than the ones randomly selected. We also validate our conclusion using the raw trace data of skitter[2] project.

Key words: Topology discovery, Traceroute sources

引言

获取一张精确的 Internet 拓扑图是对 Internet 进行建模、分析和测试的前提, 并且对研究网络的内在结构和网络管理都有重要的意义。当前, 对 Internet 骨干网络进行拓扑发现的主要方法是 traceroute 探测, 即在网络中部署一定数量的探测点, 并且获取一个分布广泛的地址列表, 然后从探测点向地址列表中的所有地址进行 traceroute 探测, 获取 traceroute 路径上经过的接口及接口之间的连接信息, 最后将所有探测点获得的上述信

息进行汇总, 就得到了一张 Internet 接口级拓扑图, 图中节点代表网络中的网络设备的接口, 链路表示接口之间的连接。再通过 whois[1]查询等工具将路由器的接口地址转化为 AS(Autonomous System)域信息, 就可以得到 Internet 的 AS 级拓扑图。目前, traceroute 探测方法被广泛应用于各种拓扑发现系统[2,3,4,5,6]。

由于拓扑发现的探测点需要进行实际的部署, 其代价较大, 因此一个典型的 traceroute 拓扑探测系统的探测点非常有限。例如, 著名的 IPv4

骨干网络拓扑发现工具 skitter[2]只有 25 个探测点,分布在美洲、欧洲、日本和澳洲,而其目的地址为 971000 多个。单个探测点的探测结果可以近似看作以这个探测点为根节点的一颗生成树。因此,traceroute 探测的结果就是将各个探测点出发的生成树进行合并的结果。由于探测点有限,其合并的结果和实际网络必然存在一定的差距。Lakhina 等通过实验[7]指出,traceroute 探测倾向于发现距离探测点较近的节点和链路,而距离探测点越远,节点和链路的发现的概率越低。DIMES 项目[8]实现了 Donnet 等人[9]中提出的 traceroute@home 算法,并且提供源程序供志愿者下载。截至 2006 年 2 月, DIMES 已经有 12769 个探测代理。Shavitt 和 Shir[10]对 DIMES 的运行结果进行了分析,发现通过大量增加探测点,可以发现许多小的 ISP 之间的链路,而这些链路是少量的探测点难以发现的。

由上文可以看出,一方面,由于 traceroute 探测自身的限制,必须适当增加探测点才能得到更准确的 Internet 拓扑图;而另一方面,大量的探测点不仅部署困难,而且由于 traceroute 探测本身需要向网络中发送大量的探测报文,必然对正常网络的使用造成很大的影响。如果由于客观条件的限制,traceroute 探测系统只能部署固定数量的探测点,则如何部署这些探测点,使得探测结果更丰富,成为一个关键的问题。许多拓扑发现的组织在部署探测点时,只是按照地理位置分布均匀的原则进行部署,并没有固定的策略。本文的主要贡献在于分析了 traceroute 探测的特性,并且针对这些特性,得出了如何部署探测点,以使得拓扑发现的结果更丰富的算法,并利用 skitter[2]拓扑发现的原始数据进行了验证。通过本文的实验可以得出,采用不同的策略部署探测点,即使探测点数目相同,其发现结果的差异仍然较大。

本文第 1 部分介绍了国内外关于探测点部署的相关工作。第 2 部分对 traceroute 探测的特性进行了分析。在此基础上,第 3 部分得出了探测点的位置与探测结果之间的关系,并利用 skitter[2]的原始数据进行了验证。第 4 部分提出了如何部署探测点的算法,在模拟图上验证了算法的有效性。最后,总结全文并展望下一步研究方向。

1 相关工作

Barford 等人[11]对增加探测点对探测结果的

边际效应进行了分析。他们发现,增加探测点对探测结果的影响远不如增加目的站点。当探测点增加到 2 以上时,增加探测点的边际效应急剧下降。Barford 等人的结论是基于 skitter 拓扑发现的原始数据的,而 skitter 在选取探测点时只是考虑了地理位置的因素。同时,他们也指出,增加探测点的边际效应的减少并不能说明发现结果已经很全面。

L Dall'Asta 等人[12]分析了网络中节点或链路的发现概率与探测点和目的地址的数量及图本身的特性(介数)的关系。文章的末尾部分提到,由于介数较小的节点被发现的概率较小,因此把探测点部署在介数较小的节点上,能够发现更多的节点和链路。但文中同时提到,对于一个未知的待发现的网络,其节点的介数是未知的,因此此方法无法应用于实际的拓扑发现过程。而文中把度较小的节点当作介数较小的节点,从而建议将探测点部署在度较小的节点上,也仅仅是猜测,并没有进行实际的实验。

