目录

[1. Interface level 2](#_Toc445995763)

[1.1 traceroute——IXPs数据完整映射 2](#_Toc445995764)

[1.2 Reverse traceroute 3](#_Toc445995765)

[1.3 测量平台——Dasu 5](#_Toc445995766)

[1.4 子网发现——Tracenet 6](#_Toc445995767)

[1.5 子网发现——Xnet 7](#_Toc445995768)

[1.6 MPLS研究 8](#_Toc445995769)

[2 Router level 10](#_Toc445995770)

[2.1 别名解析——Discarte 10](#_Toc445995771)

[2.2 别名解析——Pythia 11](#_Toc445995772)

[2.3 路由发现——MRINFO 12](#_Toc445995773)

[3 AS level 14](#_Toc445995774)

[3.1 利用traceroute拓展AS拓扑图 14](#_Toc445995775)

[4 测量工具Zmap 15](#_Toc445995776)

# Interface level

## traceroute——IXPs数据完整映射

参考文献：

[35] B. Augustin, B. Krishnamurthy, and W. Willinger, “IXPs: Mapped?” in Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. ACM, 2009, pp. 336–349.

**文章题目：IXPs数据完整映射**

**by 张晔**

之前的IXPS数据统计由于AS间的协议或者商业规则，遗漏一些peering关系，我们的方法发现了超过44k的peering关系，比以前多出了18K，

整理数据集，确定IXPs的映射关系：

1.通过BGPtable归纳

2.启发式多数选择（如果一个IP总是走固定的IXP，我们把它归属的AS分配到该IXP中）

3.DNS 反向解析（如果IXP只响应探针一次，解析出的AS属于该IXP）

相比以前我们采用新的traceroute策略

1. 先列出已知IXPs与IXP前缀
2. 源服务器选择：如果一个LG（looking glass）server发现新的IXP，我们就尽量选择这个LGs作为源服务器，我们选择更靠近新IXP的LGs作为源服务器，因为它更容易发现新的伙伴。
3. 目的IP选择：我们选择路由前缀的IP作为目的IP，这种方法可以避免回路，过慢响应，大部分情况都可以找到这种IP
4. 最后整理数据，实现对LGserver的自动化查询。

其他策略：为了减少对LGserver的依赖，使用LSRR( Loose Source Record Route)策略,让它必须走某些IXP，这种方法的缺点是LSSR可能受到阻碍无法通过，LSSR和路由探针可能被一些路由器丢弃。

实验数据：我们整理了IXP的自身信息，Members，peering与cost并与CAIDA，planetLab给出的数据比较，我们更新了更多的IXP前缀和peering关系。

验证分析结果：大部分新的IXP我们无法到达与追踪，通过BGP映射提取peering关系更精确，我们用提取出来的peering表给peering关系分级，我们更相信级别高的peering关系是可信的，我们还统计了每个IXP中的peering等级，大部分都是级别高值得信赖的peering关系。我们也证实了发现额外20%的peering，BGPtable中数据大部分都可以被核实，DNS表数据有些不准确。

缺点就是太过依赖LGs 与LSRR-capable路由，我们无法发现一个AS（没有LG或者他们的一级二级邻居都没有LG）的伙伴，有些AS之间还有可能建立专用对等通道，我们无法检测到这种关系，或者本身有配置错误，AS之间的流量交换太少。

## Reverse traceroute

参考文献：

[50] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. Van Wesep, T. E. Anderson, and A. Krishnamurthy, “Reverse traceroute,” in NSDI, vol. 10, 2010, pp. 219–234.

