AliasCluster : A Lightweight Approach to Interface Disambiguation

Larissa Spinelli
Boston University
Boston, MA
lspinell@bu.edu

Mark Crovella
Boston University
Boston, MA
crovella@bu.edu

Brian Eriksson
Technicolor Research
Palo Alto, CA
brian.c.eriksson@gmail.com

***Abstract*—Internet topologies discovered by standard traceroute-based probing schemes are limited by many factors. One of the main factors is the ambiguity of the returned interfaces, where multiple unique interface IP addresses belong to the same physical router. The unknown assignment of interface IPs to physical routers can result in grossly inflated estimated topologies compared with the true underlying physical infrastructure of the network. The ability to determine which interfaces belong to which router would aid in the ability to accurately reconstruct the underlying topology of the Internet. In this paper, we present ALIASCLUSTER, a lightweight learning-based methodology that disambiguates router aliases using only observed traceroute measurements and requires no additional load on the network. Compared with existing techniques, we find that ALIASCLUSTER can resolve the same number of true router alias pairs with 50% fewer false alarms.**

I. INTRODUCTION

Identifying the router-level structure of the Internet helps many applications, including provisioning, network security, and traffic analysis. Router-level Internet topology discovery has been the focus of a number of prior studies (*e.g.*, [1], [2], [3], [4], to name only a few). Unfortunately, the main tool available for this purpose is traceroute, and it is hampered by a central problem: resolving interfaces (aliases) to routers.

Measurements with traceroute are widely used in Internet topology studies. These measurements allow for the observation of Internet paths and adjacencies with respect to router interfaces. A conflict arises between the observation of router interface IP paths and the goal of physical Internet topology recovery – each physical Internet router contains multiple interface IP addresses. As a result, estimated topologies from current measurement campaigns can be grossly inflated with respect to the physical topology. In Figure 1, we demonstrate a simple example of multiple interfaces on routers can result in an incorrect estimated topology from traceroute measurements.

The identification of which observed interface IP addresses belong to the same physical router is commonly referred to as "Interface Disambiguation". Prior techniques have relied on exploiting routers that support Record Route [5], observed IP-ID values [6], MRinfo requests on multicast enabled routers [7], or exhaustive pairwise probing of observed router interfaces [8]. Each of these methods requires the collection of some additional information along with the traceroute measurements, using an expensive or time-consuming probing strategy that is often impractical for large-scale global Internet studies.

In this paper we present a novel method for interface disambiguation that does not rely on cooperative routers or additional heavyweight probes. Our goal is to infer the interfaces associated with each router *directly* from traceroute measurements without additional inputs. To do that, we introduce the ALIASCLUSTER methodology, a lightweight inference technique to disambiguate observed router IP addresses. This technique consists of two components. The first component identifies three unique features normally present in traceroute and uses an efficient Naive Bayesian approach to fuse information into a single estimate of pairwise alias likelihood. The second component uses these estimates to infer router clusters via a hierarchical clustering procedure that incorporates both estimated alias likelihood and confidence based on the quantity of observed measurements with respect to the interfaces.

Using the wealth of large-scale Internet measurement studies that are already available, we assess the accuracy of the ALIASCLUSTER methodology. From the CAIDA ARK study [1] and ground truth router alias classifications from the MERLIN project [7], we find that our technique can detect 25% of the true router aliases with false alarm rates on the order of 10^{-5}. When compared with competing techniques, ALIASCLUSTER declares 50% fewer false alarms.

The paper is structured as follows. Work related to the interface disambiguation problem is reviewed in Section II. The data sets we used for validation and demonstration are described in Section III. The ALIASCLUSTER methodology is detailed in Section IV. Validation results are shown in Section V, and finally, the conclusions and future work are discussed in Section VI.

