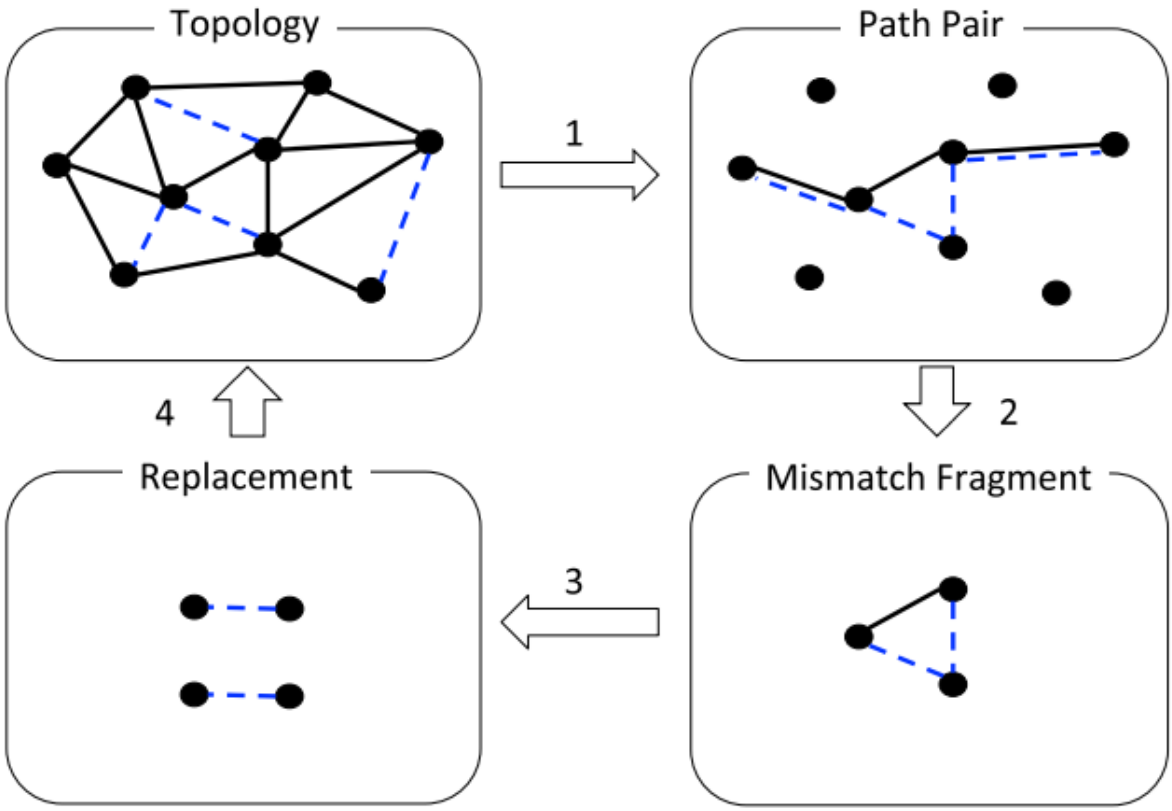
IP2AS映射相关论文总结

A Framework to Quantify the Pitfalls of Using Traceroute in AS-Level Topology Measurement

这篇文章是**JSAC 2010**的一篇文章，是**UCLA和毕军老师团队**合作的论文。

常见的基于BGP路由表将路由器IP地址映射到AS号的方法是很容易出错的。基于这一点，这篇文章的作者团队开发了一种系统化的框架，**量化AS级别拓扑推理中，使用traceroute的潜在误差。**

整个框架由以下四个步骤构成：



1.**拓扑图构建：**收集从同一个AS到同一个目的地的数百万对traceroute路径和BGP路径

2.**定位不匹配片段（核心工作）：**定位不匹配的路径对中的不匹配的片段

3.**定位一对一AS替换：**将每个不匹配的片段转换为多个一对一的AS替换

4.**虚假链路关联：**将拓扑上的每个虚假链路与创建虚假链接的替代品相关联

定位不匹配片段（核心工作）的具体方法：

作者首先将AS路径转换为字符串，基于traceroute推断出的AS路径将尽可能与BGP推断出的AS路径吻合的猜想，将定位不匹配字符串的问题转换为了一个**LCS（longest common subsequence）**问题，并提出了系统性分析的5个步骤：