JL Guillaume 和 M Latapy[13]通过模拟实验,对不同网络中 traceroute 探测的有效性进行了分析。通过比较探测结果与原始网络的差别,分析了对探测结果造成影响的因素。

Jamin 等人[14]针对 IDMaps[15]中如何部署探测点的问题进行了研究。他们研究了当探测点数目固定时,如何部署探测点的问题。但由于 IDMaps 的目标是使网络中任意一个节点距离其最近的探测点的距离最小,因此其算法无法应用于 traceroute 探测过程。

2 traceroute 探测的特性

为了研究探测点的位置与探测结果之间的关系,本文首先选取了一副模拟网络,在此网络上对 traceroute 探测进行模拟。在分析 traceroute 探测特性的基础上,将单点探测不能发现的节点和链路划为两种情况。

2.1 实验网络的选取

为了研究探测点的位置对探测结果的影响,我们需要在一副能够反应真实网络状况的拓扑图上进行相应的实验。[16]提出了复杂网络中普遍存在的 power-law 特性。[17]提出了基于 power-law 特性的 EBA 模型,而后提出的许多模型也都是基于 power-law 特性的。本文中选取 EBA 模型作为进行模拟的实验网络模型,并选取

了 EBA 模型的典型参数生成实验网络,此网络的各个参数如表 1 所示。为了验证我们的结果,我们将在后文中对 skitter 发现的实际数据进行类似的分析。

EBA 模型参数	m_0	3
	m	2
	p	0.6
	q	0.1
图属性	节点数	2000
	链路数	10985
	平均度	10.98
	最大度	128

表 1 实验网络的主要属性

为了模拟 traceroute 探测,首先需要解决的问题是路由策略的选取。实际 IPv4 骨干网络中的路由器运行着多种路由协议,如 BGP,OSGP 等。路由协议的基本策略是让网络中的报文经过最短的路径到达目的站点。虽然由于政治、经济等因素的影响,实际网络流量并不完全依照最短路径。然而,本文中为了模拟实验的方便,并不考虑上述因素,而是假设网络中的所有流量经过的都是最短路径。

对 traceroute 探测进行模拟,需要解决的另一个关键问题是最短路径不唯一的情况。当探测点和目的节点之间存在多条权值相同的最短路径时,如何选取最短路径对发现结果的影响很大。这种情况在模拟网络中和实际 Internet 网络中都普遍存在。[12]将最短路径不唯一的处理方式分为三类:USP(唯一最短路径),RSP(随机最短路径)和 ASP(所有最短路径)。对 Internet 网络进行实际探测时,混合包含了上述三种策略。在对 traceroute 探测进行模拟时,我们选取了 USP 策略,这是因为 USP 策略最接近于进行一次实际探测的情况,同时 USP 策略反应了最坏情况下的探测结果。在后文中,我们将看到,最短路径不唯一是导致探测结果的覆盖率不高的重要因素。

2.2 增加探测点的边际效应

Barford 等在[11]中提到,当部署的探测点增加时,探测结果的增加却急剧减少。这被称作边际效应。然而,增加探测点对结果的边际效应的下降并不能说明发现结果的全面性。由于[11]所

分析的是 skitter 拓扑发现的原始探测数据,无法得到实际 Internet 网络的真实拓扑,因此无法分析其发现结果的全面性。本节将在实验网络上模拟 traceroute 探测的过程,并对探测的结果全面性进行分析。

图 1 是当增加探测点数量时,traceroute 探测结果的变化情况。其中,探测点和目的站点都是从实验网络中随机挑选的节点。(a) 图为随机选取 1000 个节点作为目的站点的结果。从图中可以看到,随着探测点的增加,特别是当探测点增加到 3 个后,发现结果的增加就变的非常缓慢。当随机选取 15 个探测点时,traceroute 探测发现的节点和链路个数分别为 1054 和 4136,覆盖率分别为 52.7%和 37.6%。由此可以得出,随着探测点的增加,发现结果的边际效应急剧下降,但发现的覆盖率仍然很低。为了消除目的站点的选取对结果的影响,本文还将图中的所有节点都作为目的站点进行模拟探测。在这种情况下,网络中的所有节点都可以被发现,其链路的发现情况如(b)图所示。可以看出,即使把网络中的所有节点都作为探测点,链路的覆盖率仍然很低。使用 50 个探测点时,链路的覆盖率仍然只有 57.5%。

