doi: 10.12068/j.issn.1005-3026.2020.08.003

基于 traceroute 的多特征子网发现与分析

姚 巍, 赵 海, 朱 剑, 陈香伊
(东北大学 计算机科学与工程学院, 辽宁 沈阳 110169)

摘 要: 互联网测量的研究促进了路由器级拓扑发现的发展,而网络层的子网能为其提供更详细的中间互补视图.针对子网边界条件以及完整性考虑不足引起的准确率较低问题,提出了一种多特征结合子网发现算法.研究了同一子网 IP 的 traceroute 路径特征,将多个特征结合设计更精准的子网边界判定条件.通过筛选子网的完整性,缩小候选子网的搜索空间,启发式求解子网发现问题.实验结果表明,本文算法与现有其他算法相比,能更准确地发现子网,有效地减少子网误报情况,同时效率有所提高.最后,对六个地理上分散的 ISP 进行子网推断,并分析了这些 ISP 之间常见的各种子网特征.

关 键 词: 拓扑发现;traceroute;子网发现;多特征;拓扑分析
中图分类号: TP 393.1 **文献标志码**: A **文章编号**: 1005-3026(2020)08-1075-08

Multi-characteristic Subnets Discovery and Analysis Based on Traceroute

YAO Wei, *ZHAO Hai*, *ZHU Jian*, *CHEN Xiang-yi*
(School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China. Corresponding author: YAO Wei, E-mail: yaow.neu@gmail.com)

Abstract: The studies on internet measurement have facilitated the development of router-level topology discovery, while subnets in the network layer provide a more detailed intermediate complementary view. In order to deal with the low accuracy caused by insufficient subnet boundary conditions and completeness, a multi-characteristic subnet discovery algorithm was proposed. The characteristics of the traceroute path of IP in the same subnet were studied, and were then combined to generate more precise subnet boundary determination conditions. By filtering the completeness of a subnet, the search space of the candidate subnet was narrowed, and the problem of subnet discovery was solved iteratively. The experimental results show that the proposed algorithm can discover subnets more accurately than other existing algorithms, reduce false positive rate, and improve efficiency. Finally, subnets were inferred on six geographically disperse ISPs, and the common subnet characteristics appearing in these ISPs are analyzed.

Key words: topology discovery; traceroute; subnets discovery; multiple characteristics; topology analysis

互联网作为一种大型的复杂网络结构[1],其宏观结构需从多个角度进行研究与分析.近年来互联网结构、性能和发展趋势的研究在网络安全、网络演化以及拓扑结构的统计特征等方面已取得较多成果[2-4],然而,网络规模的爆发式增长使得网络资源管理、安全性以及探测需求等方面面临巨大挑战.为满足日益增长的对网络性能的需求,人们迫切需要对互联网结构特征有更深刻的理解,这是认知网络的必然发展过程.研究表明[5],互联网的子网发现能够为路由级拓扑的探测提供一种更有效的方法,如促进链路发现及 IP 别名解析等.掌握子网级拓扑的特征和规律,将有助于为防范网络攻击提供预警信息,在一定程度上提高 Internet 服务的性能和容错能力,为构建下一代互联网提供指导性建议.

收稿日期: 2019-10-31
基金项目: 中央高校基本科研业务费专项资金资助项目(2020GFZD014,N180716019); 国家重点研发计划项目(2019JSJ12ZDYF01).
作者简介: 姚 巍(1995-),男,河南永城人,东北大学博士研究生; 赵 海(1959-),男,辽宁沈阳人,东北大学教授,博士生导师.

子网是指位于同一连接介质上并且可以在链路层彼此直接通信的一组设备(RFC 950). 子网级(subnet level)拓扑将子网表示为顶点,将路由器描述为子网的链接. Traceroute[6-7]是目前收集子网拓扑数据的主要探测工具,近年来,一些学者利用此工具主要从以下几个方面研究子网发现问题:文献[8]和文献[9]采用同一子网 TTL(time to live)差值作为判定子网边界的条件,递归地形成并检查较小的子网,直到找到最终子网;文献[10]基于对广播地址发送的 ICMP 探测包不能出现在路径信息中的原理来识别可能的候选子网;文献[11]用一种新的思路发现子网,以不同 TTL 值对每个接口发送成对探测包,依据 Ping[7]反馈消息确定子网边界;Grailet 等[12]通过引入决策树对文献[11]的算法进行优化,将子网的拓扑表示为一个树状结构,更清晰地表示子网之间的相对位置. 然而,上述研究仍存在以下不足:①在准确性方面,大部分算法仍采用单一特征,与真实情况差异较大,无法保证准确性;②在完整性方面,对网络中仅有少量 IP 给予响应的情况考虑不足;③在适用性方面,算法复杂度较高.

