

PoPmap: Agrupamento de Roteadores de rede em Pontos-de-Presença

Geraldo Augusto Silva Junior¹, Ítalo Cunha¹

¹Departamento de Ciência da Computação (DCC)
Universidade Federal de Minas Gerais (UFMG)

{geraldo, cunha}@dcc.ufmg.br

Abstract. *Research about network topology can benefit from data in different granularities. For example, we can group interfaces of a router, at a point of presence, at an Internet exchange point, or at a facility. In this paper, we present PoPmap, a tool for topological clustering of network interfaces. We group interfaces using the k-Means clustering algorithm. To solve scalability problems, we use the PCA component reduction algorithm. We use measurements from PlanetLab to evaluate the proposed method. We observed that 41% of clusters with more than one interface have 80% of the interfaces inside one autonomous system and that 71% have average standard deviation of RTT less than 10ms.*

Resumo. *Estudos sobre a topologia da Internet podem se beneficiar de dados em diferentes granularidades. Por exemplo, podemos agrupar interfaces que estão em um roteador, ponto de presença, ponto de troca de tráfego ou centro de hospedagem. Nesse trabalho apresentamos PoPmap, uma ferramenta para agrupamento topológico de interfaces de rede. Fazemos o agrupamento usando o algoritmo k-Means. Para contornar problemas de escalabilidade, usamos o algoritmo de redução de componentes PCA. Usamos computadores do PlanetLab para coletar medições para avaliar o método proposto. Observamos que 41% dos grupos com mais de uma interface possuem 80% das interfaces dentro de apenas um sistema autônomo e que 71% possuem desvio padrão médio de RTT menor que 10ms.*

1. Introdução

Diversos sistemas de monitoramento e estudos da Internet podem utilizar dados sobre sua topologia em diferentes granularidades. Por exemplo, um sistema de monitoramento e previsão de desempenho da Internet como o iPlane [Madhyastha et al. 2006] pode usar uma topologia mais compacta para fazer menos medições. Uma das formas de conseguir isso é agrupar interfaces de rede, identificadas por seus endereços IP, relacionadas. Podemos agrupar interfaces de rede que pertencem a um mesmo roteador ou dispositivo de rede para construir uma visão da topologia dos roteadores da rede [Sherry et al. 2010, Marchetta et al. 2013]. Podemos agrupar interfaces de rede e roteadores topologicamente relacionados em pontos de presença (PoP) para construir uma visão da topologia da infraestrutura da rede [Shavitt and Zilberman 2010, Madhyastha et al. 2006]. Por fim, podemos agrupar interfaces de rede, roteadores e PoPs nos sistemas autônomos aos quais pertencem para construir uma visão no nível das redes que integram a Internet [Oliveira et al. 2010, Dhamdhere and Dovrolis 2008].

Nesse trabalho apresentamos PoPmap,¹ uma ferramenta para agrupamento topológico de interfaces de rede em pontos de presença. Para cada interface de rede, medimos a distância, em número de saltos (*hops*), a partir de vários pontos de medições para construir um *vetor de localização*, onde cada elemento é a distância, em número de saltos, entre a interface e um ponto de medição. O número de saltos entre uma interface de rede e um ponto de medição é função da rota utilizada entre eles. O PoPmap identifica quais interfaces utilizam rotas similares e estão topologicamente próximas comparando os vetores de localização. Este método foi originalmente proposto pelo iPlane [Madhyastha et al. 2006] e supõe que duas interfaces com vetores de localização similares utilizam as mesmas rotas e estão topologicamente próximas.

Usamos o algoritmo de agrupamento k-Means para agrupar as interfaces topologicamente próximas. Depois de fazermos o agrupamento, podemos criar uma estrutura da Internet em que a rota entre dois pontos é representada por uma sequência de grupos de interfaces, capturando o conceito de pontos de presença (PoP) o que aumenta a granularidade da topologia usada.

A quantidade de interfaces na Internet é muito grande. Além disso, ao aumentar a quantidade de pontos de medição, também aumentamos a quantidade de dimensões consideradas pelo algoritmo de agrupamento. Isso pode tornar o tempo de execução e o uso de memória na fase de agrupamento excessivamente grandes. Nem sempre, a adição de um novo ponto de medição traz grande quantidade de informação sobre a localização das interfaces, útil para o agrupamento, já que o grau de correlação entre a informação gerada por esse novo ponto de medição e algum anteriormente escolhido pode ser alto.

