Internet Address Space Clustering for Intelligent Route Control – DRAFT

Geoffrey Brown

***Abstract*— An intelligent route controller optimizes traffic from a local network to a set of Internet address regions called clusters by making routing decisions based upon performance measurements to representative points for each cluster over multiple Internet connections. This paper describes a method for clustering the Internet address space based upon structural (routing table based), topological, and temporal techniques.**

I. Introduction

This paper describes a strategy for for clustering the Internet address space to support intelligent route control. The basic application is illustrated in Figure 1. An intelligent route controller optimizes traffic from a subset of the Internet (IP) address space to a set of non-overlapping regions called clusters. Each cluster has a set of associated measurement points (scanpoints) which may be, but are not necessarily, identified by addresses within the cluster. To make routing decisions from the optimized network to a cluster, the route controller performs a series of measurements to the scanpoints over multiple connections to the Internet. Depending upon the results of these experiments the route controller may cause local routing to prefer one Internet connection for traffic destined for the cluster.

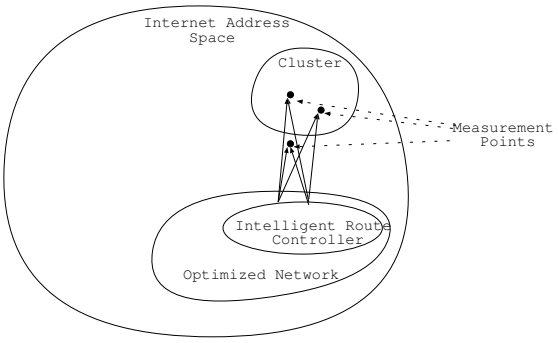

Fig. 1
Clustering for Intelligent Route Control

From the perspective of the route controller, a cluster can be viewed as an equivalence class and the scanpoints as proxies for the members of that class. A fundamental assumption is that the expected performance of traffic to destinations within the cluster over a given Internet connection will be equivalent for all possible destinations; call this the *Uniformity Assumption.* For an intelligent route controller there is a trade-off between the accuracy of the uniformity assumption and the amount of measurement that must be performed. In particular the desired accuracy of the uniformity assumption may vary in different clusters depending upon factors such as the fraction of traffic from the subnet being optimized to the cluster and the geographical distance from the subnet being controlled to the cluster.

The process used in this paper combines features of three IP address clustering techniques:

- Structural
- Topological
- Temporal

Structural clustering assumes that BGP routing tables provide natural clusters with two levels of granularity – autonomous systems (AS's) and prefixes. There are three deficiencies with this approach. The two granularities are either too coarse or too fine depending upon the application. Even with the finer granularity level (prefixes) the accuracy of the uniformity assumption can be poor. For example, prefixes exist that span the entire USA. Finally, there are frequently groups of addresses in different prefixes that could be clustered while satisfying the uniformity assumption. Nevertheless, structural clustering may be combined with other techniques to fill in holes in the available data. For example, the other approaches require gathering data on a sample of Internet addresses. Where only one sample falls in a BGP prefix, the associated data may be *generalized* to all addresses within the

prefix.

Topological clustering uses information inferred by performing traceroutes from various sources to a wide number of destinations. Two destinations are considered to be "near" if traceroutes to them intersect at common points within a few hops of the destination. [2] provides an excellent foundation for understanding how much measurement might be required to provide a reasonably complete topological map of the Internet. The most significant failing of the topological approach is that there is no natural way to vary the clustering granularity because the requirements for adjacency are so strong. Hops in the sense of traceroutes really correspond to interfaces. Two traceroutes can traverse the same router but not the same interface and the traceroutes will not be considered to merge at the common router. Two traceroutes to destinations in the same geographical region may traverse routers that are in the same POP (point of presence) and yet never go through identical routers. Often, no common hop is found for a single destination, and destinations that are near both geographically and topologically are considered distinct.

Topological techniques also tend to lead to coverage problems, i.e. after generalization large regions of the Internet address space fall in no cluster. The coverage problem for traceroute based techniques arises because two regions can only be considered adjacent if they share a common hop. If no common hop exists then we have no distinguishing information – we can't say two regions are relatively close or relatively far from each other. Thus the "shared hop" relationship is a strong but rather sparse equivalence relationship.