II. RELATED WORK

The interface disambiguation problem has been extensively studied in prior work. Initial research focused on determining aliases via specialized pairwise probes. For example, iPlane [4], Ally [3], the RadarGun framework [9] and more recently in the MIDAR framework [6], all use pairwise UDP and ICMP probes to compare if the two interface IP addresses are on same source address, have similar IP-ID and similar

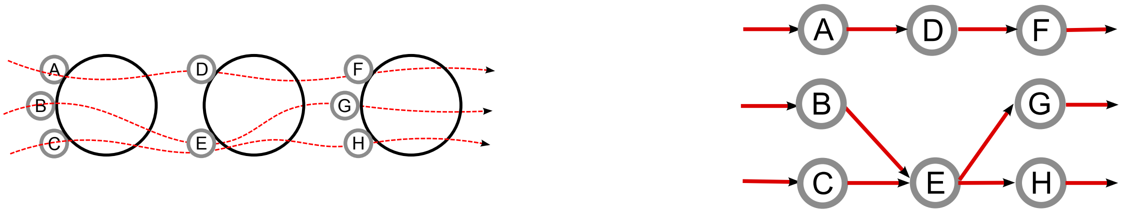


Fig. 1. Example network with three `Traceroute` measurements, where black balls are routers, gray balls are interfaces, and dotted paths are `Traceroute` observations. (Left) - Observed paths through the network, $\{A, D, F\}, \{B, E, G\}, \{C, E, H\}$, (Right) - Inflated inferred topology without interface disambiguation.

TTL responses. Additionally, the work of Sherry et. al. in [10] uses pairwise probes with an IP timestamp option. While these pairwise probing-based alias resolution techniques can result in relatively accurate estimates, the resulting large network load caused by the additional probing infrastructure limits their application for Internet-scale analysis.

Recent work has focused on exploiting router properties to reveal aliases. MRinfo [11] probing and the MERLIN project [7] have explored interfaces of multicast-enable routers, while the Discarte project by Sherwood et. al. [5] combines inferred router-level topology from `traceroute` probes with information extracted from the Record Route IP option. While these techniques can be highly accurate, they are restricted to the subset of routers with these specific configurations enabled.

Given the limitations with these prior probing and router-exploitation approaches, the ALIASCLUSTER methodology is motivated by prior graph-based and data mining approaches to the interface disambiguation problem. The closest prior work to our approach is the APAR/KAPAR framework [12], [13] which explores router IP addresses with features such as hop distance, same path assumption, common neighbor, and subnet alignment. In contrast, we take a Naive Bayesian data fusion approach (explored previously for MPLS identification [14] and IP Geolocation [15]), and develop a hierarchical clustering technique to resolve router aliases when not all the desired Internet measurements have been observed.

III. DATASETS

The use of our ALIASCLUSTER technique is predicated on a large collection of `traceroute` measurements. While widely available from numerous public repositories, including iPlane [4] and Dimes [2], for the experiments in this paper we will focus on measurements from the CAIDA Macroscopic Internet Topology Data Kit [1] taken in July 2010. This data sets consists of `traceroute` measurements from 54 monitors to a randomly chosen address in all routed /24 prefixes. This `traceroute` collection encounters 2,105,738 unique interface IP addresses in an unknown number of routers.

To evaluate the accuracy of our technique, we use a dataset of declared router aliases generated by the MERLIN Project [7]. Using MRinfo-based probing in September 2010, the MERLIN alias dataset crawls the Internet to discover multicast-enabled routers and uses MRinfo to return the valid interfaces associated with each of these routers. This data set contains 42,769 routers consisting of 400,040 unique IP addresses. From this collection, only 63,479 unique IP addresses in 19,027 routers are also encountered by the CAIDA `traceroute` data set. While the number of multicast-enabled routers with alias ground truth is limited, our particular focus on this ground truth is due to both the observation of true positives (if MRinfo declares that two interfaces are aliased, we can be certain they are aliased) and the observation of true negatives (where if interfaces i, j are aliased and m, n are aliased, we can be confident that i, m are *not* aliased).

IV. ALIASCLUSTER METHODOLOGY

To resolve router aliases, we introduce the ALIASCLUSTER methodology. This technique consists of two components:

- **Naive Bayesian Estimation of Pairwise Alias Likelihood** - For targeted interface pairs, we fuse `traceroute` extracted features and estimate the likelihood that two interfaces are aliased via an efficient Naive Bayesian likelihood estimation technique.
- **Confidence-Based Alias Clustering** - Using the estimated pairwise alias likelihoods, we perform a novel hierarchical clustering methodology for determining router alias clusters.