Para reduzir a quantidade de pontos de medição com alto grau de correlação utilizamos o algoritmo de redução de componentes PCA. Assim reduzimos a quantidade de dimensões usadas na execução do algoritmo k-Means viabilizando sua execução em escala global. Dessa forma, mantemos a maior parte da informação útil e reduzimos os requisitos de CPU e memória para a execução do algoritmo de agrupamento. Com nossas otimizações, o PoPmap consegue agrupar um conjunto de mais de 800 mil interfaces em 60 mil PoPs em menos de 72 horas.

Para aplicar a nossa metodologia, identificamos um conjunto de interfaces do núcleo da rede com a intenção de agrupar interfaces em PoPs. Escolhemos interfaces que pertencem a roteadores que encaminham pacotes IPs na Internet. Para isso, montamos um grupo de interfaces de rede observadas em medições de traceroute obtidos do Cousteau, uma API que permite coletar medições públicas do RIPE Atlas em tempo real. Para construir os vetores de localização usados no algoritmo de agrupamento, fizemos medições de ping a partir dos computadores do PlanetLab.

Para avaliar a qualidade dos grupos formados pelo PoPmap, selecionamos os grupos formados com mais de uma interface e observamos a composição de sistemas autônomos (ASes) e a variabilidade da latência entre interfaces no mesmo grupo. Esperamos que grupos concentrem grande parte de suas interfaces dentro de poucos ASes e tenham latência similar para diferentes pontos de medição. De acordo com as expectativas, observamos que 41% dos grupos com mais de uma interface possuem 80% das

¹O PoPmap e os dados utilizados em sua avaliação estão disponíveis para a comunidade científica no endereço <http://www.dcc.ufmg.br/~geraldopopMap.html>

interfaces dentro de apenas um sistema autônomo e que 71% possuem desvio padrão médio de RTT menor que 10ms.

2. PoPmap

Nessa seção apresentamos os dados, algoritmos e métodos utilizados pelo PoPmap para agrupar interfaces em PoPs. Em particular, discutimos como medir o vetor de localização de cada interface (§ 2.1), como reduzimos o conjunto de dados para melhorar escalabilidade (§ 2.3), como calculamos a distância entre duas interfaces (§ 2.2), e como configuramos o k-means (§ 2.4).

2.1. Localizando uma interface na Internet

Quando uma origem s envia um pacote ICMP echo request (ping) para um destino d na Internet, d responde com um pacote ICMP echo reply (pong). Como todo pacote IP, a resposta de d possui um campo de tempo de vida (*time to live*, TTL), que é inicializado pelo sistema operacional. O campo TTL de um pacote IP é decrementado por cada roteador por onde passa.² A função do TTL é evitar que um pacote IP seja roteado indefinidamente em rotas com ciclo.

Quando o pacote ICMP echo reply chega na origem s , o valor do TTL terá sido decrementado um número de vezes igual à quantidade de roteadores pelos quais o pacote passou. Se conhecermos o valor com o qual o TTL foi inicializado por d , podemos calcular a quantidade de roteadores na rota de d até s . A partir de agora, chamaremos essa distância em número de saltos.

Em geral, sistemas operacionais inicializam o TTL em um valor fixo conhecido [Sherwood et al. 2008]. Na maioria dos casos ele é inicializado com uma potência de 2.³ Seja ω o valor do campo TTL quando um pacote IP chega a um computador, podemos estimar o número de roteadores que atravessou (saltos) como $H = \max(32, 2^{\lceil \log \omega \rceil}) - \omega$.

No PoPmap fazemos medições de distância em número de saltos de um conjunto de computadores \mathcal{C} para um conjunto de interfaces \mathcal{I} que desejamos agrupar. O PoPmap faz uma medição ping de cada computador $c \in \mathcal{C}$ para cada interface $i \in \mathcal{I}$ para medir o número de saltos na rota de i para c , denotado δ_{ic} . Para cada interface, construímos um vetor de localização com as distâncias medidas por cada computador, i.e., $L_i = [\delta_{i1}, \delta_{i2}, \dots, \delta_{i|\mathcal{C}|}]$.