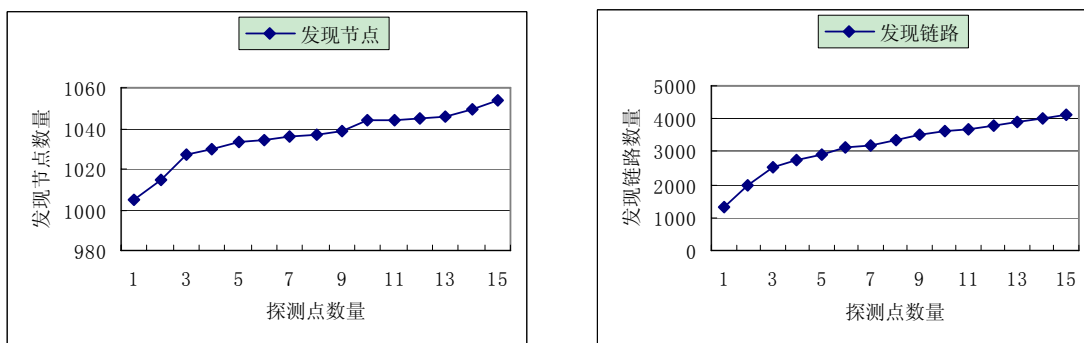
由上面的实验我们得出,(1)如果随机选取探测点,探测结果的边际效应会随着探测点的增加急剧减少,但节点和链路发现的覆盖率很低;(2)即使目的站点分布的非常广泛(极端情况下所有的节点都是目的站点),链路的发现覆盖率仍然很低。下节中将对 traceroute 探测难以发现的节点和链路进行分析。

2.3 单点探测的缺陷

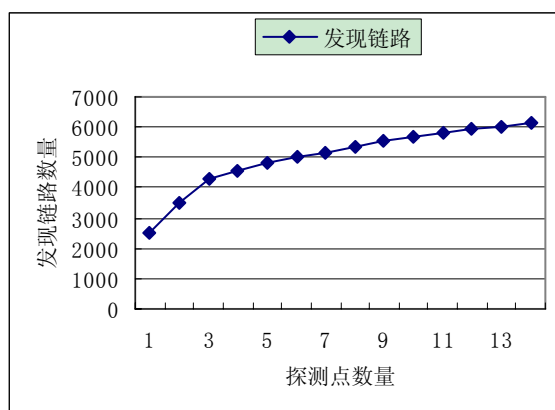
[12]中指出,一个节点或者链路能够被探测到的概率取决于探测点和目的站点的分布密度和这个节点或链路本身的介数。本文侧重于研究探测点的选取对于探测结果的影响。因此,我们首先假设 traceroute 探测的目的节点分布广泛而均匀。从上节的实验可以看到,当探测点个数为 1,所有节点都是目的节点时,探测发现的链路数为 2924 条,仅占原始图的 26.6%,链路发现的覆盖率很低。其实,对于单点探测,其发现不了的节点和链路可以分为两种情况:

2.3.1 cross-link

单点探测的结果本质上是从这个探测点出发到各个目的节点的一颗生成树。单点 traceroute



(a) 1000 个目的站点



(b) 所有节点作为目的站点

图 1 随着探测点数量的增加，traceroute 探测发现结果的增加情况

探测的过程实际上可以看作是从探测点对这颗生成树进行广度优先遍历的过程，遍历过程中的层数对应于 traceroute 探测中的跳数。为了方便描述，在下文中，某个节点的“层数”就表示此节点在从探测点开始进行广度优先遍历过程中的层数，也就是从探测点到此节点进行 traceroute 探测的跳数。cross-link 是在对这颗生成树进行广度优先遍历的过程中发现不了的链路，也就是位于生成树的同一层节点之间的连接。如图 2，monitor 在向 dest1 和 dest2 中进行探测的过程中，并不能发现 A 和 C 以及 D 和 H 之间的节点和链路。这就是 traceroute 的 cross-link 问题。传统解决 cross-link 的方法是分布更多更广的探测点。如在 H 点部署一个探测点，则 H 在探测 dest1 的过程中就可以发现 H 和 D 之间的节点和链路。

2.3.2 最短路径不唯一现象

另一种 traceroute 探测较难发现的情况是前文提到的最短路径不唯一的情况。如图 3 所示，由 monitor 到 dest1 的最短路径共有四条，但一次 traceroute 只可能发现其中一条路径上的节点和