Temporal clustering utilizes temporal measurements between a set of S servers and a number of destinations. These measurements may be gathered actively by sending probes (e.g. pings) to the destinations or passively by monitoring traffic (e.g. TCP syn-ack intervals). The result of these measurements is an S-dimensional "distance" vector for each destination. These distance vectors provide a convenient basis for determining whether two destinations are near (e.g. through Euclidean distance). Clustering may be performed in a number of ways including subdividing the BGP prefixes until all "fractional prefix" clusters contain destinations meeting some distance requirement. Alternatively, the distance vectors could be used to estimate the physical distance of each destination from a set of known reference points and clusters built in relationship with these reference points.

The major contributions of this work over previous IP address clustering techniques are:

- Our technique supports variable cluster granularity with the option to use finer granularity for regions of the Internet address space for reasons such as greater structural complexity or greater traffic.
- The clusters generated by our technique may contain multiple contiguous address blocks and may span multiple BGP prefixes.
- Our technique, while primarily based upon temporal measurements, utilizes topological clustering to minimize the amount of temporal measurement required, and structural clustering to generalize the results to larger address blocks.
- Our technique orders representatives of a cluster by their proximity to the cluster center and thus provides a mechanism to determine the most representative points for utilization by our route control system.

The remainder of this paper is organized as follows. In Section II we discuss our clustering techniques in detail and present experimental results in Section III. A central assumption of our work is that temporal clustering is sensible because a reasonable relationship exists between measured round-trip times and distance. We present data to support this assumption in Section IV. We discuss related work in Section V and conclude with a discussion of issues and extensions.

II. Internet Address Clustering

The process described in this paper involves selecting a set of representative IP addresses, which we call seedpoints, and then collecting multiple round trip time measurements to each seedpoint from a distributed network of S servers. The measurements from a single server to a single seedpoint are aggregated to provide one dimension in an S dimensional coordinate for the seedpoint.

The collection of seedpoints, along with their S dimensional coordinates are clustered using traditional data clustering techniques. Because traditional data clustering techniques allow control over the number of clusters, we are able to trade accuracy against granularity. Furthermore, the temporal

measurements provide a natural way to determine relative closeness between two seedpoints. Given a cluster, we can determine which of the seedpoints is nearest the centroid (mean or median point) of the cluster. This provides a convenient mechanism for choosing the most representative seedpoint(s) for each cluster. The output of this process is a set of clusters defined as an ordered list of seedpoints for each cluster. We then "generalize" the results to groups of addresses.

As mentioned in Section I, topological clustering techniques can indicate when destinations are likely to be near to each other, but suffer from "false negatives" which leads to granularity and coverage issues. Our work does not abandon the use of topological data, but uses temporal measurements to augment these data. We perform a topological clustering step on the seedpoints before we perform the temporal measurements. This topological clustering identifies groups of "equivalent" seedpoints. In the measurement and subsequent temporal clustering phase, we utilize only one representative from each equivalence group. In the subsequent generalization phase, we use the full set of seedpoints where the cluster information of each representative seedpoint is assigned to all members of the equivalent group.

In the following we elaborate on our process which consists of six major steps

1) Seedpoint Selection
2) Topological Clustering
3) Measurement
4) Temporal Clustering
5) Generalization
6) Scanpoint Selection

Once the seedpoints have been clustered and ranked based upon their distance from the cluster centroid, scanpoint selection for the intelligent route controller can be performed by performing a traceroute to the most representative seedpoint of each cluster and then selecting an intermediate hop as a scanpoint for direct or TLL limited probing. As we shall show, the manner in which our clustering system is implemented generates scanpoints as a side effect.

We begin with a discussion of each of six major steps described above and then point out specific design decisions in the implementation used in our work. We present results based upon the our process in Section III

A. Seedpoint Selection

The goal of seedpoint selection is to select a set of representative addresses from the routable IP address space that respond to the probes used for temporal measurement. An important design decision is determining the finest granularity required. For example, most routes in a BGP routing table cover at least 256 addresses (/8 ... /24 prefixes); a reasonable assumption is that structure finer than /24 prefixes is uninteresting. [1] In the following we discuss two basic approaches – using seedpoints collected from traffic statistics, and utilizing brute force search.