**文章题目：反向traceroute**

**by 郝科委**

Traceroute是当今使用最为广泛的网络诊断工具，网络运营商使用它来帮助识别路由故障，性能优劣，路由器配置错误。研究人员用它来映射互联网结构，预测性能，定位路由器和ISP的性能分类。但是，Traceroute有一个影响所有这些应用的根本限制：不提供反向路径信息。虽然许多互联网上的Traceroute服务器提供了一些可见性，但是没有一般方法来从任意目的地确定一个反向路径。

在本文，我们通过建立一个reverse traceroute系统解决了这个长期存在的问题。该系统提供和Traceroute同样的信息，但是却是反向的路径。在实际测试中，在平均情况下我们的工具命中了87%的路径节点，而当前通常的手段只能达到38%。Traceroute运行时是一个单独的程序，而reverse traceroute系统是一个分布式的系统包含了几十个到几百个有利的观察节点。和标准的traceroute对目标的需求完全一样，不需要对目的节点的控制，不需要路由器和其他网络设备提供新的功能，所以可以用在任何目标。

系统使用多种方法测量反向路由跳，递增的将它们拼接起来，构造成一条反向路径。其主要技术和实现步骤如下：

1. 测量观测节点到数据源的路径，得到一个已知的网络拓扑图，因为网络路由选择是基于目标的，所以一旦从目标到源头的路径经过该拓扑图，那么就可以推导出剩下到源头的路径。
2. 使用3种主要的测量技术，从目标向后构造路径直到与步骤1得到的网络拓扑相交。
   1. 由于网络路由选择是基于目标的，所以我们可以将路由跳转一个一个的拼接起来。
   2. 采用IP timestamp（TS）和 record route options（RR）来识别沿着反向路径的路由跳。

RR-Ping：源节点向目标节点发送一个设置了RR选项的ICMP Echo Request，当目的节点发送响应包时，在反向路径上路由器就会记录一些路由信息，帮助测量反向路径。

TS-Query-Ping： 假设目的IP为D，设一个IP值R，源节点向目的节点发送一个设置了D和R的timestamp query的ICMP ping，到达D时会打上时间戳，响应包如果经过R就会再次打上R的时间戳，所以，如果源节点收到的响应里有R的时间戳，那么就说明R节点在反向路径上。

* 1. 在观测节点之间使用了源地址欺骗，RR-Ping的方法有距离限制，只有小于9跳的目标，才可以使用该方法，所以使用源地址欺骗的方法，让距离目标更近的观测节点伪装为源节点的IP进行RR-Ping，这样就克服了距离的限制，从而利用位置最好的观测节点更好的进行测量。

1. 将目标到已知网络拓扑的路径与相交点到数据源的路径合并，最终得到完整的反向路径。

## 测量平台——Dasu

参考文献：

[46] M. A. Sanchez, J. S. Otto, Z. S. Bischof, D. R. Cho\_nes, F. E. Bustamante, B. Krishnamurthy, and W. Willinger, "Dasu: Pushing experiments to the Internet's edge," in Proc. of USENIX NSDI, 2013.

**文章题目：Dasu：将实验推向互联网边界**

**by 秦超逸**

文章主要介绍了一个测量实验平台Dasu，他支持控制网络实验和宽带描述，并为互联网测量实验提供数据。

由于P2P技术的广泛使用，Dasu被构建为最流行的P2P系统——BitTorrent的一个扩展。由于BitTorrent的使用模式和很长的会话时间，Dasu能够保证自己可以几乎连续地运行测量实验。同时，BitTorrent的流行性可以保证Dasu能够获得足够规模和覆盖范围的数据。

Dasu由大量客户端的一个分布式采集和一系列管理服务组成。Dasu客户端提供了期望的覆盖范围，并参与测量，这些测量为宽带特型和互联网实验提供数据。管理服务包括配置服务、实验管理服务、协调服务和数据服务。这些管理服务分配客户端配置和实验信息，并管理数据收集。

初始化时，客户端使用配置服务宣布自己，并获得多种配置设置(包括测量的频率和持续时间，以及实验结果发给的地址)。Dasu客户端定期地与实验管理服务(用于分配测量任务)和协调服务联系来提交关于完成的测量的更新，并取回对不同实验任务的测量限制。最后，客户端使用数据服务来报告完成的实验的结果。