链路，这对应于[12]中提到的 USP 探测。实际的 traceroute 探测过程是由 USP，RSP 和 ASP 混合组成的，而随着探测时间和次数的增加，从探测点到目的站点的路径可能会发生改变，从而使得 traceroute 探测趋向于 ASP 探测。但最短路径不唯一现象在很大程度上影响了短期探测或一次性探测的全面性。即使对于长期探测，由于 Internet 中大量的备份路由的存在，单点探测仍然难以发现这些路径上的节点和链路。解决上述问题的方法是部署更多的探测点。在图 3 中，如果 D 点也是一个探测点，那么 D 在向 dest2 探测的过程中就可以发现更多 monitor 难以发现的链路。

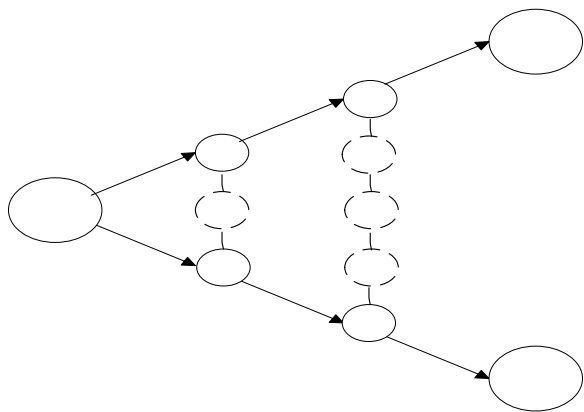


图 2 cross-link 示意图

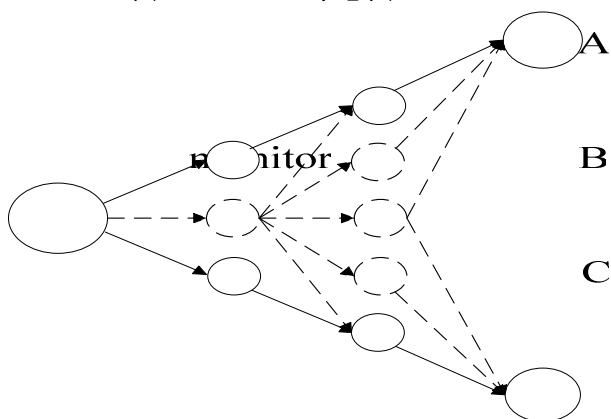


图 3 最短路径不唯一现象示意图 实线的节点和链路为 traceroute 发现的节点和链路。虚线表示由于最短路径不唯一而发现不了的节点和链路

表 2 是对实验网络进行单点探测的统计结果。当把图中所有节点都作为目的节点时，单点探测能够发现的链路仅占全部链路的 22.7%。由表 2 可以看出，由于 traceroute 探测自身的缘故，即使目的节点的分布非常广泛，一个探测点所能发现的链路也只是整个网络的一小部分。针对 cross-link 和最短路径不唯一的情况，部署更多的探测点都可以发现更多的节点及链路。

链路类型	数量	百分比
总链路	10985	
发现链路	2494	22.7%
由于 cross-link 发现不了的链路	5050	45.9%
由于最短路径不唯一而发现不了的链路	3441	31.3%

表 2 单点 traceroute 探测结果统计

3. 探测点的位置与探测结果的关系

3.1 探测点之间结果差异与相对位置的关系

单点探测的结果是以探测点为根的一颗生成树。而将多个探测点的探测结果合并，就得到了最后的拓扑图。也就是说，traceroute 探测的结果是各个探测点发现结果的并集。[7]中提到，对于单点探测，不同探测点之间发现结果的覆盖率差别不大。而当探测点数量固定时，要使得合并后的结果最丰富，本质上就是要使各个探测点之间的交集最小。假设存在探测点 M 和 M'，其单点探测的发现结果分别为 S 和 S'。当 S 和 S' 完全相等时，合并后的探测结果最少，为 S(或 S')；而当 S 和 S' 完全不同时(S 与 S' 的交集为空)，合并后的结果达到最大，为 S 与 S' 结果之和。本节将首先分析如下问题：给定探测点 M 及其探测结果 S，如何选取探测点 M'，使得 M' 的发现结果 S' 与 S 的交集最小。traceroute 的发现结果包含节点和链路，而节点和链路的发现覆盖率往往成正比关系，因此本节中将主要分析链路的发现情况并假设图中所有节点都是目的节点。由上节可知，探测点 M 发现不了的链路可以分为两种情况：第一种为 cross-link 中包含的链路，第二种为最短路径不唯一而发现不了的链路。下文将对这两种情况分别进行分析：