	(a) substitute	(b) end-extra	(c) unmapped	(d) tie-break	(e) loop	(f) missing tail	(g) omission
BGP path	A B C D	A B	A B C	A B C D	A B	A B C	A B
Traceroute path	A E C D	A B C	A ? C	A C B D	A C A B	A D	A * B
LCS solution	=A -B +E =C =D	=A =B +C =$	=A -B +? =C	=A+C=B-C=D,=A-B=C+B=D	=A +C +A =B	=A -B -C +D =$	=A +* =B
Mismatch fragment	=A -B +E =C	=B +C =$	=A -B +? =C	=A +C =B , =B -C =D	=A +C =A	=A -B +D =$	=A +* =B

Fig. 3. Examples of AS path pairs with their LCS solution and mismatch fragments. '^' and '$' are omitted if not being in mismatch fragments.

- **Special tokens：**两个字符串匹配的结果中出现了特殊字符就必定存在路径不匹配

- **Tie-break**：存在多种替代方式时，使用在BGP路径中=符号出现得尽量比较早的那种
- **Loop**：不匹配的片段是内部循环
- **Missing tail**：AS路径尾部丢失
- **Omission**：丢弃出现的额外路径（出现+的）

最后对不匹配片段出现的原因进行了分析

评价：这篇文章主要提出了一种能够自动定位AS拓扑错误的框架，进而进行了分析。其中分析的部分是整个文章的大头，主要对AS拓扑的错误出现原因进行了具体的分析，让我感觉这篇文章读着有点报告的感觉。

bdrmap: Inference of Borders Between IP Networks

这篇文章是**IMC 2016**的一篇文章，主要作者是**UCSD CAIDA**团队。

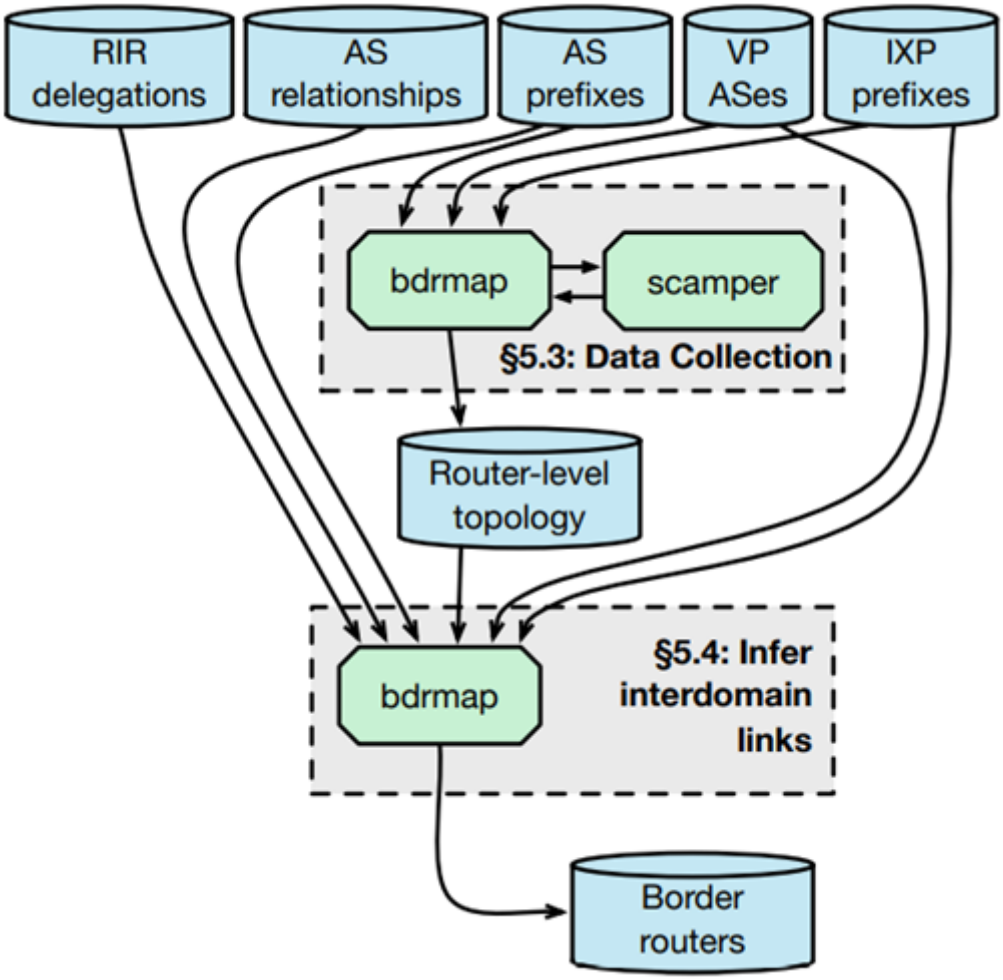
准确识别网络边界对于当前学术、工业界的很多问题都十分重要，但是这样的互联网路由级拓扑发现和推理是十分容易出错，主要由于以下几个原因：