2.2. Definição de distância entre duas interfaces

Esperamos que duas interfaces i e j que estejam próximas topologicamente tenham vetores de localização L_i e L_j similares, com elementos de L_i próximas das de L_j . Para medir o quanto uma interface está próxima usamos a distância euclidiana (norma L2). Dessa forma, quanto menor for a distância euclidiana entre os vetores L_i e L_j , mais próximas consideramos as interface i e j . Podemos usar essa definição de distância para aplicar um algoritmo de agrupamento em que cada elemento do vetor de localização será uma dimensão, como será detalhado a seguir.

²Alguns roteadores podem não decrementar o valor do TTL [Sherwood et al. 2008]. Porém, como esses roteadores são invisíveis a medições de traceroute em geral não são identificados para agrupamento.

³Como várias rotas na Internet passam por mais de 16 roteadores, valores comuns para inicialização do TTL incluem 32, 64 e 128.

2.3. Redução do número de coordenadas do vetor de localização

Seja φ_i a variável aleatória que recebe valores das medições de número de saltos dos computadores em \mathcal{C} para a interface i . Podemos medir o grau de correlação entre um par de variáveis φ_i e φ_j pelo cálculo da covariância $\text{cov}(\varphi_i, \varphi_j)$. Quanto maior for o módulo dessa covariância, maior será o grau de correlação entre as variáveis φ_i e φ_j , o que significa que parte da “informação” presente na variável φ_i está na variável φ_j , e vice-versa. Isso pode ocorrer, por exemplo, por quê interfaces i e j estão topologicamente próximas. Quando consideramos todas as $|\mathcal{I}|$ variáveis aleatórias, poderá haver correlação entre várias delas, indicando redundância na informação coletada pelos computadores usados como pontos de medição.

Para acelerar o processo de agrupamento, gostaríamos de reduzir a dimensão dos vetores de localização. O algoritmo de redução de dimensionalidade PCA faz isso minimizando a perda de informação. O PCA transforma o conjunto original de variáveis, que podem ser correlacionadas, em novo conjunto de variáveis ortogonais que não são correlacionadas. Esse novo conjunto de variáveis é chamado de componentes principais. A transformação é definida de forma que a primeira componente principal tenha a maior variância possível. As demais componentes são definidas para terem a maior variância possível sob a condição de serem ortogonais às componentes já definidas.

Aplicamos o PCA e identificar as primeiras componentes principais, que capturam a maior parte da informação contida nos dados, e reduzir a dimensionalidade dos vetores de localização. A quantidade de dimensões removidas depende do grau de correlação entre as variáveis originais e a quantidade de variância que será mantida. Neste trabalho somos conservadores e mantemos 95% da variância dos dados, o que removeu 60% das dimensões. Outros trabalhos podem manter mais ou menos variância para controlar os recursos necessários para a execução do k -means.

2.4. Aplicação do algoritmo de agrupamento k -means

Depois da redução da quantidade de dimensões, aplicamos o algoritmo k -means para agrupar as interfaces que estão topologicamente próximas. O algoritmo k -means calcula um centroide para cada grupo de interfaces formado. As coordenadas dos centroides são calculadas como a média das coordenadas das interfaces em um grupo. A *inércia* é uma medida da qualidade de um agrupamento e é calculada somando-se as distâncias euclidianas entre cada interface e o centroide mais próximo.

Nós consideramos a utilização de diferentes algoritmos de agrupamento, como o x -means. Porém, optamos por utilizar o k -means, de menor complexidade, por que as implementações apresentam desempenho e consumo de memória adequados para execução em nossas bases de dados.

Executamos o k -means repetidas vezes para diferentes valores de k e plotamos um gráfico da inércia em função do valor k . Se escolhermos k muito pequeno teremos uma inércia muito grande e poucos grupos, com muitas interfaces, não representativos de PoPs na Internet. Se fizermos k tender ao número de elementos, o valor da inércia tenderá a zero e teremos muitos grupos, com poucas interfaces, que podem representar apenas parte de um PoP na Internet. Neste trabalho, propomos escolher um valor de k tal que a variação da inércia em função de k seja pequena. Escolhido o valor de k , basta usar os grupos formados para descobrir as interfaces IP que estão próximas na rede.

No PoPmap, usamos as implementações dos algoritmos k -means e PCA da biblioteca scikit-learn e utilizamos o algoritmo de inicialização de componentes k -means++.