3.1.1 cross-link

由图 2 可以看出，cross-link 是同层节点之间的链路，单点探测完全不能发现 cross-link 所包含的链路。假设将 M' 部署在第 N 层，则 M' 在向第 N 层其它节点进行探测的过程中，就可以发现第 N 层的 cross-link 所包含的链路。而第 N 层的节点越多，则其所含的 cross-link 链路也就越多。所以，将 M' 部署在节点最多的那一层，就可以发现最多的 M 发现不了的 cross-link 链路。

3.1.2 最短路径不唯一

一个网络的链路可以分为两类：一类是同层节点之间的互相连接，也即 cross-link；另一类是相邻层节点的互相连接，如果同一层的多个节点都与下一层的某一个节点相连，则会出现最短路径不唯一的情况，如图 3 中，第一层的 A 和 B 都与第二层的 D 有连接，因此从 monitor 到 D 有两条最短路径，而采用 USP 策略的单点探测只能发现其中的一条。图 4 是单点探测中，此类链路在

dest2

各层上的分布情况。可以看到，如果某一层节点与其上一层节点之间的链路越多，则由于最短路径不唯一，单点探测未能发现的链路也就越多。[7]通过实验指出，tracert 探测更容易发现距离探测点较近的链路。因此，将 M' 部署在链路较多的层，就可以发现较多的由于最短路径不唯一而致使 M 发现不了的链路。

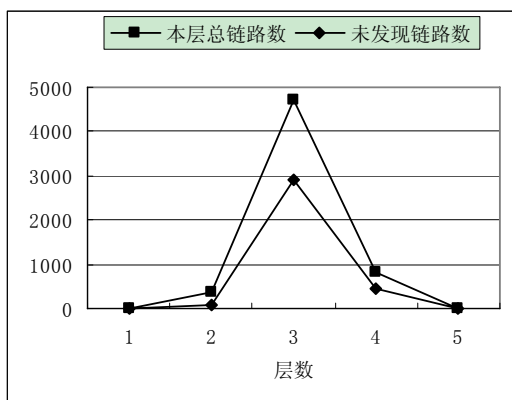


图 4 单点 traceroute 探测结果中链路的分布情况。总链路代表本层节点与上一层节点之间连接的链路，未发现链路表示在这些链路中，单点 traceroute 探测未能发现的部分

3.1.3 如何部署 M'

由上文的分析可以看到，为了发现更多 M 难以发现的 cross-link 包含的链路，应该把 M' 部署在节点较多的那一层；而为了发现更多的由于最短路径不唯一而发现不了的链路，应该把 M' 部署在与相邻层连接较多的那一层。一般情况下，如果某一层所包含的节点越多，则本层节点之间的 cross-link 以及本层节点与相邻层节点之间的连接也就越多，如图 5 所示。因此，将 M' 部署在节点较多的层，就可以发现较多 M 难以发现的链路。为了验证此结论，我们选取图 4 和图 5 实验中的探测点作为探测点 M，然后对 M 进行广度优先遍历，在其遍历的各个层上分别选取 4 个探测点作为 M'，并以 M' 为探测点进行探测，结果如图 6 所示。图 6 中的纵轴表示在 M' 的探测结果中，M 发现不了的链路数。由图 6 可以看出，在节点分布最多的第 3 层上的探测点能够发现更多的 M 难以发现的链路。