- TCP/IP 体系结构没有提供检测域间边界的机制
- 未能准确解析同一路由器的此类别名将导致网络之间链路数量的夸大推断
- 运营商地址分配和路由实施实践限制了准确性
- Traceroute 反复采样部分处于中心位置的链接但忽略更远的链接
- 运营商担心向竞争对手透露他们的拓扑

因此，这篇文章开发了一种方法来准确判断网络边界，具体的贡献如下所示：

- 引入一种可扩展的方法来准确**推断给定网络的边界**，以及附加在每个边界上的其他网络
- 开发一个有效的系统，允许在资源有限的设备上部署他们的方法
- 使用来自四家网络运营商的真实情况以及 IXP 地址使用数据库来验证其算法的正确性
- 通过分析大型接入 ISP 的拓扑结构以了解现代互连协议，展示算法的效用
- 公开发布源码实现

分析框架如下：



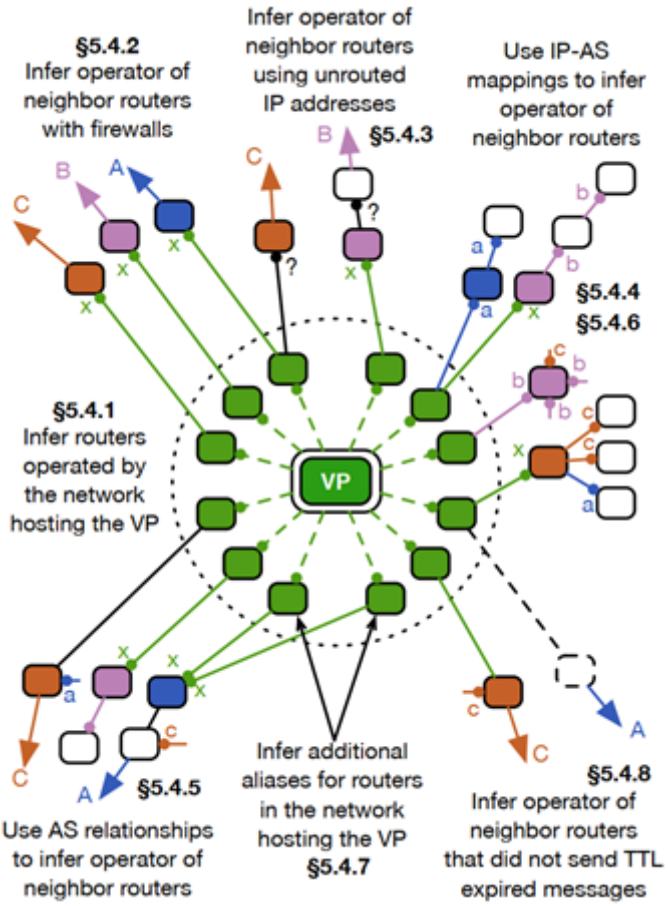
主要有三步，其中第三步是核心：

- 收集用于通知数据收集和分析的路由和寻址数据
- 部署一个高效的 traceroute，以跟踪从每个 VP 到全局 BGP 路由系统中观察到的每个路由前缀的路径，应用别名解析技术来推断用于域间互连的路由器和点对点链路，构建整个网络初步的拓扑结构
- 使用收集到的数据来组合约束，使用启发式算法以推断路由器所有权