Dasu对于实验规格使用了一个基于规则的陈述模型。在这个模型中，一个规则是一个简单的when-then结构，该结构说明了在保持特定激活条件时需要运行的活动集。规则的左侧是条件部分(when)，说明了需要匹配的状态；右侧是规则的结果或活动部分(then)，也就是将要运行的活动清单。一些规则形成了一个程序，而一系列相关的程序定义了一个实验。

Dasu通过实验管理服务管理并发的实验，实验管理服务根据实验要求和客户端的特点将任务分配给客户端。协调服务能够异步地调用测量元，同时控制负载并保证志愿主机的安全性(控制测量实验对主机的网络和系统资源的影响)。数据服务用于收集从测量元中返回的测量数据。

文章介绍的Dasu是一个动态可扩展的平台，旨在促进互联网的测量实验，同时控制主机上的资源影响和底层网络。Dasu关键的修改是选择了一个编程接口，该接口既是灵活的(例如：支持大范围的实验)、也是安全的(例如：不允许失控的程序)，该接口使得复杂的、协调的测量成为可能。

## 子网发现——Tracenet

参考文献：

[53] M. Tozal and K. Sarac, \Tracenet: an Internet topology data collector," in Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM, 2010, pp. 356-368.

**文章题目：子网拓扑映射——tracenet**

**by 陶天一**

与traceroute相比，tracenet能够收集一条点对点路径中更完整的拓扑信息。也就是说，traceroute返回路径中转发路由接口的IP地址，tracenet可以访问到路由间的子网，并且标注IP地址之间的关系和通过测试获得的子网掩码。设A、B、C、D为四个路由器，AC、BD是两条链路且相交于一点。若A为起点，C为终点，traceroute不会发现B和D的存在，而tracenet能够捕捉到A、B、C、D间形成的子网，从而发现B、D这两个路由。

为了完全的探索子网拓扑结构，tracenet首先生成了一个暂时性的子网S，设定其掩码为/31，这样只包括了当前节点与目的节点两者的IP，通过递减掩码的值，来获得更多的IP地址，同时进行IP存在性判定（节点未应答则判定为IP不存在）。若发现当前网段某IP不存在，则返回掩码值加一作为所探测的子网S的掩码。然而ICMP请求存在转发的情况，非直接的应答说明此IP可能不属于当前子网，故tracenet对IP地址会做位置确定的工作，来保证发掘出的IP地址不会越出当前子网边界，倘若发现IP越界，则停止当前探索，返回值为当前掩码值加一。Tracenet对于子网的探索完成度是基于点对点的路径条数的，路径越多则tracenet获得的结果越准确。

在测试工作中，tracenet从三个节点A、B、C针对同一个目标IP地址数据集进行探测，这些IP地址属于四个商业ISP。结果显示ABC三点获得的数据有百分之六十是相同的，其中B、C两点获得的数据有百分之八十重合。也就是说，网络中还存在另外某点可获得更准确的测量值。同时也说明了使用tracenet工具需要一定量的起始节点数目，才能更加完整、准确的进行子网拓扑探测。

## 子网发现——Xnet

参考文献：

[52] M. E. Tozal and K. Sarac, \Subnet level network topology mapping," in Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International. IEEE, 2011, pp. 1-8.

**文章题目：子网拓扑映射-XNET工具**

**by 陶天一**

因特网的网络层拓扑包含了路由和子网两个部分，也就是点对点连接或者是多路连接。网络测量的研究较为关注路由层面的映射并且分析平均度、聚类系数和中间状态等路由特性。考虑到子网在网络拓扑的构建中也起到了关键作用，对于子网层映射的研究有着一定的重要性。这一层级将子网当作顶点，将路由描述成边，连接各个顶点。

traceNET和exploreNET都是用于子网发掘的工具，然而exploreNET建立在traceNET的基础之上，弥补了前者的不足之处。traceNET可以收集因特网中两台主机间的子网信息。但是在几种典型的网络拓扑结构中不能发挥太大的作用，这其中具有代表性的就是AS体系中hot-potato策略了，发掘结果较正确结果相比缺少了很多浅存子网信息。exploreNET用于发现独立的网络层子网，能够发现潜在的子网层拓扑映射而不是根据动态路由信息来获取映射关系。