针对上述存在的问题,本文设计了判定子网边界的精准条件,同时考虑每个子网的存活 IP 数量以满足完整性,提出了一种多特征结合的子网发现算法. 实验证明,所提算法在真实网络上有较好的准确性和效率. 最后,对六个不同地理位置的 ISP(internet service provider)网络的子网特点进行统计与分析,加深了对子网分布规律的认识.

1 子网特征分析

1.1 相关定义与度量

从单一探测源来说,子网可以看作是一组响应接口(假设为 M 个),包括三个重要组成部分:首先,在到达目标子网之前由数据包穿过的最后一个路由器(图 1 中的 R_1,图中 s_1 为探测源);其次,属于此路由器上的子网 IP 接口(图 1 中的黑色圆圈 b);最后,目标子网中的若干其他 IP 接口,它们都具有相同的跳数(图 1 中的白色圆圈 c 和 d).

综上,给出本文所用的符号及定义如下.

定义 1 子网:子网 X_s^p 表示该子网地址为 s,子网掩码长度为 p,即该子网上所有 IP 地址共有最左边的比特数. 从实际来看,对子网内的任意两个接口 IP 地址,v_{ip} 和 w_{ip},可以被假设为两种情况:①$p(1\leqslant p\leqslant 32)$子网,则它们的最左前 p 位地址相匹配;②不同的子网,则它们的最左前 p 位地址至少有一位不能匹配.

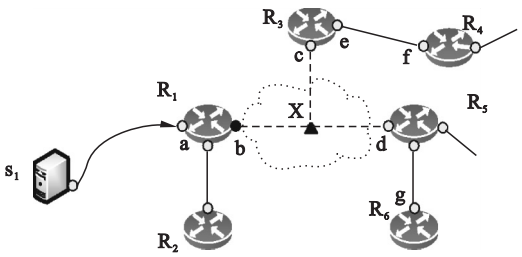

图 1 度为 3 的子网 X

Fig. 1 Subnet X with a size of 3

定义 2 子网度:子网 X 的度数 X^{degree} 表示该子网接口数量. 如图 1,子网 X 中有 3 个 IP 接口(即 b,c,d),则该子网的度为 3.

定义 3 路径:Trace(s,T_{dst}) 表示目的地址 dst 的探测路径,即$(T_{\text{dst}}:s,\cdots,R_y,R_z,\text{dst})$,其中包含沿路径的一系列 IP 接口. s 表示探测源的接口地址,R_y 和 R_z 分别表示最后两个路由器到目标地址 dst 的接口地址. 在实际中由于负载均衡等策略,Trace(s,T_{dst})可能不等于 Trace(T_{dst},s). 为了简便,本文假设 s 已知,用 Trace(T_{dst})代替 Trace(s,T_{dst}).

定义 4 枢纽接口:给定任意一个子网 X,其中都包含至少一个入口路由器的接口,称这类接口为枢纽接口;其中还包含若干个同一子网内的其他接口,称这类接口为非枢纽接口. 图 3 中,入口路由器 R_1 的接口 b 是枢纽接口,c 和 d 是非枢纽接口.

1.2 子网边界条件

以往算法仅根据子网的单一特性来推断子网,例如 TTL 距离,但实际中不同子网的地址也有可能具有相近的 TTL;因此,仅依据此单一条件无法保证子网发现的准确率. 本文从多个角度考虑子网边界的特征,设计更精准的子网边界判定条件,从而更有效地发现子网. 依据上述表达的相关定义,给出同一子网的多个特征.

特征一:同一子网中至少包含一个枢纽接口和若干个非枢纽接口,且接口彼此之间 TTL 差值不超过 1.

依据 traceroute 探测路径方向,在到达探测目标子网之前都需经过一个入口路由器的枢纽接口,而非枢纽接口为入口路由器的下一个路由器接口. 因此,非枢纽接口的 TTL 等于枢纽接口的 TTL 加 1.

如图 2 中,路由器 R_2,R_3,R_4 和 R_5 之间为多

路访问链路，R_1 到 R_2 以及 R_5 到 R_6 为点对点链路，其中 R_2 为入口路由器. 在子网 X 中，有一个枢纽接口 d 和三个非枢纽接口 e，f，g，枢纽接口 d 的 TTL = m，非枢纽接口 e，f，g 的 TTL = $m+1$，且所有接口之间 TTL 差值不超过 1.