第二步的主要步骤：

- 生成要探测的地址块列表
- 收集traceroute
- 解析路由地址的IP别名
- 推断点对点链接
- 限制虚假别名
- 构建路由器级别拓扑图

第三步的主要方法：



最后对推理结果进行了验证，并对比了分别使用traceroute和BGP的推理结果

MAP-IT: Multipass Accurate Passive Inferences from Traceroute

这篇文章是**IMC 2016**的文章，两个作者都是**宾夕法尼亚大学**的，通讯作者是Jonathan M. Smith教授，一作是他的学生，现在去**UCSD CAIDA**了。

这篇文章主要着眼于如何可靠的识别AS之间的链路的地址。

准确识别AS之间的链路地址主要有两个困难：

- 在两个AS之间的链路可能被分配来自同一个地址空间的IP地址
- Traceroute artifacts可能导致错误

这篇文章的主要贡献如下：

- 描述了一种新颖、稳健且高度精确的多通道算法MAP-IT，用于从 traceroute 推断用于 AS 间链路的接口（核心）
- 使用来自二级区域网络提供商 Internet2 的实际情况验证算法，达到 100% 的精确度和 96.9% 的召回率
- 使用从 ISP 级别 3 和 TeliaSonera 中的接口的 DNS 主机名得出的近似基本事实来确认这些结果
- 将 MAP-IT 与用于识别 AS 间链接接口的现有方法进行比较，展示出更高的准确性

MAP-IT算法更多地关注对算法效果的提升而不是使用更多的监测设施提供效果上的提升

- Discard and Sanitize Traces
    - 路由器的配置不当、负载均衡以及瞬时路由更改可能会导致traceroute伪影
    - 2步对污染进行缓解:
        - 删除由转发 TTL=1 数据包的有问题的路由器引起的虚假邻接
        - 确定在跟踪期间是否发生负载平衡或瞬时路由更改

- Determine Interface Other sides

Algorithm 2 Direct Interfaces

Require: f ▷ $0 \leq f \leq 1$

1: **for** each IH, h, w/o a direct inference **do**
2: Find AS_N which appears more than any other AS in h's $\mathbb{N}$ using previous IP2AS
3: **if** COUNT$(AS_N) \geq$ COUNT$(neighbors) \times f$ **then**
4: **if** previous IP2AS$(h) \neq AS_N$ **then**
5: Mark a direct inference for h
6: Update current IP2AS$(h) \leftarrow AS_N$

  - 点对点链接可以从 /30 或 /31 前缀分配地址
  - 使用启发式推断每个接口的另一侧
  - 提取任何Traceroute中看到的所有地址，包括丢弃的Traceroute
  - /30 前缀中的所有非主机地址都被分配到 /31 前缀的另一侧
  - 对于剩余的有效主机地址，检查数据集中是否出现了不同的地址，该地址将是其 /30 前缀中的保留地址。 如果是这样，将其分配给 /31 前缀的另一侧，否则假设它来自 /30 前缀。
- Extract Neighbors Sets
  - 使用清理后的Traceroute，创建第 3 节中描述的 NF 和 NB。接口的 Ns 包括所有跟踪中恰好在它之前 (NB) 或之后 (NF) 一跳看到的所有地址，不包括空跳和私有/共享地址 . 不在 Ns 中包含私有/共享地址，因为它们不是全局可路由的或唯一的，并且不应出现在跟踪路由中。
  - 在数据集中与至少一个其他地址相邻的 4,752,201 个接口地址中，449,602 个具有多个地址的 NF，1,139,087 个具有多个地址的 NB。 对于在第 4.4.1 节中对特定接口进行的直接推断，其 NF 或 NB 必须至少包含 2 个地址。
- Adding Inferences
  - 一个接口被确定用于跨域链接分4个步骤：