A. Naive Bayesian Estimation of Pairwise Alias Likelihood

To determine the likelihood that two router interfaces are aliased, we first extract relevant features of these two interfaces from the observed `traceroute` measurements. In addition to the feature of common IP subnet (which has been explored in previous disambiguation techniques, [6]), we also examine pairs of interfaces that have common out-degree and commonly observed hop counts. For interface pairs, we restrict these features to commonly observed down-path and hop counts that are within a small neighbor of both interfaces (*i.e.,* within either three or four hops), this ignores potentially non-informative distant down-path interfaces from corrupting our features. The five features we will consider in this paper[1] are detailed in Table I, with illustrated examples for two of these features in Figure 2.

By fusing information from all five features, we construct the likelihood that two interfaces i, j are aliased given the observation of our five extracted features using Bayes Rule

[1] We do not claim that this collection is the complete set of useful features. However these features can be easily extracted from `traceroute` measurements; and additional features can easily be added to the ALIASCLUSTER framework.

TABLE I
traceroute-DERIVED ALIAS FEATURES FOR A PAIR OF ROUTER INTERFACES.

Feature	Name	Description
$\mathcal{M}_1$	IP Subnet	Size of the common IP prefix
$\mathcal{M}_2$	Percent Out-Degree Match (hop count ≤ 3)	Fraction of down-path interfaces within three hops commonly observed
$\mathcal{M}_3$	Percent Out-Degree Match (hop count ≤ 4)	Fraction of down-path interfaces within four hops commonly observed
$\mathcal{M}_4$	Percent Hop Match (hop count ≤ 3)	Fraction of down-path interfaces within three hops with the same hop count
$\mathcal{M}_5$	Percent Hop Match (hop count ≤ 4)	Fraction of down-path interfaces within four hops with the same hop count

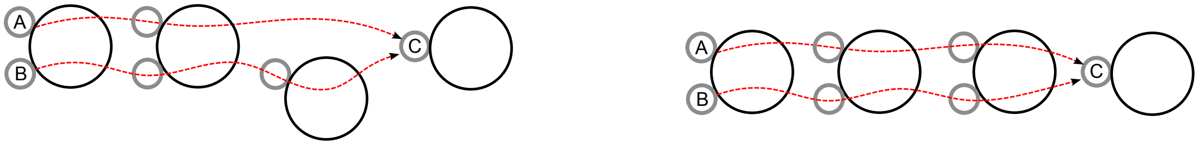


Fig. 2. Extracted feature examples from Traceroute measurements. (Left) - Out-degree match, where interface C has been observed on a down-path for both interface A and interface B (*i.e.,* where observed paths exists where one path contains interfaces $\{A, C\}$ and another path contains interfaces $\{B, C\}$), and (Right) - Hop count match, where interface C has both been observed on a down-path for interface A and interface B and the hop count between the two (in this case, two hops) is the same for both.

and a Naive Bayesian independence assumption,

$$\begin{aligned}
&P\left(i,j \text{ alias} \mid \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_5\}\right) \\
&= P\left(\{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_5\} \mid i,j \text{ alias}\right) P\left(i,j \text{ alias}\right) \\
&\approx P\left(\mathcal{M}_1 \mid i,j \text{ alias}\right) \cdot P\left(\mathcal{M}_2 \mid i,j \text{ alias}\right) P\left(\mathcal{M}_3 \mid i,j \text{ alias}\right) \\
&\qquad \cdot P\left(\mathcal{M}_4 \mid i,j \text{ alias}\right) P\left(\mathcal{M}_5 \mid i,j \text{ alias}\right) P\left(i,j \text{ alias}\right)
\end{aligned}$$

This Naive Bayesian assumption breaks up the problem from estimating a single five-dimensional distribution (which may require prohibitively many training observations [16]), to the more tractable problem of estimating five one-dimensional distributions. To estimate each one-dimensional distribution, we use the estimated kernel distributions from a training set of interface pairs with known router assignment as seen in Figure 5 and 6. As Figure 6-(Right) demonstrates, some features will be more valuable than others, therefore we can approximate the pairwise alias log-likelihood ratio (for two interfaces i, j) using a weighted approach and taking the difference between the aliased and non-aliased log-likelihoods.