For a route optimization service where an intelligent route controller is placed at each customer site, collecting traffic statistics is relatively simple. For example, we collect traffic data at our customer sites that allow us to identify all of the active /24 prefixes in their traffic. One strategy would be to search for an active endpoint in each of these /24 prefixes. Alternatively, representative points could be harvested from web logs.

Seedpoint selection can also be performed using brute force techniques. There are currently roughly 4.7 million routable /24s in a typical BGP table. [20] demonstrated experimentally that testing a particular sequence of 11 destinations within an active /24 has a 90% chance of finding a live endpoint; roughly 50 million addresses need be tested to find seedpoints in a high fraction of active /24 address blocks.

B. Topological Clustering

Many of the seedpoints generated during seedpoint discovery can be shown to be equivalent using topological techniques where traceroutes are performed from one or more sources to each seedpoint; seedpoints that are determined to be topologically close are deemed to be equivalent. Only one representative from each equivalence class need be selected for temporal measurement. As we shall show later, even a simple heuristic results in a 4:1 reduction in the amount of measurement required.

As mentioned previously, topological clustering has the deficiency of being too fine grained. Temporal clustering provides a natural mechanism for coarsening the results of topological clustering.

[1] A recent RouteViews BGP table had only 123,000 addresses covered by prefixes more specific than /24 [16].

C. Measurement

There are a number of techniques that can be used to measure round trip times to a seedpoint. These include measuring the ICMP echo request/response period to the seedpoint with a ping tool, measuring the time to establish a TCP connection, or using a TTL limited probe sent to the seedpoint, but with a TTL smaller than required to reach the destination.

D. Temporal Clustering

Temporal clustering utilizes standard data clustering techniques such as those described in [10]. There are many published approaches to data clustering that may be applied including divisive and agglomerative hierarchical techniques and iterative techniques such as Kmeans.

The general strategy of temporal clustering is to associate each representative seedpoint with the data obtained by measuring the latency from multiple servers to the seedpoint (or its proxy scanpoint). Seedpoints that have similar measurements are considered to be equivalent. For example, if measurements are made from three servers S_1, S_2, and S_3 to destinations D_1 and D_2 as illustrated in Figure 2, we obtain two sets of "coordinates:" (T_{11}, T_{21}, T_{31}), (T_{12}, T_{22}, T_{32}). We can compute the Euclidean "distance" between D_1 and D_2 using the formula:

$$distance_{12} = (\sum_{i=1}^{3} |T_{i1} - T_{i2}|^2)^{1/2}$$

Other distance metrics include Manhattan distance.

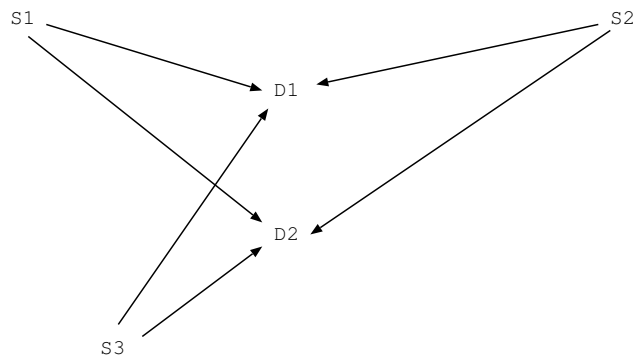

Fig. 2
MEASUREMENT FROM THREE SOURCES TO TWO DESTINATIONS

As an example, we performed measurements from four servers located in Seattle, San Jose, Boston, and Atlanta, to a set of 806 scanpoints associated with AS3356 (Level 3). These scanpoints were generated using a process described in Section II-G. Briefly, we searched for a set of seedpoints, selected representatives based upon topological clustering, and chose scanpoints "in front" of these representatives based upon traceroutes. From the 806 scanpoints we associated with AS3356 we randomly selected 25 for the following clustering example. The primary reason for selecting AS3356 is that Level 3 tends to give geographically significant names to their routers. Using a set of tools implementing standard agglomerative hierarchical clustering algorithms [12], we clustered the points into five groups. The particular algorithm used was "complete link" with Euclidean distance. Complete Link clustering determines the "similarity" of two clusters based upon the distance between the most distant members. The results of this clustering process are illustrated by the dendrograph in Figure 3.