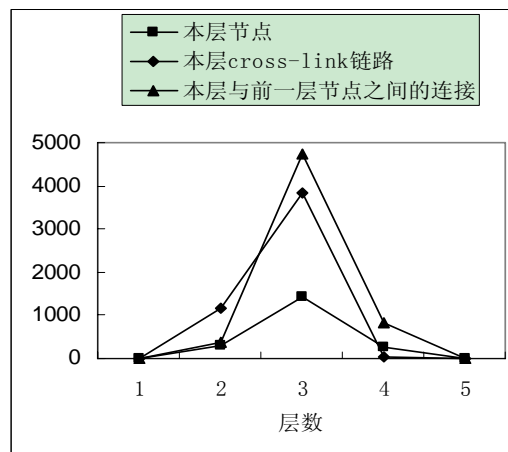


图 5 节点数与链路数的关系图

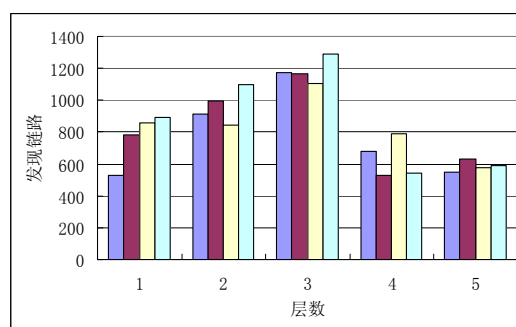


图 6 横轴表示探测点相对于 M 的层次，纵轴表示这个探测点的发现结果中，M 发现不了的链路数

在上面的实验中，我们首先假设网络中的所有节点都是目的节点，然后对单点探测发现的链路数进行分析。而如果某个探测点能够发现较多的链路，这些新发现的链路中必定包含了较多新的节点。下面我们将考虑部分节点作为目的节点的情况。在模拟图中随机选取 1000 个节点作为目的节点，进行与上节相同的实验，其节点发现情况如图 7 所示。由图 7 可以看出，将探测点 M' 部署在节点最多的层，不仅可以发现更多 M 发现不了的链路，也可以发现更多的节点。

3.2 skitter 数据分析

上一节中，我们分析了已知一个探测点 M，如何部署第二个探测点 M'，使得 M' 能够发现最多 M 难以发现的节点和链路的方法，并在模拟图上验证了此方法。本节中，我们将对 skitter[2] 的原始探测数据进行分析，进一步验证上述方法。

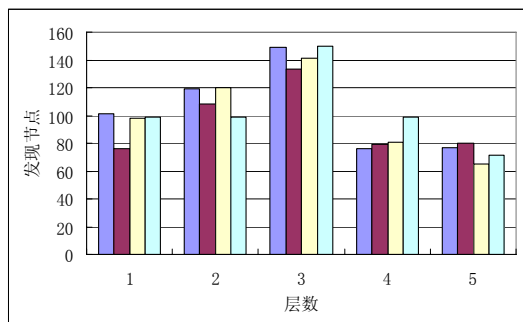


图 7 横轴表示探测点相对于 M 的层次，纵轴表示这个探测点的发现结果中，M 发现不了的节点数

skitter[2]用于 IPv4 拓扑发现的探测点共有 25 个，其中每个探测点对目的列表进行一次 traceroute 发现的周期通常为 2-3 天。由于各个探测点进行独立的 traceroute 探测，各个探测点的探测周期开始时间不尽相同。本文选取了 2006 年 11 月 2 日为探测周期开始时间的 11 个探测点的原始数据进行比较。表 3 列出了这些探测点的探测结果。

探测点名称	发现节点数	发现链路数
cdg-rssac	441117	528179
f-root	516695	613610
h-root	330862	401380
i-root	432354	511793
champagne	497823	593135
uoregon	460739	547773
arin	512445	609243
sjc	529833	630252
riesling	482590	572730
mwest	496489	588622
lhr	504501	594219

表 3 Skitter 各个探测点发现情况

由表 3 可以看出，各个探测点发现的节点和链路数量大致相同，但有少数探测点的发现结果异常。由于实验需要比较探测点之间的发现结果，为了消除一些特定于探测点的因素的影响，我们首先将发现结果较少的探测点(cdg-rssac, h-root, i-root, uoregon)去除。对 skitter 各探测点发现结果之间的比较如图 8 所示。每张图都是某个探测点 M 与其他探测点进行比较的结果(图 a、b、c 中，M 分别为 cdg-rssac、champagne 和 f-root)。

图中的四条曲线分别表示各探测点所在层上，M 发现的节点和链路的数量，以及各探测点能够发现而 M 发现不了的节点和链路的数量。从图中可以看出，四条曲线的趋势是一致的。因此，从 skitter 的原始数据可以得出，相对于某个探测点，其不同层次上的探测点的探测效率和本层已发现的节点和链路数量有相关关系。