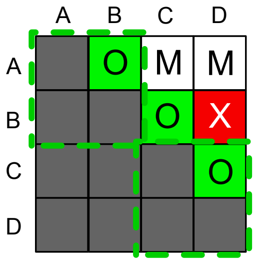
$$\begin{aligned}
s_{i,j} \;=\; &\sum_{k=1}^{5} w_k \log\left(P\left(\mathcal{M}_k \mid i,j \text{ alias}\right)\right) - \qquad (1)\\
&\sum_{k=1}^{5} w_k \log\left(P\left(\mathcal{M}_k \mid i,j \text{ not alias}\right)\right)
\end{aligned}$$

Where the weights for each extracted feature (*i.e,* $\{w_1, w_2, w_3, w_4, w_5\}$) are learned via a bisection search methodology on hold-out cross validation training sets.

B. Clustering-Based Alias Classification

Unfortunately, two limitations occur with the estimated pairwise alias likelihoods. First, routers often have more than two interfaces, meaning that there exists more than a single pair of interfaces at this router. Secondly, due to potential limitations in the traceroute measurement infrastructure, we may not have enough information to estimate alias likelihood for some interface pairs (*e.g.,* consider the case where two interfaces have no commonly observed down-path interfaces) resulting in missing pairwise likelihoods.

Using the incomplete set of pairwise alias likelihood, we want to determine which interfaces are aliased. One simple approach could be to threshold the pairwise alias likelihood values, and declare two interfaces to be aliased if their likelihood is above a specified value. A problem with this approach is the potential for "chaining" [17]. Chaining is the problem of resolving a string of large values resulting in a cluster, even though very little information is known between most of the items in the cluster. For example, consider four interfaces $\{A, B, C, D\}$ and the estimated likelihood values, $S_{A,B}, S_{B,C}, S_{C,D}$ are all above the threshold. This would imply that interfaces B and D are aliased together even if no information, or even negative alias information, has been observed. An example of this can be seen in Figure 3.



	A	B	C	D
A	■	O	M	M
B	■	■	O	X
C	■	■	■	O
D	■	■	■	■

Fig. 3. The pairwise likelihood matrix ($\mathbf{S}$) of four interfaces $\{A, B, C, D\}$ with the true aliases being $\{A, B\}$ and $\{C, D\}$. From the feature extraction and Naive Bayesian methodology, we observe $S_{A,B}, S_{B,C}, S_{C,D}$ as greater than some threshold (indicated by an 'O') and $S_{B,D}$ as below the threshold (indicated by an 'X'). Due to lack of measurements, likelihoods $S_{A,C}, S_{A,D}$ have not been observed (indicated by an 'M'). Due to chaining, all four interfaces may be clustered using the thresholding method.

To avoid these limitations, we developed a modified version of agglomerative clustering we call "Confidence-Based Clustering", where subsets of interfaces are declared as aliased only if enough information has been observed. The procedure has input confidence parameter $\alpha \in (0, 1]$, such that all clusters formed must have confidence greater than α, where confidence is defined here as the fraction of pairwise likelihoods of the cluster items that were observed.

The method proceeds as follows. Initially, all interfaces are considered to be singleton clusters; as clusters are formed we merge rows and columns of $\mathbf{S}$ (the matrix of pairwise log-likelihood ratios using Equation 1) so that every row/column continues to represent a single cluster. At each step of the algorithm, we find the pair of clusters corresponding to the largest pairwise likelihood and with confidence (*i.e.,* fraction of observed likelihood pairs with respect to all possible pairs in the clusters) above the specified confidence threshold α. This can be considered as the two clusters that are the most similar with confidence that enough measurements were observed to make an informed decision. We merge those two clusters, which defines a binary tree structure whose leaves are a subset of interfaces.