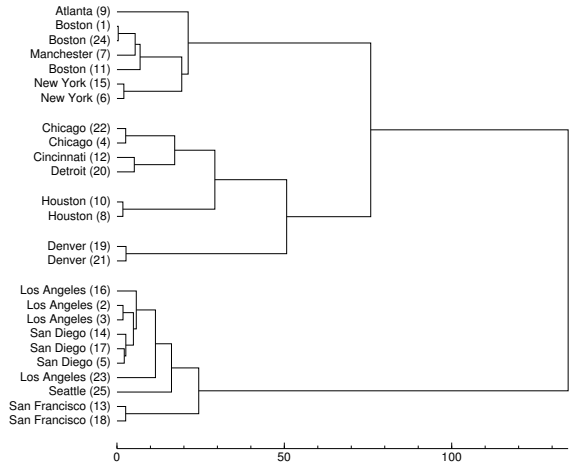

Fig. 3
CLUSTERING SCANPOINTS IN AS3356

The dendrograph lists the scanpoint locations (as determined through DNS lookups and traceroutes) on the left. The numbers in parenthesis are provided in place of IP addresses for use in the following discussion. The graph shows the result of recursively combining clusters until a single cluster is reached at the right. The bottom provides a scale indicating the maximum Euclidean distance (in milliseconds) between the farthest two members of a cluster. The groupings correspond to the clusters generated if clustering terminates at 5 clusters. In this example,

limiting the results to 5 clusters results in clusters covering relatively large geographic regions.

As an additional output of the clustering process, we can generate an ordered list of the scanpoints used to generate a cluster. These scanpoints are ranked by the distance to the cluster centroid (mean or median). This ordering provides a mechanism for choosing the most representative points in a given cluster for use in our route optimization service. For the example illustrated in Figure 3, the scanpoints nearest the centroid of each cluster are:

Cluster 1: Boston (11)
Cluster 2: Detroit (20)
Cluster 3: Houston (10)
Cluster 4: Denver (19)
Cluster 5: Los Angeles (2)

E. Generalization

The goal of the generalization phase is to expand the clustering results which generated equivalence classes containing specific IP addresses (i.e. the seedpoints) into equivalence classes containing blocks of IP addresses. The generalization phase combines the structural information obtained from a set of BGP routing tables with the clustering information from topological and temporal clustering.

Generalization is performed upon the seedpoints in each cluster. Generalization then utilizes a prefix table formed from aggregating multiple BGP tables to perform structural generalization. This process may be performed either top-down or bottom-up. In the top-down process, each seedpoint is assigned to the longest matching prefix. Prefixes that have been assigned multiple seedpoints in different clusters are divided into two more specific prefixes and the seedpoints reassigned to the longest matching prefix. This process continues until no prefix has been assigned seedpoints from multiple clusters.

As an example, consider top down generalization of BGP prefixes as illustrated in Figure 4. Assume that the seedpoints have been associated with clusters as described previously, and have then been associated with a set of prefixes using the standard longest matching prefix approach. In this example, consider that there is a single initial prefix 63.200.0.0/16 with which three seedpoints have been associated. Each seedpoint is shown along with an integer representing the cluster to which it belongs. In this case one iteration is required before the termination condition is met – that no prefix has seedpoints associated with it from more than one cluster.

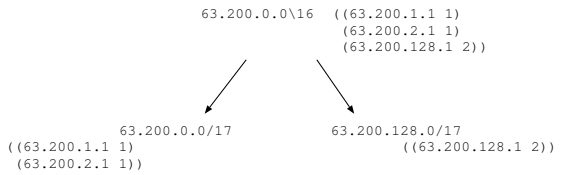

Fig. 4
TOP DOWN GENERALIZATION

The bottom up approach begins by assigning each seedpoint to its corresponding /32 prefix. The process proceeds by merging adjacent prefixes that contain seedpoints from at most one cluster. The merging process forbids merging across the BGP prefix boundaries and terminates when no more merges are permitted.

F. Scanpoint Selection

Although the measurement process described in Section II-G yields scanpoints as a natural side effect, it may be beneficial to generate scanpoints from each intelligent route controller over each of its Internet connections. The clustering technique provides for each cluster a ranked list of seedpoints. To find a scanpoint for a given seedpoint over a particular Internet connection, the intelligent route controller can perform a traceroute to the seedpoint over the Internet connection. A scanpoint is then selected from one of the traceroute hops and measurements performed either directly to the scanpoint or indirectly by using TTL limited probes aimed at the seedpoint.