真实环境下，路由器策略(如具有子网备用接口的路由器)可能使一个子网出现多个枢纽接口，因此为了更符合实际，子网至少包含一个枢纽接口.

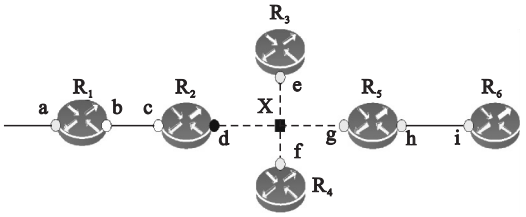

图 2　网络拓扑示例图
Fig. 2　A sample of network topology

特征二：同一子网中接口 IP 地址的路径信息在最后一跳或者两跳不同.

依据 IP 地址分配规则，若观察到的 IP 地址属于某个子网，则同一子网中所有接口都具有相同的最大 p 位前缀的 IP 地址，即子网前缀. 因此，同一子网的 Trace 具有相似的路径信息，且只有最后一个或两个 IP 信息不同.

如图 2 所示，接口 e 和接口 g 的路径分别为 Trace(a，c，e)和 Trace(a，c，g)，它们在 R_2 之前的路径是相同的，且只有最后一跳不同. 而对于接口 d 的路径 Trace(a，d)，最后两跳都不同.

实际中，多路径路由策略可能会使探测的中间路径信息发生变化，但到达目标子网之前都需经过入口路由器(图 2 中的 R_2)，因此，只需保证路径信息上倒数第三跳地址相同即可.

综上，给出判定子网的边界条件：对任意候选子网中的 IP，若子网内所有 IP 同时满足该两个特征，则 IP 属于同一子网；否则，条件不成立.

2　子网判定算法

本文算法流程如图 3 所示. 首先对探测的目标列表预处理，得到存活的 IP 列表，然后送入子网推断阶段，此过程对每个存活目标 IP 形成的候选子网进行迭代筛选，得到满足完整性和边界条件的子网.

2.1　预处理阶段

本文算法的目标是发现 Internet 中的子网级网络. 通常情况下，对目标 IP 的探测最终返回“*”信息是由于防火墙等策略使得路由对探测包不响应，从而无法获得反馈消息. 为确保最终所发现的子网的准确性和完整性，初始阶段对所有目标 IP 列表进行并发 Ping[7] 存活探测，不响应的 IP 接口不会出现在存活列表中，因此在下一个阶段中不会再次探测它们. 通过列出具有初步并行步骤的响应 IP，为后续算法步骤节省了大量时间.

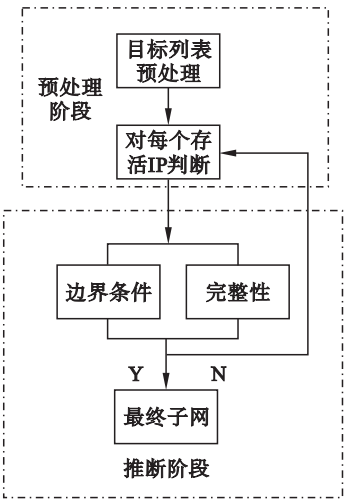

图 3　算法流程
Fig. 3　Process of the algorithm

2.2　子网推断阶段

在预处理阶段结束后，算法得到存活目标的 IP 地址列表. 下一步需对目标地址进行子网探测. 此阶段，为了提高子网推断准确率，采用启发式对子网进行探测，尽可能寻找满足子网边界条件的候选子网. 主要过程分为以下几个步骤：

①对列表中每个存活目标 t_{ip}，从/31 前缀开始递归，形成包含该目标的候选子网 X'_t(如/30，…，/24).

②对候选子网 X'_t 中所有地址 t'_{ip} 路径探测，记录 Trace(t'_{ip})和 TTL.

③候选子网边界判定：由 1.2 节可知，子网边界条件满足特征一和特征二，因此，通过对比候选子网中接口 IP 的路径相似度、TTL 差值，以及子网是否包含枢纽接口，从而判断该子网是否满足条件. 该步骤若找到某个 IP 地址不属于当前候选子网的一部分，子网收缩(前缀长度加 1，如/25→/26)，迭代停止，返回收缩后的子网；否则进入步骤④.

④候选子网进一步筛选判断：以往算法忽略子网中仅有少量 IP 响应的情况，从而将较小子网错误推断为较大子网. 例如，网络管理者为子网分配一个/28 的地址空间，但只使用了整个/28 地址空间的前两个活动 IP 地址，在这种情况下，以往算法可能会形成更大的/27 或/26 子网. 另一方

面,若子网中的部分响应地址不连续,也有可能将一个大型的子网分成几个较小子网.