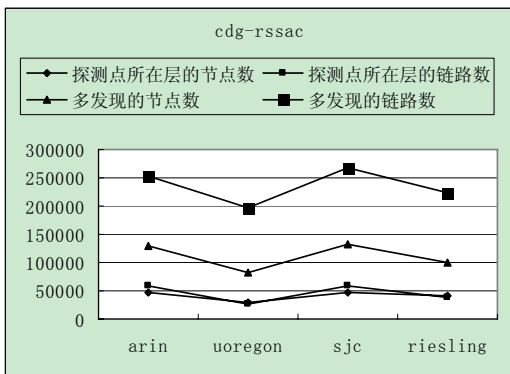
4 提出算法及实验

4.1 探测点部署算法

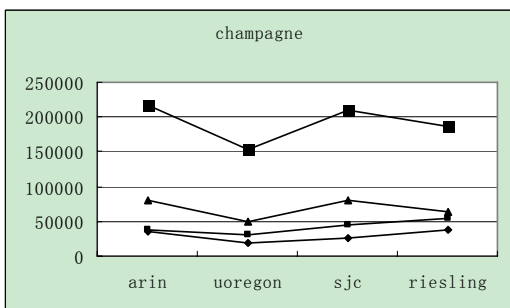
通过上一章可以得出，对于一个探测点，选取其探测结果中节点或链路最多的层上的节点作为探测点，能更有效发现其自身发现不了的节点和链路，也就是使得两个探测点的探测结果的交集最小。根据这一原理，本文提出选取探测点的迭代算法，其中中心思想是，每增加一个探测点，都要保证这个探测点的探测结果与所有已有探测点的探测结果的交集都是最小。具体步骤如下：

1. 初始化。在网络中随机选取一个节点作为初始探测点 M，利用 M 对目的节点进行 traceroute 探测，并对探测结果按照路径长度进行分类。假设 M 在第 N 跳发现了的节点最多，则初始化集合 S 为 M 在第 N 跳发现的所有节点。
2. 如果 S 不为空，则从 S 中随机选取一个节点作为下一个探测点，并把这个节点从 S 中去掉；否则，算法结束。
3. 用 2 中选出的探测点进行 traceroute 探测，对探测结果按照路径长度进行分类。假设这个探测点在第 N' 跳发现的节点最多，则初始化 S' 为此探测点在第 N' 跳发现的所有节点。
4. S 设为 S 与 S' 的交集。转入 2

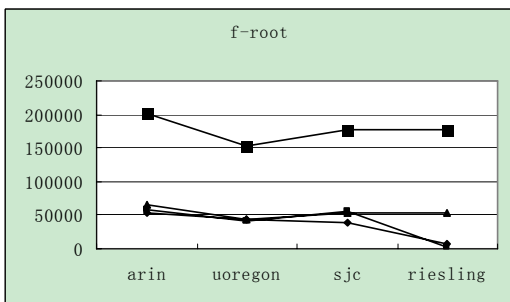
用上述算法选择探测点时，第一个探测点 M 为随机选取，集合 S 被初始化为 M 的探测结果中节点最多的层上的所有节点。此后每增加一个探测点，都将其发现结果最多的层的节点的集合与 S 做交集，以保证其后选取的探测点与所有已有探测点的探测结果的交集最小。



(a) cdg-rssac



(b) champagne



(c) f-root

图 8 skitter 各探测点发现结果比较。横轴表示与 M 进行比较的探测点

4.2 模拟网络的实验

用上一节中提出的选取探测点的迭代算法，我们对实验网络进行模拟探测。结果如图 9 和图 10 所示。可以看出，用上述迭代算法选取探测点，

每增加一个探测点，其发现结果的增加要明显好于随机选取探测点的结果。当随机选取 1000 个节点作为目的节点，探测点的数目为 15 时（图 9），用本文提出的算法选取的探测点比随机选取的探测点多发现的节点和链路分别为 57 和 1630，分别占原图的 2.8% 和 14.8%。而当图中所有节点都为目的节点（图 10），探测点的数目为 15 时，迭代算法比随机选取探测点能够多发现 1986 条链路，占总链路的 18%。由此可以看出，在探测点数量一定的条件下，使用本文提出的选取探测点的方法，能更有效的发现网络中的节点和链路。

从图 9 还可以看出，本文提出的迭代算法发现链路的效率要高于发现节点的效率。这是由于增加探测点时，traceroute 探测发现节点的边际效应本身就很低。这也就说明，如果要显著增加节点的覆盖率，必须增加目的站点的数量。

5 结论

本文在分析 traceroute 探测特性的基础上，得出了两个探测点的探测结果的差异与探测点之间相对位置的关系，并利用模拟网络与 skitter 数据进行了验证。根据此关系，本文提出了在 traceroute 探测中选取探测点的迭代算法，其中心思想是：每增加一个探测点，都要保证此探测点与所有已有探测点的交集最小。最后，本文对上述迭代算法与随机选取探测点的策略进行了比较，进一步说明了此方法的有效性。

本文提出的迭代算法可以用于指导实际 traceroute 探测中探测点的选取。从对 skitter 的原始数据(图 8)的分析可以看出，各个探测点之间探测结果的差异和他们的相对位置是有关系的。如果在选取探测点时，能够保证所有探测点两两之间的交集都为最小，则将各探测点合并后的结果中，节点和链路的覆盖率就能达到最大。