向exploreNET指定目标IP地址t，程序通过一定量的请求数据包（ICMP请求）来发掘子网S在包含t的前提下，还囊括了哪些IP地址，同时反馈子网掩码的值。简单的来说，目标IP地址确定后，exploreNet从前缀为/31开始，对于该前缀包含的每一个子网中的IP，发送数据包来验证该IP是否存在，并且确认该IP是否在子网边界以内。当所有的IP地址通过测试后，前缀的值进行减一操作，重复之前过程，验证新子网掩码包含的IP地址的存在性。若当前某IP被确定不存在，则此时子网掩码值进行加一操作，从而完成探索过程。

ExploreNET的测试建立在Pansiot提供的数据集基础（M）上，所收集数据的正确性由Mrinfo进行验证。可信数据由很多子网组成，它们归属于全球不同的ISP（因特网服务提供商）。M包含5536个不同的子网。随机挑选一个IP地址作为输入，exploreNET运行结果可分为6大类由A-F表示。A类为完全无法响应的子网，即这些子网对于ICMP请求不作应答。B类包括了为被正确发现的子网，共计3702个。C类中，exploreNET对于673个子网无法找到合适的主接口进行访问。D类中142个子网被exploreNET发现但是Mrinfo没有任何记录。E类中，196个子网提供的IP地址未响应。F类中55个子网未被Mrinfo记录且exploreNET的结果显示它们包含的IP地址对请求不作回应。

## MPLS研究

参考文献：

[31] J. Sommers, P. Barford, and B. Eriksson, “On the prevalence and characteristics of MPLS deployments in the open Internet,” in Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. ACM, 2011, pp. 445–462.

**文章题目：开放互联网中MPLS部署的现状与特性**

**by 李秉睿**

文章利用CAIDA Ark在08-11年的路径测量数据作为研究数据集，利用traceroute支持MPLS发现的特性，对MPLS部署的现状和特性进行分析，研究MPLS的演变，并根据数据分析观察得到的结果作为依据提出判断MPLS接口和MPLS路径的方法。

因为MPLS隧道有两种模式：uniform mode和pipe mode，因为只有运行在前者模式下的MPLS隧道才可被traceroute发现，故文章主要分析的是在该模式下的数据集，因此文章最终采用08年6月到11年8月中traceroute数据（08年上半年数据为pipe mode）。

文章利用CAIDA的AS前缀来进行IP-to-AS匹配，并用UCLA Cyclops 数据对AS类型进行识别。然后对处理过的数据进行基于AS的分析，识别部署MPLS的AS的数量和特征，发现以下特点：

1. 配置MPLS的AS数量基本稳定，占总数7%
2. AS类型自顶（Tier-1）向下（Stub）配置MPLS数量递减，但每一类型数量的百分比都很稳定，Tier-1基本都有，stub只占5%
3. 所有AS中总MPLS隧道数在08年锐减，09年中反弹并持续上升，目前为止共350k个隧道，该趋势与道琼斯指数变化惊人相似，推测MPLS与经济息息相关，并认为可能是经济原因引发电信领域合并
4. 隧道少于10个的AS数量常年占20%，不超过200个的占80%
5. 大的网络更可能出现在端到端的连接中，且部署MPLS

随后文章对所有穿过某个AS且在该AS内部穿且仅穿过一个隧道的路径进行分析，将穿过AS且在AS内部的路径分为隧道前，隧道中，隧道后三部分，对每部分的跳数进行统计分析发现：

1. 只有隧道中的跳数在三年期间逐渐减少，从平均超过4跳变为3跳
2. 对部分AS进行隧道长度分析发现很多长度为1的隧道，并解释可能是因为配置penultimate hop-popping（PHP）造成的现象
3. 发现90%的隧道时延在150ms以下
4. 对嵌套隧道分析， 发现77-90%AS最大嵌套数为1，10-20%为2