3. Coleta e preparação dos dados

O primeiro passo para aplicar a PoPmap é obter um conjunto de interfaces IP \mathcal{I} para agrupamento. Nesse trabalho, consideramos interfaces observadas em medições de traceroute entre vários pontos de medição e vários destinos. Dessa forma, selecionamos as interfaces que pertencem a roteadores na Internet.

Usamos as medições de traceroute do RIPE Atlas disponibilizadas em tempo real pelo Cousteau para formar o nosso conjunto de entrada de interfaces. Reunimos 863.860 interfaces IPv4 em um período de nove dias iniciando 1º de março de 2018. Mapeamos as interfaces para sistemas autônomos na Internet usando a base de dados do Team Cymru e verificamos que as interfaces cobrem 45.031 ASes na Internet.

Para coletar as medições de distância em número de saltos, utilizamos 63 nós PlanetLab para realizar medições de ping para cada interface $i \in \mathcal{I}$. Em nossos dados, 33.4% das interfaces deixam de responder a pelo menos um ping enviado pelos nós PlanetLab. Neste caso, o vetor de localização da interface terá coordenadas com valores indeterminados. Se estiverem faltando poucas coordenadas, o vetor continuará representando bem a localização da interface. Neste trabalho, calculamos as coordenada indeterminadas como a mediana dos valores conhecidos da coordenada desde que a quantidade de coordenadas indeterminadas seja menor que 10% da quantidade $|\mathcal{C}|$ de coordenadas.⁴

4. Avaliação dos PoPs

Nessa seção, avaliamos os mecanismos utilizados pelo PoPmap e caracterizamos os PoPs resultantes do agrupamento. Agrupamos as interfaces em $k = 60.000$ PoPs. Realizamos um estudo da variabilidade da inércia do agrupamento em função de k e observamos que o ponto de inflexão da curva é aproximadamente em $k = 20.000$ e escolhemos $k = 60.000$ conservadoramente para minimizar o impacto da variabilidade de k na inércia. Não mostramos esta análise por restrições de espaço.

4.1. Avaliação da estabilidade da distância em número de saltos

Agrupamos interfaces de acordo com seus vetores de localização, compostos pelos números de saltos entre as interfaces e os pontos de medição. Usando o PlanetLab, os pings necessários para medição do número de saltos duram cerca de 2 dias para as 863.860 interfaces, já que limitamos a taxa de sondagem devido à limitações da plataforma. Nesse período, rotas podem mudar fazendo com que os valores de TTL mudem e fiquem inconsistentes.

Ao fazer as medições consideramos que os valores de TTL ficarão estáveis nesse período de 2 dias para que seja possível capturar localizações consistentes para todas as interfaces. Se essa premissa não se verificasse e os valores variassem muito, não faria sentido fazer uma medição que dure dois dias.

⁴Não consideramos interfaces cujos vetores de localização têm mais de 10% das coordenadas indeterminadas no processo de agrupamento. Estas interfaces podem ser agrupadas a outras interfaces depois que os grupos forem formados.

Para verificar que os valores de TTL permanecem relativamente estáveis em um período de 2 dias, selecionamos um subconjunto de 100.000 interfaces do conjunto original e coletamos medições periódicas a partir de 63 computadores do PlanetLab. Cada computador fez 16 medições de ping para cada uma das interfaces amostradas em intervalos de 3 horas durante dois dias. Para cada interface i , cada computador c produziu um vetor com 16 medições de número de saltos. Seja σ_{ic} o desvio padrão do vetor de medições entre i e c . A figura 1 indica mostra a fração acumulada de pares (i, c) em função do desvio padrão σ_{ic} . Observamos que 94.4% dos pares (i, c) possuem desvio padrão menor do que 1, indicando que as distâncias, em números de saltos, permanecem relativamente estáveis num período de 2 dias.