为了尽可能防止此类情况发生,对候选子网响应的接口 IP 数量设定阈值:若候选子网中存活 IP 数量少于该子网所容纳数量的 1/3,则该子网收缩(前缀长度加 1),判断收缩后的子网是否满足以下情况:第一,若子网包含至少一个枢纽接口,则当前迭代停止,返回收缩后的子网;第二,若子网不包含枢纽接口,则将该候选子网舍弃,进入步骤②,对下一个候选子网进行判断.

在子网推断过程中,阈值太大会舍弃大量的子网,太小则影响子网的准确性,因此将阈值设为 1/3,一定程度上保证了子网准确性. 由于/30 和/31 子网是点对点链路,其子网本身容纳 IP 数量较少,并不适用于此情况;因此,对除/30 和/31 外的候选子网采取此策略能更好地筛选子网.

⑤重复步骤①~④,直至遍历完目标地址列表.

2.3 算法伪代码

文本算法以找到一个或多个枢纽接口的子网为原则进行探测. 在实际中,一个子网可以具有多个枢纽接口,这可能是由于网络问题(如重定向)或网络策略(例如,第一个失败的情况下,路由器为子网分配了备用接口)引起的,缺少枢纽接口可能是由于其他网络问题所导致. 判定子网算法的伪代码如下.

输入目标 IP 列表

输出子网集合 X

1) 初始化 $X \leftarrow \varnothing$

2) 预处理探测目标获得活动 IP 列表 List

3) for each t_{ip} in List do

4) for i from 31 to 20 do

5) 对目标地址 t_{ip} 构建/i 候选子网 X_t'

6) for each t_{ip}' in X_t' do

7) Trace(t_{ip}')并记录其 TTL 值

8) end for

9) if X_t' 满足特征一与特征二 then

10) 记录中心接口

11) else

12) $|X_t'| \leftarrow i+1, X \leftarrow X_t' \cup X$

13) break

14) end if

15) if $|X_t'| < (2^{32-i}))/3$ then

16) if X_t'包含枢纽接口 then

17) $|X_t'| \leftarrow i+1, X \leftarrow X_t' \cup X$

18) break

19) else

20) break

21) end if

22) end if

23) end for

24) end for

步骤 2)为预处理阶段,步骤 3)~24)表示整个子网推断阶段①~⑤,其中步骤 4)~13)表示子网推断阶段的①~③,步骤 15)~23)表示候选子网筛选过程④. 假设探测目标 IP 数量为 C,预处理阶段时间开销为 $O(C)$,预处理后存活目标 IP 数量为 N,每个存活目标 IP 遍历次数为 k,子网推断阶段时间开销为 $O(kN\lg N)$,则算法总开销为 $O(kN\lg N + C)$.

3 实验与分析

本实验采用一个真实拓扑数据集[12](AS4711),从该数据集中选取 355 个真实子网和部分不是真实的子网作为实验数据,子网大小从/24到/31. 从这些子网中选取若干 IP 地址构成目标列表,并将其提供给本文算法作为输入. 本实验所使用的处理器为 Intel i7 - 8700 3. 19GHz,内存 8GB,采用 linux 系统,编程语言为 golang. 为了提高算法时效性,实验中对目标 IP 设置 2s 的响应等待时间,每个 IP 探测 3 次,采用 ICMP 协议探测.

3.1 准确性和效率分析

为了体现该算法的准确性,本文选取了现有的三种主流算法 Cheleby[9],TreeNet[12] 和 XNet[11] 进行比较. 实验结果表明,采用本文算法发现子网的精确率最高,可达 76% ,远高于其他三种算法. 本文算法虽然召回率较低,但本文以寻找正确的子网为目标,因此误报率方面远好于其他三种算法,有效地减少了子网误报的情况. 在运行时间方面,本文算法也少于其他算法. 因此,本文在保证整体评估指标优于其他算法的前提下,时间效率也得到一定的提高.