文章通过对比非MPLS的数据发现MPLS的接口时延差相对比non-MPLS的更接近0，MPLS的接口的IP前缀相对于non-MPLS的更长,并根据以上结果建立模型：

1. 提出了贝叶斯数据融合方法来判断接口是否是MPLS接口，结合了IP定位数据和 路径测量信息。随后验证该方法发现了超过55%的MPLS接口，并且只有10%的non-MPLS错误率  
2. 利用路径长度，接口信息来发现MPLS路径，最终发现80%的路径，只有10%错误率

文章最后得出了结论：1. 过去三年MPLS部署的增长趋势和由于Tier-1提供者广泛使用MPLS使得数据包遇到MPLS隧道的可能性大大增加；2. 隧道可能在10ms级的时间内穿过整个transit provider；3. 检测到流量的分类和使用虽有变化，但是分类多样性依然。

文章在最后利用上述方法应用在pipe mode数据集中（08年上半年数据），并以08年6月份作为参考信息，最终上述方法的结果与实际基本相符。

# Router level

## 别名解析——Discarte

参考文献：

[61] R. Sherwood, A. Bender, and N. Spring, Discarte: a disjunctive Internet cartographer," in ACM SIGCOMM Computer Communication Review, vol. 38, no. 4. ACM, 2008, pp. 303-314.

**文章题目：Discarte工具**

**by 余卓勋**

TCP/IP设计没有显式给出底层路径，不同网络之间无从直接得知其他网络的拓扑，因而需要拓扑发现。接口级别拓扑测量最常用的手段是TR(traceroute)。然而现有的方法使用TR具有易错性(error-prone)：路由器回复的不一致性(路由器不回复，匿名路由器等)，观察点不足，MPLS隧道的存在。

本文提出的DisCarte系统，利用了IP选项中的record route字段，结合traceroute工具，得到RR+TR的路径数据集合。将数据集合预处理，在TR上只差一跳的IP接口互为探测地址对(probe pair)。用probe pair集合作为输入，使用析取逻辑规划(DLP)这一编程范式，把根据工程经验得到的RR的不同实现和导致推断错误的拓扑陷阱转化为DLP的规则(rules)，将TR+RR二者的数据按照路由器对齐。系统最终输出整合了TR，RR数据集并将接口聚合后的路由器级拓扑。

DisCarte的优势是能够准确地对齐TR数据和RR数据，别名解析正确率高。DisCarte和同样使用了RR选项的Passenger Tool比，对齐率96%，别名只错3%。然而，DLP的方法的可扩展性不强，不适用于全网级别的数据集大小。

## 别名解析——Pythia

参考文献：

Pythia: Yet Another Active Probing Technique for Alias Resolution, Pietro Marchetta

**文章题目：Discarte工具**

**by 余卓勋**

Pythia是一种主动探测技术，使用了TS(prespecified timestamp)选项，与之前工作不同之处在于，pythia以一类特殊的路由器(any-interface stamping router)为对象，采用不同的方法构造TS选项，并且探测包使用UDP而不是ICMP。

pythia系统以traceroute得到的候选别名对集合作为输入，输出是聚合了any-interface stamping router的接口的拓扑。pythia采用两个机制，UDP探测，向目的接口发送一个端口不可达的包，应答ICMP包的负载中会包含有原始的UDP探测包，可以从中提取出TS选项。另外一个机制，在构造TS选项时，遵循目的地址IP优先的原则，这样就不会有路径上的中间节点在TS中插入自己的时间戳来扰乱测量。

pythia的流程首先进行初步测试，构造形如A|AAAA，A|AZZZ的TS选项的UDP探测包，发送给需要别名解析的IP集合，根据回复将IP进行分类，排除掉行为有异常的接口。第二步，逐个对剩余IP集中的IP使用形如A|ABCD的探测包进行探测，根据回复推断IP间的关系，如此反复迭代，知道找到所有的别名。

## 路由发现——MRINFO

参考文献：

[63] J.-J. Pansiot, P. M´ erindol, B. Donnet, and O. Bonaventure, “Extracting intra-domain topology from mrinfo probing,” in Passive and Active Measurement. Springer, 2010, pp. 81–90.