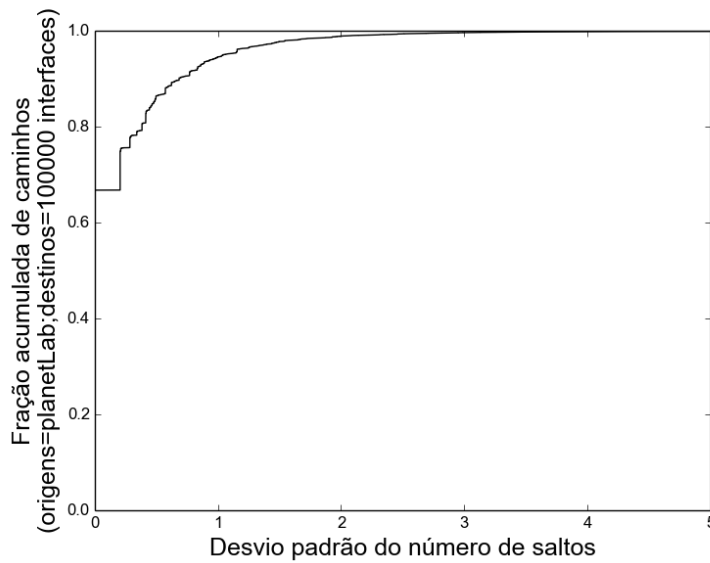


Figura 1. Distribuição do desvio padrão do número de saltos medidos entre cada par (interface i , computador c)

4.2. Verificação da qualidade dos grupos com o mapeamento de IP para AS

Esperamos que PoPs possuam grande parte de suas interfaces dentro de um mesmo sistema autônomo (AS). Para verificar que isso acontece com os grupos formados pelo PoPmap, fizemos um mapeamento de IP para AS usando a base do Team Cymru e calculamos a quantidade de ASes em cada grupo. A figura 2 mostra a distribuição da porcentagem de interfaces no AS com maior quantidade de interfaces em cada grupo (AS dominante). Observamos que 47.2% dos grupos formados possuem possuem porcentagem de interfaces no AS dominante superior a 80%.

4.3. Verificação da qualidade dos grupos por latência

Espera-se que a latência para interfaces em um PoP seja aproximadamente a mesma a partir de vários pontos de medição, já que interfaces em um PoP estão geograficamente e topologicamente próximas. Para verificar isso, medimos a latência para todas as interfaces em um grupo a partir de todos os pontos de medição. Para cada par (ponto de medição, grupo), calculamos o desvio padrão das medições de latência correspondente. A figura 3 mostra a distribuição acumulada para pares (ponto de medição, grupo), do desvio

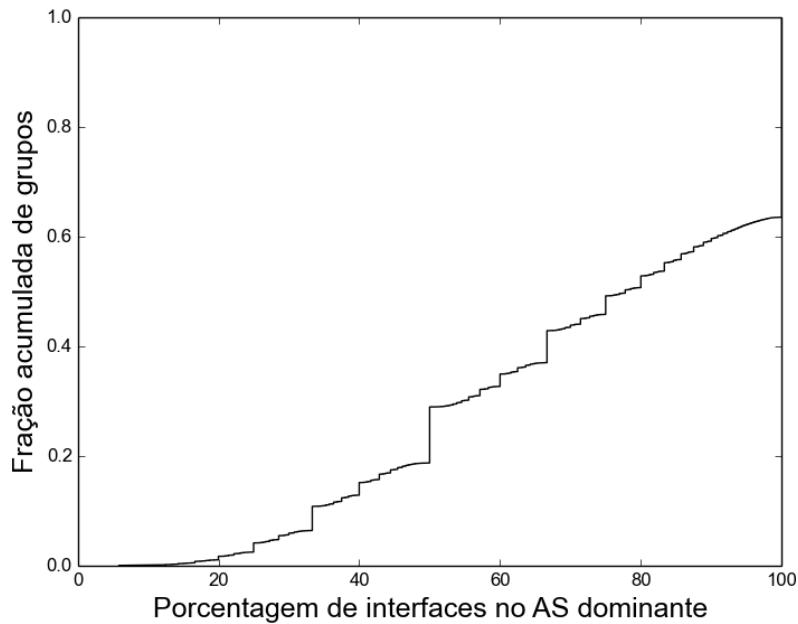


Figura 2. Distribuição da porcentagem de interfaces no AS dominante. O AS dominante é aquele que possui mais interfaces dentro do grupo.

padrão das medições de latência. A curva sólida corresponde a grupos formados pelo PoPmap e a linha pontilhada corresponde a grupos formados aleatoriamente, mas seguindo a mesma distribuição de tamanho dos grupos formados pelo PoPmap. Observamos que a variação da latência entre interfaces de grupos formados pelo PoPmap é pequena.