This technique provides the opportunity to generate multiple scanpoints for important clusters and to generate “backup” scanpoints in case some scanpoints cease responding to measurement probes.

G. Implementation Details

Our current implementation attempts to identify one seedpoint for each active addressable /24 in a set of BGP routing tables with the exception of regions containing .gov or .mil domains. In practice only a fraction of the /24s in a routing table contain active endpoints. The seedpoint selection process

consists of searching for an endpoint in each /24 that responds to ping requests. Our experience suggests that the .1 addresses in roughly 50% of active /24s respond to ping requests. This is confirmed by [20] which also provides a particular sequence of 11 addresses within an active /24 to test in order to find live endpoint. Our implementation searches for seedpoints using a subset of these addresses.

An important step in our process is to utilize topological clustering techniques to reduce the amount of temporal measurement required. For the work discussed in this paper we perform a single traceroute to each seedpoint from a single source. The endpoints of traceroutes that contain the same penultimate hop are deemed to be to equivalent. An obvious extension is to treat two endpoints as equivalent if they are within some temporal distance of a common hop.

In our implementation, we select the penultimate hop on the traceroute to the seedpoint performed in the topological clustering phase as a measurement point (scanpoint). Because of issues relating to multi-homing, we reject scanpoints that are not in the same AS as their corresponding seedpoint.

Our implementation performs one ping/hour to each destination from each source for 24 hours. There is nothing fundamental about either the frequency or duration of the measurement process; our goal is to establish a good estimate of the minimum round-trip time between each source and each scanpoint. The resulting pings are aggregated and scanpoints that do not respond to ping requests in a reliable and stable manner are discarded. For each source and destination, we choose the minimum ping response time. We then form an S dimensional distance vector for each destination (assuming S sources). While our implementation performs measurements to scanpoints, alternative implementations might measure directly to representative seedpoints.

As mentioned above, we discard scanpoints that do not respond reliably to pings. For example, roughly 7% of scanpoints do not respond at all to pings. We also reject scanpoints for which there is too much temporal variation.

Some scanpoints respond reliably to pings but with excessive delay. This results in outliers in the clustering process. This situation can be handled by finding the closest source from the ping experiments and rejecting any destinations that are farther than some maximum “distance” from their closest source; this threshold may differ for the various sources.

Our clustering implementation performs agglomerative complete link clustering on a per AS basis. For each AS we determine the number of clusters to be generated based upon the complexity of the AS (e.g. in terms of the fraction of unique scanpoints within that AS). Alternatively, the cluster “budget” for an AS might be based upon customer traffic statistics. Where several seedpoints have been deemed to be equivalent through topological means, only one representative is used during the clustering process.

In order to eliminate outliers, we cluster each AS twice – first with a larger cluster budget than our target budget. After the first clustering, we discard scanpoints that fall into clusters containing too small a fraction of the scanpoint total. We then re-cluster the remaining scanpoints.

As a refinement to this process, the clustering process can be constrained to respect geographical regions. In particular, the measurement servers can be partitioned into geographical regions and each scanpoint associated with the region of its nearest server. Only scanpoints in the same region are permitted to be placed in the same cluster. It is reasonable that in clustering a particular region, only the data from a subset of the servers will be used.

III. Experimental Results

In this section we describe the results of performing the clustering techniques described using data collected on 10/11/2002 from servers in Hong Kong, Seoul, Seattle, San Francisco, Chicago, Atlanta, London, Madrid, and Frankfurt. Data from servers in Sydney and Boston were discarded due to high ping loss rates. The obvious objective of this section is to demonstrate that the techniques described are effective. However, a secondary objective is to define methods for evaluating the results of any clustering technique for the purposes of route optimization.

We begin with a brief discussion of how the data was collected. We then elaborate on the specific heuristics and parameters used to cluster the data. Finally, we evaluate the quality of the clustering results using a variety of tests.

A. Data Collection

Seedpoint discovery and cluster generalization were performed using a composite BGP table downloaded from RouteViews [16] on 9/27/2002. This table contains prefixes covering 4,663,903 /24s. We deleted /24s from the seedpoint discovery process that fell within a "restricted list" of .gov and .mil domains that was generated from ARIN and other databases. The result was a set of 4,143,103 /24s. Seedpoint discovery was performed from Boston using the process described in Section II-A which tested a sequence of four addresses (.1,.129,.254, .2) in each /24. We were able to successfully perform ICMP trace routes [2] to addresses in 745,000 /24s of which 584,000 had penultimate hops in the same AS as the seedpoint. These 584,000 seedpoints had 149,000 unique penultimate hops (scanpoints).