**文章题目：路由级拓扑发现——利用mrinfo收集到路由器的邻居信息，进行router-to-AS的映射**

**by 李秉睿**

MRINFO利用IGMP协议，向路由器发送ASK NEIGHBORS信息，接收路由器回复的包含路由器的所有邻居信息在内的NEIGHBORS REPLY信息进行数据收集。

文章使用自主开发的基于mrinfo的mrinfo-rec工具，从单一的路由器地址出发，递归的探测发现到的路由器，将结果作为数据集进行分析，提出了router-to-AS mapping的算法，实现路由器到AS的匹配。

文章从04年开始，利用mrinfo-rec从单一路由器开始探测，直到08年结束。期间平均每日探测10万接口（不包括无法路由到的IP地址，特殊IP地址，隧道，无效接口等），发现1万左右路由器。同时利用每日的BGP TABLE DUMP信息，对探测结果进行IP-to-AS的匹配，排除无法匹配到AS的IP地址（0.5%）和匹配多个AS号的IP地址（2-3%），基本每天识别400-650个AS，4年总共发现超过850个AS，among Tier-1, Transit, and Stub networks, Transit being the most represented。

文章随后提出基于规则池的router-to-AS的匹配算法进行数据处理：

概率规则

1. global election（ elec ）：每一个路由器都匹配其所有接口匹配数量最多的ASN
2. lan rule：局域网接口检测，一旦发现有局域网接口（非网关接口），那么认为该路由器是属于该接口所属的AS

经验规则

1. lb rule：关注路由器的环回接口，当ISP配置换回接口时，通常用ISP自己的地址空间。因此就可以利用DNS域名反响解析识别环回接口，router-to-AS mapping就变成了IP-to-AS mapping
2. N rule：假定域间连接是映射两个AS的互联。因此当一个SAS（即存在接口AS不一致的边界路由器）被其他规则确定其ASN时，利用该规则可以判断与之相连的其他路由器的ASN
3. c2p rule： AS通常将自己的IP地址空间分配给客户AS连接。因此当一个路由器有自己的AS接口，也有客户AS的接口时，该路由器属于客户AS

通过对这些规则进行诸如优先级等处理，最终实现router-to-AS，解决了AS边界路由器判定的问题，结果准确率颇高。

另外，通过mrinfo的信息也可以发现交换机的存在，例如当mrinfo返回路由器某个接口对应多个同网段接口时，表明中间有路由器存在。这个是traceroute等方式也很难发现的。

# AS level

## 利用traceroute拓展AS拓扑图

[45] K. Chen, D. R. Cho\_nes, R. Potharaju, Y. Chen, F. E. Bustamante, D. Pei, and Y. Zhao, "Where the sidewalk ends: Extending the Internet AS graph using traceroutes from P2P users," in Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM, 2009, pp. 217-228.

**文章题目：利用traceroute扩展AS拓扑图**

**by 秦超逸**

文章前半讲述了如何构建AS级拓扑结构。首先，文章通过从全世界的大量P2P用户获取的traceroute数据收集和分析了路径。其次，文章提出了一套彻底的启发法来从traceroute数据中推断出AS级路径，即提出了影响traceroute测量准确性的问题的分析并说明文中的启发式如何解决这些问题。

文章首先使用P2P来监控拓扑结构，通过在监控点上使用Ono来获得traceroute测量数据。数据集来自P2P traceroutes、BGP Feeds和Ground-Truth Data。从P2P traceroutes获取IP路径信息；从BGP Feeds中获取AS连接信息；从Ground-Truth Data获取路由器配置和系统记录。

文章使用一下步骤来构建AS级拓扑结构：

1.预处理IP路径：在进行IP与AS对应之前，要检查每条IP级路径。这些路径中可能出现一些问题，例如：负载均衡、零ttl转发和边界路由地址重写导致的路由循环。移除这些路径。