子网推断分布情况如表 1 所示. 其中第一行表示真实子网分布情况,最后一列是每个方法得到的子网总数量. 整体来看,本文算法子网发现数量最少,究其原因是点对点链路(/30 和/31 子网)的数量较少,因为在探测过程中本文无法找到一部分包含枢纽接口的点对点链路,同时遗漏了对部分子网的探测扫描,从而导致了此情况的发生. 另外,针对部分没有被发现的/24 子网,本文算法错误地将其分成了更小的子网. 这种情况的发生一部分是由于防火墙过滤掉了探测包,导

致舍弃了完全无响应的子网，或者只利用小部分响应地址来推断子网. TreeNet 算法也能较准确地发现较大的子网，这是由于该算法对候选子网构建决策树，对错过枢纽接口的候选子网进行二次探测，保证了子网发现的准确性，但仍有部分/30 和/31 子网没能被发现，原因和本文类似. XNet 算法对具有少量地址响应的较大子网没有进一步筛选判断，忽略了该情况，并且认为子网只存在唯一的枢纽接口，因此它倾向于将大型子网分成多个较小(不完整)的子网. 例如，用本文方法探测出 203 个子网，其中/24 子网正确推断出 2 个；而 XNet 算法所发现的子网不适合真实情况下各种规模的子网，因而被分成更多的/25，/26，/27，/29 和/30 子网，导致子网被低估. 针对部分点对点链路 XNet 算法也发生推断错误. Cheleby 算法没有考虑枢纽接口的存在，从较大前缀长度往后递归形成候选子网(本文方法是从后往前递归)，探测过程中没有对部分响应的子网进行判断，从而也误将实际较大(/24)的子网分成多个较小(/25，/26，/27 和/30)的子网；同时，该算法仅依靠子网中接口 TTL 差值不超过 1 的距离条件，将一小部分较小(/30 和/31)的子网合并推断成较大的/28 和/29 子网. XNet 和 Cheleby 算法对子网的全面扫描能力较强，一定程度地减少了漏探情况. 另一方面，由于防火墙策略导致路由器不响应，本文与其他三种算法均不能识别部分子网.

表 1　子网发现数量比较

Table 1　Comparison of the number of subnets discovered

算法	/24	/25	/26	/27	/28	/29	/30	/31	总数量
本文	2	0	4	12	20	75	69	21	203
Cheleby	1	9	20	31	48	91	112	74	386
XNet	0	13	18	35	55	109	125	98	453
TreeNet	2	1	3	25	35	64	85	40	255
真实情况	4	0	4	3	23	74	145	80	355

图 4 统计了不同方法判断正确的子网数量和判断错误的子网数量占比情况. 其中，将正确子网又分为完全响应子网和部分响应子网. 从图中看，相比其他方法，本文方法发现部分响应的子网占比最多，尤其发现的较大子网数量较多，这要归功于设定的阈值筛选条件. 同时，Cheleby 和 XNet 算法占比情况类似，完全响应的子网比例大于部分响应的子网比例，因为它们本质上都没有对部分响应子网作进一步考虑，导致对部分响应子网判断正确的数量较少. TreeNet 算法对可能的候选子网进行二次探测，也能有效地识别出部分响应的子网.

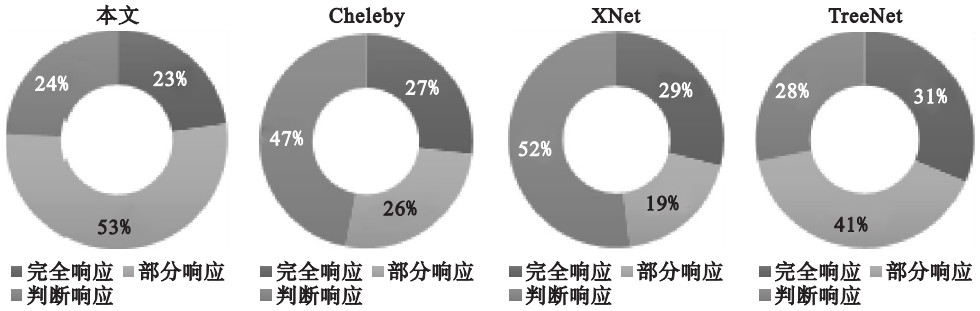

图 4　子网结果分类

Fig. 4　Taxonomy of subnet results

为了更全面有效地评价本文和其他算法，采用精确率、召回率、误报率和 F 值四项评估指标，结果如表 2 所示.