In terms of the likelihood matrix ($\mathbf{S}$), this merge operation is performed by creating a new row/column, i', such that for all rows and columns k, the new likelihood values are the maximum of the two observed values (*i.e.,* $s_{i',k} = s_{k,i'} = \max\{s_{i,k}, s_{j,k}\}$). The rows and columns associated with i, j are then removed from the matrix. This process is repeated until no further pairs exist with confidence above the specified threshold (or all the items have been merged together), defining a hierarchical clustering tree, $\mathcal{T}$.

Finally, given the resolved hierarchical clustering, we estimate a set of router clusters. This is performed by pruning the hierarchical tree structure given the maximum observed probability at each interior node, such that all interfaces with interior probability greater than some threshold are in the same router cluster. An example of this pruning methodology can be seen in Figure 4.

C. AliasCluster Methodology

The AliasCluster methodology combines the Naive Bayesian pairwise alias likelihood estimation of Section IV-A, and the confidence-based router clustering approach of Section IV-B to resolve router aliases. The complete methodology is summarized in Algorithm 1.

V. Results

To evaluate the AliasCluster methodology, we take a two stage approach. First we evaluate the discriminatory power of each of AliasCluster chosen `traceroute`-extracted features. Then, the performance of our clustering approach is analyzed with respect to ground truth aliases. As stated previously, we focus on router interfaces that have been found via MRinfo. This gives us the advantage of knowledge of both true aliases (*i.e.,* subsets of router interfaces which belong to the same physical router), along with confident false aliases (*i.e.,* two interfaces that are not declared as aliases but have been declared aliased elsewhere). We demonstrate that this ground truth with respect to both *true* and *false* alias pairs is critical for evaluating disambiguation performance.

To learn the alias detection characteristics of our data set, we perform hold-out cross validation, where at-random half of our interfaces are used as training data (with known alias assignment from the MERLIN ground truth), while the remaining half of our interfaces are used as test data to determine the accuracy our the AliasCluster technique.

```
Algorithm 1 - AliasCluster($\mathbf{T}, \alpha, \lambda$)
Given :
  1) Set of Traceroute paths, $\mathbf{T}$, containing $N$ unique
     router interface addresses, $\mathbf{X} = \{x_1, x_2, ..., x_N\}$.
  2) Clustering parameters : confidence threshold, $\alpha$, and
     likelihood pruning threshold, $\lambda$.
Process :
  • Given $N$ unique router interfaces in the traceroute
    paths, construct $N \times N$ likelihood matrix, $\mathbf{S}$.
      – For each pair of interfaces $i, j$, if at least one of the
        features from Table I is non-zero, use Equation 1
        to find the alias likelihood value for the pair of
        interfaces, $s_{i,j}$.
  • Find a hierarchical clustering, $\mathcal{T}$, where all cluster have
    confidence $\geq \alpha$.
Output :
      Return the set of router aliases by pruning the
      hierarchical clustering $\mathcal{T}$ using pruning threshold, $\lambda$.
```

As stated in the related works section, while numerous interface disambiguation methodologies have been introduced, here we will focus on comparing performance against the KAPAR methodology [13] due to its reliance only on `traceroute`-observed features. We use the Kapar version 0.2 implementation that is publicly available at [18].

A. Evaluation of Features

The basis for the AliasCluster technique is the choice of extracted features from observed Internet measurements. To begin, we assess how discriminatory these features are between aliased and non-aliased pairs of observed router interfaces. Using kernel density estimates, the probability distributions for these five extracted features can be seen for our training subset of the CAIDA ARK data set in Figures 5 and 6-(Left) and 6-(Center). With respect to ground truth alias data via the MERLIN data set, we find that IP subnet and out-degree generally have different values for alias and non-alias pairs, and that the hop count feature very clearly discriminates between aliased and non-aliased pairs.

The false alarm and detection rates for all five features are shown in Figure 6-(Right). The results follow from observations of the distributions of each feature, and also show that the fusion of all five features is more accurate than any single feature (*i.e.,* has fewer declared false alarms for a given detection rate). We find that while the hop count match feature has very low false alarm rate, this feature only detects a minority of the alias pairs – this motivates not relying too heavily on any single feature.