5. Trabalhos Relacionados

A principal inspiração para esse trabalho foi o iPlane [Madhyastha et al. 2006] e o Sibyl [Cunha et al. 2016]. Esses sistemas apresentam técnicas para prever rotas na Internet. O iPlane propôs uma solução diferente para o problema de agrupamento de interfaces em PoPs. O iPlane, porém, foi descontinuado e sua técnica de agrupamento é mais difícil de executar na prática pois requer informações mais difíceis de coletar (como geolocalização de interfaces de rede via nome de domínio DNS). [Shavitt and Zilberman 2010] apresentam uma técnica para agrupar interfaces em PoPs através de medições de traceroute. Nossa solução se diferencia por usar apenas medições ping, com custo menor, e por discutir escalabilidade do processo de agrupamento via redução de dimensionalidade dos dados.

6. Conclusão

Nesse trabalho apresentamos PoPmap, uma ferramenta para agrupamento de interfaces topologicamente próximas usando a distância, em número de saltos, entre cada interface e pontos de medição na Internet. O PoPmap utiliza o PCA para reduzir a dimensionalidade dos dados, escalando o k -means para centenas de milhares de interfaces e redes como a Internet. A avaliação do PoPmap demonstrou que interfaces em um PoP concentram-se em poucos ASes e que interfaces em um PoP têm latência similar para pontos de medição geograficamente distribuídos.

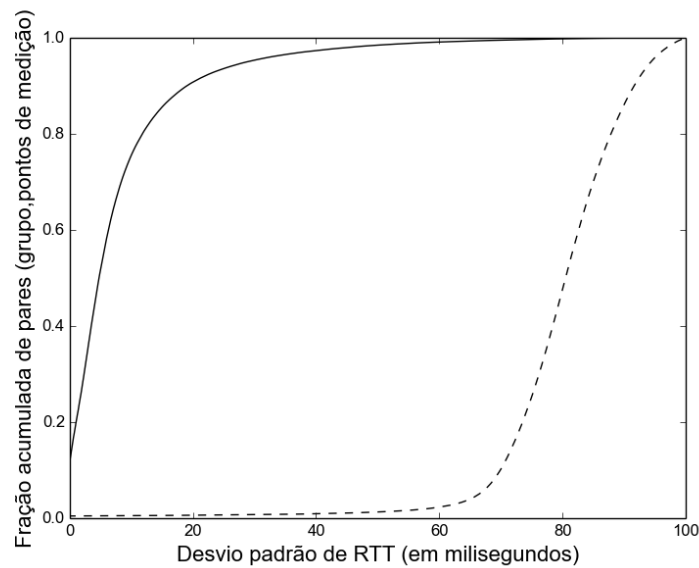


Figura 3. Distribuição do desvio padrão dos valores de RTT obtidos para cada par (grupo, ponto de medição). A curva sólida se refere aos grupos feitos pelo PoPmap e a curva tracejada se refere a grupos gerados aleatoriamente.

Referências

- Cunha, Í., Marchetta, P., Calder, M., Chiu, Y.-C., Schlinker, B., Machado, B. V., Pescapè, A., Giotsas, V., Madhyastha, H. V., and Katz-Bassett, E. (2016). Sibyl: A practical internet route oracle. In *USENIX NSDI*.
- Dhamdhere, A. and Dovrolis, C. (2008). Ten years in the evolution of the internet ecosystem. In *ACM IMC*.
- Madhyastha, H. V., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A., and Venkataramani, A. (2006). iPlane: An information plane for distributed services. In *USENIX OSDI*.
- Marchetta, P., Persico, V., and Pescapè, A. (2013). Pythia: yet another active probing technique for alias resolution. In *ACM CoNEXT*.
- Oliveira, R., Pei, D., Willinger, W., Zhang, B., and Zhang, L. (2010). The (in) completeness of the observed internet as-level structure. *IEEE/ACM Transactions on Networking (ToN)*, 18(1):109–122.
- Shavitt, Y. and Zilberman, N. (2010). A structural approach for pop geo-location. In *IEEE INFOCOM Workshops*.
- Sherry, J., Katz-Bassett, E., Pimenova, M., Madhyastha, H. V., Anderson, T., and Krishnamurthy, A. (2010). Resolving ip aliases with prespecified timestamps. In *ACM IMC*.
- Sherwood, R., Bender, A., and Spring, N. (2008). DisCarte: a Disjunctive Internet Cartographer. In *ACM SIGCOMM*.