表 2　子网评估指标比较

Table 2　Comparison of the evaluation index of subnets

%

评估指标	本文算法	Cheleby	XNet	TreeNet
精确率	75.9	52.6	47.9	71.8
召回率	43.4	57.2	61.1	51.5
误报率	18.3	39.6	35.9	19.5
F 值	55.2	54.8	53.7	59.9

精确率(precision)描述的是在子网发现结果中，探测正确的子网所占比例：

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{1}$$

召回率(recall)表示在实际子网中，被正确判定为真实子网所占比例：

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2}$$

误报率(false positive rate, FPR)描述在实际不是真实的子网中，被错误判定为真实子网的比例：

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{3}$$

F 值(F-score)是一项综合考虑 precision 和

recall 的指标:

$$F\text{-score} = \frac{2\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (4)$$

在上述评估指标中,TP 表示实际存在且探测结果也存在的子网;FP 表示实际不存在但探测结果为存在的子网;FN 表示实际存在但探测结果为不存在的子网;TN 表示实际不存在且探测结果也不存在的子网.

从精确率来看,本文算法对候选子网进行多特征判断,把不满足阈值条件的部分响应子网过滤掉,从而误报数量 FP 较少,最终正确的子网 TP 所占探测结果的比例最高;同样,TreeNet 算法误报数量 FP 也较少,保证了一定的精确率. 与此相反,Cheleby 和 XNet 算法没有针对响应不完整的子网作进一步判断,错误地将较大子网分解成多个更小的子网,导致 FP 数量增大,最终子网数量较多,精确率较低. 在误报率方面,本文算法最低,低于 TreeNet 算法 1.2% ;XNet 算法最高,几乎是本文算法的两倍. 本文针对部分响应接口的子网通过更严格的筛选,一定程度地减少了正确子网推断错误的情况. TreeNet 算法利用决策树表示出子网的相对位置,对候选子网增加额外探测,从而子网误报率也较低. 相比之下,XNet 和 Cheleby 算法误报率较高,这主要是二者误将仅有少量响应接口的较大子网分成了若干较小子网. 召回率方面,本文算法最低,且较其他算法的总子网数少,这是由于一部分子网的漏探,以及对候选子网的筛选设置所导致. 而 XNet 算法更多地发现了/30和/31 子网,TP 数量超过其他三种算法,因此真实子网数量占比也最多,召回率最高. F 值是综合体现精确率和召回率的一项调和评估指标,因此能够均衡客观地展现算法的有效性. XNet 算法虽然召回率较高,但在精确率方面远远低于其他方法,因此 F 值最低. 本文算法由于精确率的升高,导致其召回率较低,因为,在候选子网范围经过严格的筛选后,误报数量 FP 减少,漏报数量 FN 增加. 本文算法的 F 值略低于 TreeNet 算法,但优于其他二种方法.

对每个算法运行 50 次,平均每次运行时间结果如表 3 所示. 分析可知,Cheleby 算法在搜索候选子网时,仅舍弃不满足距离条件的子网,因此,Cheleby 算法的运行时间大大缩短. TreeNet 算法对可能的候选子网进行二次探测,增大了探测代价,时间复杂度较高. 本文算法探测消耗时间虽然略高于 Cheleby 算法,但远远低于 TreeNet 算法(TreeNet 算法是本文算法的 1.4 倍),充分说明了该算法在时间复杂度指标上的优越性.

表 3 运行时间比较

Table 3 Comparison of time cost

算法	本文	Cheleby	XNet	TreeNet
平均完成时间/s	676	631	749	983

3.2 完整性分析

子网的完整性一定程度地体现其准确性. 由前文分析可知,由少量 IP 推断的子网均可能被高估或低估,因此,本节研究子网推断阶段完整性的影响,用完整率表征.

/24 ~ /29 子网数量的完整率分布情况如图 5 所示. 横坐标代表当前子网的数量,纵坐标代表子网完整率. 完整率定义如下:

$$\text{完整率} = \frac{\text{存活 IP 数量}}{\text{当前子网容纳 IP 数量}}.$$

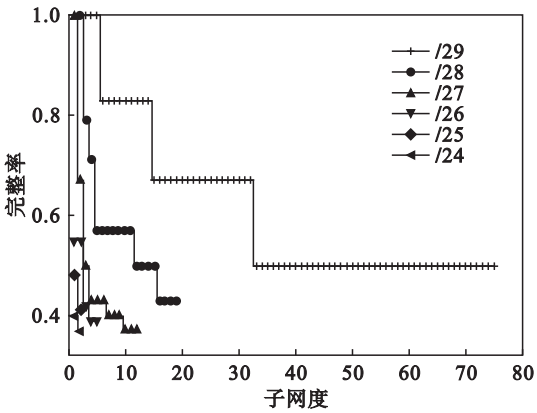

图 5 子网完整性分布

Fig. 5 Distribution of subnet completeness

/30 和/31 子网的完整率始终为 100% ,因此在图中没有标出. 从图 5 可知,/29 子网的完整率最高,子网数量也最多,/24 子网的完整率最低,仅为 37% . 随着子网前缀长度的缩短,其地址空间容量越来越大,响应 IP 比例越来越少. /27 子网的完整性分布较为均匀. 此外,若将本文算法的阈值设置为 50% ,可以发现/24 和/25 子网将会全部被舍弃.