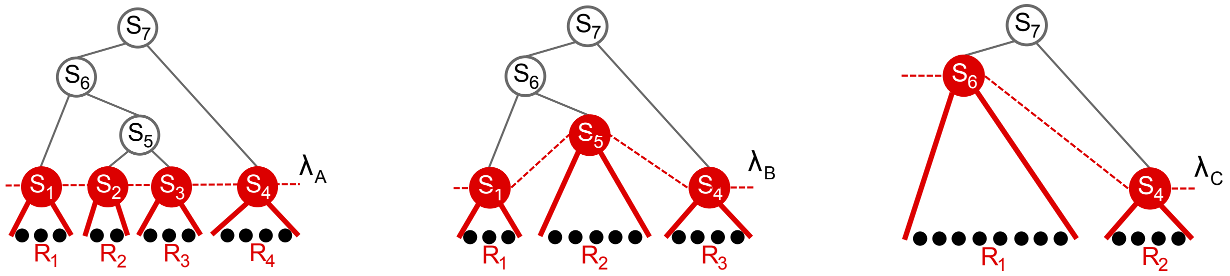


Fig. 4. Hierarchical Clustering pruning examples for $S_1 > S_2 > ... > S_7$, where the leaf nodes represent observed router interfaces. (Left) - Four routers found with threshold λ_A such that $S_4 > \lambda_A > S_5$, (Center) - Three routers found with threshold λ_B such that $S_5 > \lambda_B > S_6$, (Right) - Two routers found with threshold λ_C such that $S_6 > \lambda_C > S_7$.

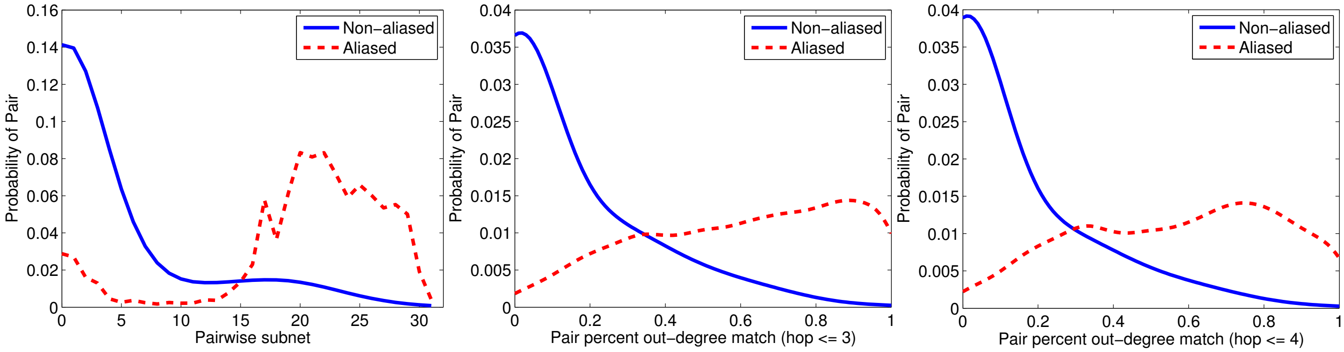


Fig. 5. Extracted feature distribution of alias and non-aliased interface pairs for (Left) - IP Subnet, (Center) - Out-degree match for hop ≤ 3, (Right) - Out-degree match for hop ≤ 4.