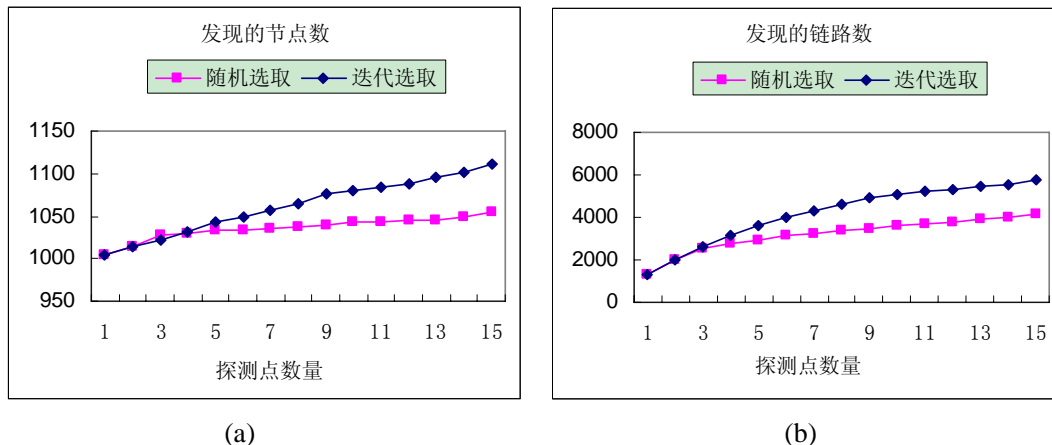


图 9 随机选取 1000 点作为目的站点，用本文的迭代算法选取探测点和随机选取探测点的探测结果对比

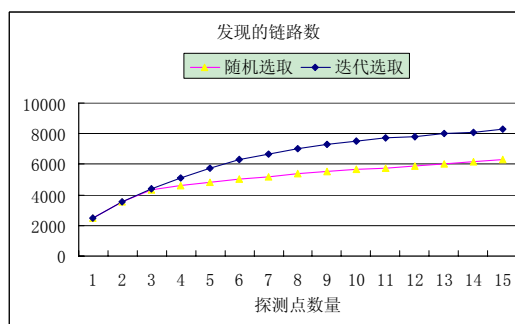


图 10 选取所有节点作为目的站点，用本文的算法选取探测点和随机选取探测点的探测结果对比

参考文献 References

[1] Whois tool. <http://www.ripe.net/ris/riswhois.html>

[2] Cooperative Association for Internet Data Analysis Skitter tool. <http://www.caida.org/tools/measurement/skitter/>

[3] The National Laboratory for Applied Network Research (NLNR). <http://moat.nlanr.net/>

[4] Topology project. <http://topology.eecs.umich.edu/>

[5] SCAN project. <http://www.isi.edu/div7/scan/>

[6] Internet mapping project at Lucent Bell Labs. <http://www.cs.bell-labs.com/who/ches/map/>

[7] A Lakhina, JW Byers, M Crovella, and P Xie. Sampling biases in IP topology measurements. In IEEE INFOCOM, 2003.

[8] DIMES project. <http://www.netdimes.org/>

[9] B Donnet, P Raoult, T Friedman, M Crovella. Efficient algorithms for large-scale topology discovery. ACM SIGMETRICS Performance Evaluation Review, 2005.

[10] Shavitt, E Shir. DIMES: Let the Internet Measure Itself. ACM SIGCOMM Computer Communication Review, 2005

[11] P Barford, A Bestavros, J Byers, M Crovella. On the Marginal Utility of Network Topology Measurements. Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, 2001.

[12] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vázquez, and A. Vespignani. Exploring networks with traceroute-like probes: Theory and imulations. Theoretical Computer Science, Special Issue on Complex Networks, 2005.

[13] JL Guillaume, M Latapy. Relevance of massively distributed explorations of the internet topology: Simulation results. In IEEE INFOCOM, 2005.

[14] S Jamin, C Jin, Y Jin, D Raz, Y Shavitt, L Zhang. On the placement of Internet Instrumentation. In IEEE INFOCOM, 2000.

- [15] Paul Francis, Sugih Jamin, Vern Paxson, Lixia Zhang, Daniel Gryniewicz, and Yixin Jin. An architecture for a global internet host distance estimation service. In IEEE INFOCOM, 1999.
- [16] C. Faloutsos, P. Faloutsos, and M. Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proceedings of the ACM SIGCOMM*, 1999.
- [17] Réka Albert and Albert-László Barabási. Topology of Evolving Networks: Local Events and Universality. *Physical Review Letters*, 2000