2.移除IXPs中的IP地址：使用一系列已知的IXP IP前缀，从每个IP路径中移除任何属于一个IXP前缀的跳转。本步骤能够正确地推断出两个在IXP上相互连接的AS间的直接链接。

3.IP到AS的映射：使用Team Cymru提供的IP-to-AS映射工具完成IP级路径到AS级路径的转换。

4.预处理AS路径：使用[1]中的技术对处理不完整的AS路径问题。

5.处理兄弟自治域的问题：从CAIDA下载兄弟AS信息。对于一个兄弟AS对(X,Y)，可能出现AS路径是[…WXYZ…]，而BGP中AS路径是[…WXZ…]或[…WYZ…]，此时将traceroute AS路径修改为[…W{X,Y}Z…]；还可能出现AS路径是[…WYZ…]，为BGP中AS路径是[…WXZ…]，此时用BGP中的AS路径代替traceroute中的。

6.处理剩余问题：文中的算法1处理了traceroute AS路径和BGP路径之间的不一致性问题。算法能够处理循环丢失的、额外的、替代跳转。本文使用了DIST，DIST被定义为两个AS之间的跳转数量，DIST为1代表直接连接。

1. 循环：由于未宣布的IP地址、兄弟AS或转发路径上的路由异常，使得循环在traceroute AS路径上发生时，应当适当地舍弃这些路径。
2. 丢失跳转：当DIST为2，并且BGP上对应的AS路径为[…BXC…]时，对于traceroute上的AS路径，在中间增加跳转X来保证一致性。
3. 替代跳转或额外的跳转：如果DIST为2，并且连接B和C的中间节点是X，但在BGP中找不到相应的AS路径[…BXC…]。解决办法是:如果能够在BGP中找到相应的路由，通过删除或替换traceroute AS路径中间的跳转X来使其与BGP保持一致。

文章描述了使用traceroute获取IP级路径，通过IP地址和AS的映射，实现构建AS级拓扑结构的方法，并设计了一个算法处理了traceroute AS路径和BGP路径之间的不一致性问题，从而提高拓扑结构的正确性。

[1] Z.M. Mao, J. Rexford, J. Wang, and R. Katz, “Towards an Accurate AS-Level Traceroute Tool,” Proc. ACM SIGCOMM, Aug. 2003.

# 测量工具Zmap

Durumeric, Z., Wustrow, E., & Halderman, J. A. (2013). ZMap: Fast Internet-wide Scanning and Its Security Applications. In Proceedings of the 22nd USENIX Security Symposium (pp. 605–619). Retrieved from https://zmap.io/paper.pdf

**论文题目：**

**ZMap: Fast Internet-Wide Scanning and its Security Applications**

**by 郝科委**

互联网范围的网络扫描有许多安全应用，如发现新的漏洞，跟踪网络防御机制等。但是现有的网络扫描工具不能满足大规模网络扫描的速度需求。一种新的模块化开源网络扫描工具zmap在一定程度上解决了这个问题，单台主机，在千兆网卡状态下，45分钟内完成全网络IPv4地址的扫描和探测，速度是常用的网络映射工具nmap的1300多倍。Zmap的性能提高得益于其针对大规模网络应用的特别优化，其中的核心优化总结如下：  
1. 探测优化，使用一个（cyclic multiplicative group）来随机选择目标地址，以保证目标路径不会拥塞（而不是像nmap一样限制传输速度来避免拥塞），跳过TCP/IP协议栈直接生成以太帧，从而以网卡的极限速度发送探测包，大大提高了探测速度。  
2. 无状态，不维护跟踪每个探测的连接状态而是使用类似SYN cookies的方法来区别有效的响应和网络背景流量，对每个扫描的主机，ZMap都依据特定的扫描密钥计算目标地址的MAC（使用UMAC方法），这个MAC值则分散在主动探测模块里的任何可用字段，用于校验。从而大大节省了内存等开销。  
3. 不重传，由于不保存连接状态，所以不需要重传，每个目标都是固定的探测包数目（缺省是1个），而nmap为了保存连接状态需要适应性的重传丢失的数据包，这样就大大提高了速度。