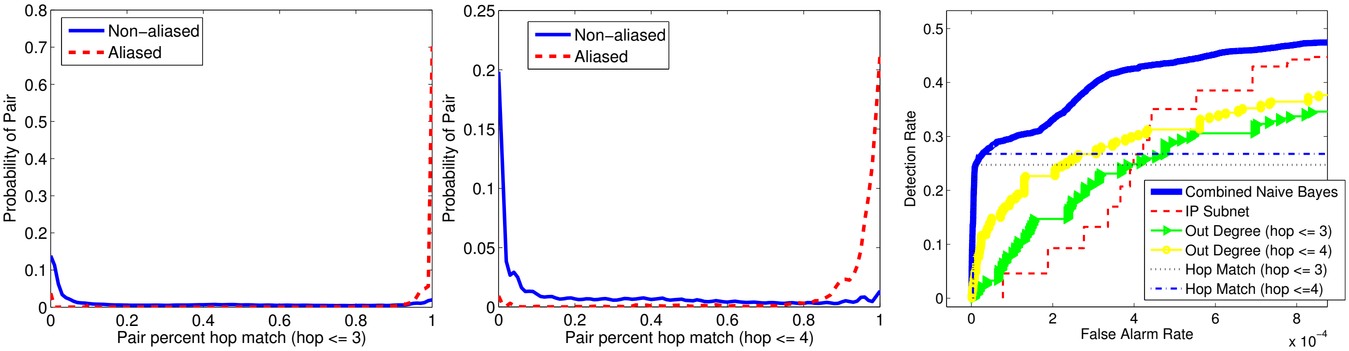


Fig. 6. Extracted feature distribution of alias and non-aliased interface pairs for (Left) - Hop count match for hop ≤ 3, (Center)- Hop count match for hop ≤ 4. (Right) - Detection and False Alarm characteristics for individual features and combined Naive Bayes likelihood (from Equation 1).

TABLE II
FALSE ALARMS RATES FOR SPECIFIED ALIAS DETECTION RATES FOR THE ALIASCLUSTER METHODOLOGY USING TWO DIFFERENT CONFIDENCE VALUES, $\alpha = \{0, 0.8\}$ ON SEPTEMBER 2010 CAIDA ARK DATA SET.

	Detection Rate				
Technique	10%	20%	30%	40%	50%
ALIASCLUSTER ($\alpha = 0$)	2.22×10^{-6}	7.92×10^{-6}	1.26×10^{-4}	7.58×10^{-4}	2.09×10^{-3}
ALIASCLUSTER ($\alpha = 0.8$)	2.11×10^{-6}	6.35×10^{-6}	7.68×10^{-5}	6.79×10^{-4}	2.10×10^{-3}

B. MERLIN-Based Validation Results

We now show the performance of ALIASCLUSTER and the competing KAPAR approach at estimating router aliases from observed `traceroute` measurements. We run our ALIASCLUSTER technique with two different values of the confidence parameter ($\alpha = \{0, 0.8\}$, where $\alpha = 0$ is standard hierarchical clustering and $\alpha = 0.8$ penalizes missing pairwise likelihoods) and using a range of likelihood threshold values (with λ chosen between the smallest observed likelihood and the largest observed likelihood), producing the detection and false alarm rates found in Table II. We find that for a detection rate of $\leq 40\%$, using the confidence-based technique (*i.e.,* $\alpha \neq 0$) results in a lower false alarm rate.

Unlike the ALIASCLUSTER technique, the KAPAR methodology returns a single set of router aliases, and thus only a single detection and false alarm rate (26.012% of total aliases pairs detected with false alarm rate of 3.3047×10^{-5}). Using the detection results from the KAPAR methodology, we compare the false alarm rates in Table III. For the same detection rate, we find the use of the confidence-based clustering in ALIASCLUSTER leads to improvements in the false alarm rate, with 50% fewer declared false alarms than the KAPAR approach. While the declared false alarm rate for KAPAR is higher than specified in [6], this deviation can possibly be explained by the much larger set of interfaces considered here.

TABLE III
INTERFACES DISAMBIGUATION ACCURACY ON SEPTEMBER 2010 CAIDA ARK DATA SET FOR KAPAR TECHNIQUE DETECTION RATE ($5,075$ TRUE ALIASED PAIRS FOUND IN THE TEST SET, 26.012% OF TOTAL ALIASED PAIRS).