3.3 不同 ISP 子网分析

为了分析 Internet 中子网分布特征,首先对六个地理位置分散(十万到百万级 IP 数量)的 AS(autonomous system)级网络进行子网推断;AS 级网络包括 Cisco Systems Inc. (AS - 1),AAPT Limited (AS - 2),TATA Communications (AS - 3),Orange S. A. (AS - 4), Internap Corporation (AS - 5), CERNET2 IX at Northeast University(AS - 6). 对 AS 级网络的子网推断结果如表 4 所示.

表 4　子网推断结果
Table 4　Results of subnets discovery

参数	AS－1	AS－2	AS－3	AS－4	AS－5	AS－6
IP 数	1 165 568	844 544	551 680	784 384	93 952	464 896
响应 IP	9 580	75 484	52 520	21 617	8 131	2 322
子网数	1 053	4 007	3 792	747	738	104

整体来看,尽管每个 AS 有十万到百万的 IP 数量,实际环境中的防火墙以及路由策略过滤掉了大量 ICMP 报文,导致探测源无法收到响应消息,仅有少部分的响应 IP. 对比可以发现,AS－2 的响应 IP 数量最多,发现的子网也最多. 相反,AS－6 发现子网数最少.

其次,考察子网度分布的幂律性,对不同 AS 的子网进行统计,得到补累计分布函数(complementary cumulative distribution function, CCDF)与度(degree)的关系,如图 6 所示. 为了更清晰地看到每个 AS 内的子网度分布,对局部曲线进行缩放. 由图 6 可以看到,大部分子网的度为 2,其中 AS－1 的 CCDF 曲线在度为 2 开始时为 51% ,AS－5 的 CCDF 曲线则从 77% 开始,其余部分在中间取值,这表明度的分布很集中. 在子网度高的部分其 CCDF 值较小,曲线近似幂律分布.

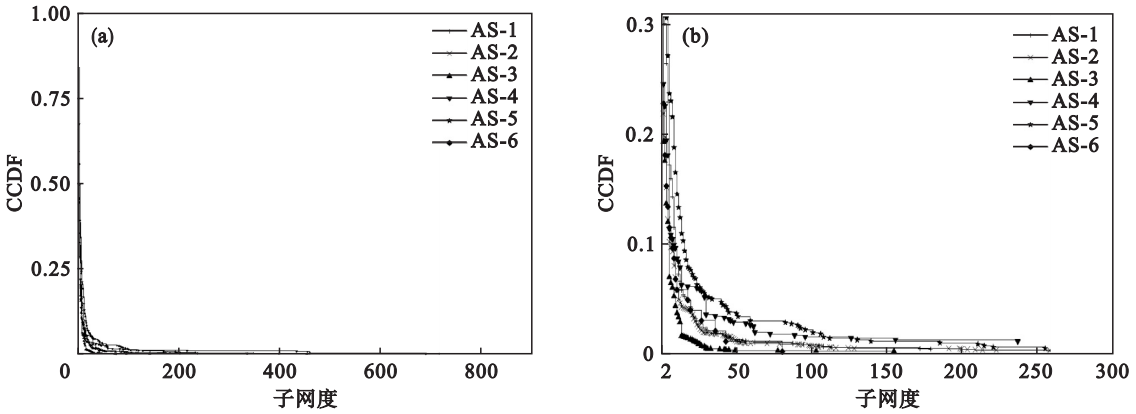

图 6　子网度分布
Fig. 6　Distribution of subnet degree
(a)—子网度分布;(b)—局部放大图.

最后,统计子网度的各项指标,各个 AS 级网络的子网差异性如表 5 所示. 在 6 个 AS 级网络中,AS－4 和 AS－5 具有最高的平均度值,这是由于其拥有的子网数量较少,同时又含有较大前缀长度的子网. 另一方面,与 AS－5 中的小型子网相比,其本身度较大的子网数量并不多,因此 AS－5 子网度的标准差较大;而 AS－4 子网度的标准差低于 AS－5. 此外,对 AS－2 和 AS－3 的子网度占比进行统计,发现 AS－3 中 68. 6% 的子网度为 2,且只有一个前缀长度为/23 的子网;因此,AS－3 的子网度均值和中位数最为接近,最小的标准差说明了该网络有着更稳定的度分布. AS－2虽然有度最大(3009)的子网,但统计得到 57. 4% 的子网度为 2 且子网数量较多,最终导致度的平均值不高,其标准差较大的原因和 AS－4 相同. AS－6 中子网的指标都低于其他 AS 级网络,这主要是因为发现的子网数量较少,且最大子网前缀长度仅为/25.