    •使用 Ns 和当前的 IP2AS 映射对 IH 进行直接推断（第 4.4.1 节）；

    •更新每个直接推理另一侧的映射（第 4.4.2 节）；

    •解决对同一接口向前和向后进行推断的矛盾（第 4.4.3 节）；

    •解决对相邻 IH 进行的逆推理6，保留一个推理并丢弃另一个（第 4.4.4 节）。
- Remove Inferences

Algorithm 3 Removing Interfaces

1: **repeat**
2: **for** each direct inference on h to AS_N **do**
3: **if** the inference would no longer be made **then**
4: Make the inference indirect
5: Discard indirect inferences w/o direct inference
6: Remove updates for discarded inferences
7: **until** no inferences were discarded

  - 当访问每个具有直接推理的 IH 时，该算法会根据当前的 IP2AS 映射检查连接的 AS 是否仍占其 N 的一半以上。
  - 如果不是，那么最初将推理从直接推理更改为间接推理，但保留其 IP2AS 映射。
  - 每次通过 IH 后，所有没有关联直接推理的间接推理以及它们的 IP2AS 更新都将被丢弃
- Overall Convergence

Algorithm 4 Low Visibility and NAT Heuristic

1: **for** each IH, h_f, w/ a single neighbor n_b **do**
2: **if** no inference for h_b or n_b & $AS_H \neq AS_N$ **then**
3: **if** AS_N is a stub AS **then**
4: Mark a direct inference for h_f
5: Mark an indirect inference for h'_b
6: Update mappings for h_f and h'_b to AS_N

    - 由于存在不确定推理，整个算法重复添加和删除步骤，可能永远不会达到不能添加推理和不能删除推理的地步。
    - 相反，它会收敛到不断添加和删除相同推论的点。
    - 因此，在移除步骤结束时寻找一个重复的状态作为停止标准，这表明不能做出更可靠的推断。
    - 在实验中，这发生在主 while 循环的 3 次迭代之后
- Traceroute Artifacts
    - 如果邻居用于 AS 间链路，则应该可以进行反向推断，因为提供商通常会从许多入口点为其客户接收数据包，这可能会暴露边界路由器上的多个接口
    - 如果隐藏 AS 中只有一跳，则第一个 AS 和末节 AS 之间可能存在隐藏 AS。 这会导致断开连接的 Ases 之间的推断无效
    - 第三方地址不会导致此步骤的错误推断。 存根 AS 返回的第三方地址将是其提供者之一
- Stub AS Heuristic
    - Traceroute伪影：
        - 错误
        - 输出接口
        - 临时路由更改
        - 每包负载均衡
    - 可能导致错误或阻止进行推断。 即使是旨在避免大多数类型的负载平衡的巴黎跟踪路由，也不能免受工件的影响

Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale

这篇论文是**IMC 2018**的，是上面**MAP-IT的宾大的组和UCSD的CAIDA组合作**做的，基于bdrmap的工作和MAP-IT的工作展开的，应该是IMC 2016上两个组的social的。

bdrmap

使用来自特定网络的目标跟踪路由、别名解析探测技术和 AS 关系推断，来推断该特定网络的边界以及连接在每个边界上的其他网络

MAP-IT

解决了在从许多不同网络启动的大量跟踪路由存档集合中推断所有 AS 级网络边界的艰巨挑战

Contributions如下：

- 开发并实现了 bdrmapIT，它使用复杂的 bdrmap 启发式算法作为 MAP-IT 边界定位算法的附加输入。 在第 2 节中，讨论了如何利用 bdrmap 和 MAP-IT 的优势，第 3 节概述了合成方法，第 4、第 5 和第 6 节详细介绍了算法。
- 展示了 bdrmapIT 优于之前任一方法的准确性和覆盖范围。 尽管没有在任何验证网络中使用 traceroute VP，还是根据来自第 1 层、大型访问和两个 R&E 网络的真实情况验证了 bdrmapIT，实现了 91.8%-98.8% 的准确率。 还证明了 bdrmapIT 的准确性与有利位置的数量无关：当将 VP 的数量从 80 减少到 20 时，性能是等效的。

- 发布实现和源代码以提高可重复性，以便其他人可以使用工具进行他们自己的分析。 将 bdrmapIT 纳入 CAIDA 的 ITDK生成过程。