Technique	Number of False Alarms Declared	Percentage of False Alarms Declared
KAPAR [18]	16,644	3.3047×10^{-5}
ALIASCLUSTER ($\alpha = 0$)	8,388	1.6655×10^{-5}
ALIASCLUSTER ($\alpha = 0.8$)	7,603	1.5096×10^{-5}

VI. CONCLUSIONS / FUTURE WORK

Current Internet discovery techniques are limited by the unresponsive routers, heavy network load, and aliased router interfaces. The ability to mitigate any of these limitations would result in more accurate and useful Internet topologies for the application of traffic analysis, network security, and provisioning. We presented the ALIASCLUSTER framework for resolving interface IP addresses that belong to the same physical router. This technique uses extracted features from existing `traceroute` measurements, in combination with a confidence-based agglomerative clustering technique. On real-world Internet studies, we find that ALIASCLUSTER returns a false alarm rate improvement of over 50% compared with competing methods. For future work, we will look to supply the community with a tool based on the AliasCluster methodology, perform a longitudinal study by examining estimated aliases from past topology data sets, and integrate additional features in the ALIASCLUSTER framework, including IP-ID observations.

ACKNOWLEDGMENTS

This work was partially supported by NSF grants CNS-0905565, CNS-1018266, CNS-1012910, and CNS-1117039, and by ARO grant W911NF-11-1-0227.. Any opinions, findings, conclusions or other recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the NSF or the ARO.

REFERENCES

[1] CAIDA, "The CAIDA UCSD IPv4 Routed /24 Topology Dataset," http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml, July 2010.

[2] Y. Shavitt and E. Shir, "DIMES: Let the Internet Measure Itself," in *ACM SIGCOMM Computer Communications Review*, vol. 35, October 2005.

[3] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP Topologies with Rocketfuel," in *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, February 2004, pp. 2–16.

[4] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: An Information Plane for Distributed Services," in *Proceedings of USENIX OSDI*, Seattle, WA, November 2006.

[5] R. Sherwood, A. Bender, and N. Spring, "Discarte: A Disjunctive Internet Cartographer," in *Proceedings of the ACM SIGCOMM*, August 2008, pp. 303–314.

[6] K. Keys, Y. Hyun, M. Luckie, and k. claffy, "Internet-Scale IPv4 Alias Resolution with MIDAR," in *To appear in IEEE/ACM Transactions on Networking*, 2012.

[7] P. Merindol, B. Donnet, J. Pansiot, M. Luckie, and Y. Hyun, "MERLIN: MEasure the router level of the INternet," in *EURO-NGI Conference on Next Generation Internet*, June 2011, pp. 1 – 8.

[8] N. Spring, M. Dontcheva, M. Rodrig, and D. Wetherall, "How to Resolve IP Aliases," in *University of Washington Technical Report*, 2004.

[9] A. Bender, R. Sherwood, and N. Spring, "Fixing Ally's Growing Pains with Velocity Modeling," in *Proceedings of ACM IMC*, November 2008, pp. 337–342.

[10] J. Sherry, E. Katz-Bassett, M. Pimenova, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy, "Resolving IP Aliases with Prespecified Timestamps," in *Proceedings of ACM IMC*, November 2010, pp. 172–178.

[11] J.-J. Pansiot, P. Mérindol, B. Donnet, and O. Bonaventure, "Extracting Intra-Domain Topology from MRinfo Probing," in *Proceedings of the Passive and Active Measurement Conference*, April 2010, pp. 81–90.

[12] M. H. Gunes and K. Sarac, "Resolving IP Aliases in Building Traceroute-Based Internet Maps," in *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, December 2009, pp. 1738–1751.

[13] K. Keys, "Internet-Scale IP Alias Resolution Techniques," in *SIGCOMM Computer Communications Review*, vol. 40, no. 1, January 2010, pp. 50–55.

[14] J. Sommers, P. Barford, and B. Eriksson, "On the Prevalence and Characteristics of MPLS Deployments in the Open Internet," in *ACM Internet Measurements Conference*, November 2011, pp. 445–462.

[15] B. Eriksson, P. Barford, , J. Sommers, and R. Nowak, "A Learning-based Approach for IP Geolocation," in *Proceedings of Passive and Active Measurements Conference*, Zurich, Switzerland, April 2010.

[16] L. Wasserman, "All of Nonparametric Statistics (Springer Texts in Statistics)." Springer, 2010.

[17] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

[18] CAIDA, "The KAPAR Project," http://www.caida.org/tools/measurement/kapar/, 2012.