表 5　子网度统计指标结果
Table 5　Statistical results of subnet degree

指标	AS－1	AS－2	AS－3	AS－4	AS－5	AS－6
平均值	7. 68	6. 69	4. 25	11. 56	11. 13	5. 73
中位数	4	3	3	3	4	3
最大值	784	3 009	497	755	637	82
最小值	2	2	2	2	2	2
标准差	29. 12	53. 8	9. 51	54. 38	28. 3	10. 04

为了验证 AS－2 中度最大子网的准确性,对子网内响应 IP 进行 DNS 名称解析查询,发现所有 IP 地址的 DNS 名称与 a184－51－80－201. deploy. static. akamaitechnologies. com. 中的 a184－51 共享相同的前缀,且都属于 Akamai Technologies.

综上可知,大多数子网的度为 2(/30 或/31

子网),同时也接近中位数,可见子网在构建时趋向于点对点链路;因此,子网的度分布是高度倾斜的,尾部符合幂律分布. 另外,不同于其他 AS 级网络,AS-2 拥有少部分度较大的子网.

4 结 语

为解决子网发现过程中因仅采用单一特性导致探测准确率低且适用性不足的问题,本文提出一种多特征结合的子网发现算法. 该算法考虑多个特征作为子网边界,建立更精准的判定条件;同时,通过对完整性不足的候选子网作进一步有效筛选,缩小了搜索范围,从而在提高准确性的同时保证了探测效率. 另外,本文对于在真实世界中不同子网规律的研究与分析,可以为构建下一代互联网提供指导性建议.

参考文献:

[1] Yang B, Lu Y, Zhu K, et al. Evolution of the Internet and its measures [C]//2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS). Harbin, 2017: 1-4.

[2] Ding W, Yan Z, Deng R H. A survey on future Internet security architectures [J]. *IEEE Access*, 2016, 2016(4): 4374-4393.

[3] Basheir L. Power-law degree distribution consistency in the AS-level Internet topology [C]//2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). Khartoum, 2018: 1-5.

[4] 刘晓,赵海,张君,等. 互联网拓扑结构中的弹性网络特征[J]. 东北大学学报(自然科学版),2016,37(4):486-490.
(Liu Xiao, Zhao Hai, Zhang Jun, et al. Elastic network characteristics in Internet topology [J]. *Journal of Northeastern University (Natural Science)*, 2016, 37(4): 486-490.)

[5] Bakhshaliyev K, Canbaz M A, Gunes M. Investigating characteristics of Internet paths [J/OL]. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, 2019 [2019-09-16]. https://www.researchgate.net/publication/335286599_Investigating_Characteristics_of_Internet_Paths.

[6] Hu H, Liu W, Fei G, et al. A novel method for router-to-AS mapping based on graph community discovery [J]. *Information*, 2019, 10(3): 87-102.

[7] Motamedi R, Rejaie R, Willinger W. A survey of techniques for Internet topology discovery [J]. *IEEE Communications Surveys & Tutorials*, 2015, 17(2): 1044-1065.

[8] Gunes M H, Sarac K. Resolving IP aliases in building traceroute-based Internet maps [J]. *IEEE/ACM Transactions on Networking*, 2009, 17(6): 1738-1751.

[9] Kardes H, Gunes M, Oz T. Cheleby: a subnet-level Internet topology mapping system [C]//2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012). Bangalore, India, 2012: 1-10.

[10] Sun M M, Wang J L. Discovering subnets in router-level topology studies [C]//2011 IEEE 3rd International Conference on Communication Software and Networks. Xi'an, 2011: 68-72.

[11] Tozal M E, Sarac K. Subnet level network topology mapping [C/OL]//30th IEEE International Performance Computing and Communications Conference. Orlando, 2011: 1-8.

[12] Grailet J F, Tarissan F, Donnet B. TreeNET: discovering and connecting subnets [C/OL]//8th International Workshop on Traffic Monitoring and Analysis (TMA). Louvain La Neuve, Belgium, 2016 [2019-09-18]. https://www.researchgate.net/publication/308164876_TreeNET_Discovering_and_Connecting_Subnets.