On 10/11/2002, ping data were collected hourly for each of the scanpoints from each of the servers. We discarded data for any scanpoint where the ratio of 33rd percentile/min ping values exceed 1.5 for any server. To illustrate the effect of this filtering, consider Figure 5 which illustrates the cumulative probability function of the median/min ping values for measurements between all (source,destination) pairs in our data set where at least 50% of the pings were successful. For the remaining 127,381 scanpoints we selected the minimum ping time for each of the 9 servers.

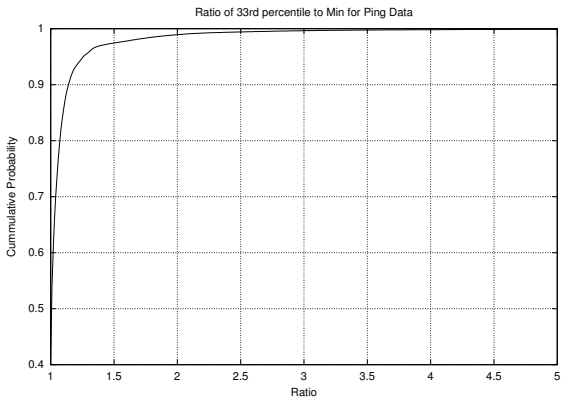

Fig. 5
PING DATA

[2]Trace routes were performed with a custom tool that used TTL limited ICMP echo request messages.

B. Clustering

To perform the clustering on our test data we utilized a set of freely available tools [12] that perform hierarchical agglomerative clustering. The underlying clustering algorithm utilized Euclidean distance with "complete link" clustering. The clustering budget for each AS was defined using the following heuristic – 20 clusters for each 1% of the total scanpoints in the AS with a minimum of 1 cluster/AS. There really is no scientific justification for this heuristic; however, it satisfied the measurement budget of our route optimization system and does appear to be related to the "complexity" of the ASs being clustered. As mentioned previously, one characteristic of our technique is that the clustering budget is selectable and may be based upon various criteria.

We performed clustering on ASs with budgets greater than one cluster twice – the first time with twice the cluster budget and the second time with the cluster budget. After the first clustering round we discarded the scanpoints from any cluster with fewer than four points. While this is an arbitrary criterion, the expected number of points/cluster with a uniform distribution is 70. Outlier elimination is an area requiring additional work; however, our experience suggests that some method for outlier elimination is required. This process resulted in 6759 clusters in 5859 ASs.

The output of the clustering process was generalized using a bottom up implementation of the process described in Section II-E driven by the aggregate BGP table downloaded from RouteViews with any prefixes more specific than /24 deleted.

C. Analyzing the Results

In analyzing our results we address three basic questions:

- Coverage – do the resulting clusters represent a significant fraction of the Internet address space?
- Consistency – are the clusters "tight" and do the clusters make geographical sense?
- Uniformity – is the scanpoint nearest the cluster centroid a reasonable performance proxy for arbitrary endpoints within the cluster?

Each of these has to be evaluated with respect to the number of clusters – we are interested in how good clustering is for a given "budget." For

example, if we simply cluster the set of seedpoints that share common penultimate hops where the seedpoint and penultimate hop reside in a single AS then we get coverage results that are similar to those described in this section without significant worries about either consistency or uniformity. However, this results ∼150,000 clusters in contrast with the 6759 generated by our techniques.

1) Coverage Analysis: Coverage analysis attempts to determine the fraction of the routable IP address space that is included in some cluster. This question can be addressed in either a static sense – the actual fraction of routable space – or in relation to a weighted traffic model (we call this dynamic). The problem with static analysis is that there are significant inefficiencies in address space utilization[3] and the importance of holes in cluster coverage depends upon where they fall. Dynamic coverage is a better measure, for a given traffic model, of the effectiveness of a set of clusters. It has the deficiency that the it depends upon the relevance of a traffic model to a particular use of the cluster data.

In practice, we generalize the results for any AS that contains a single cluster to all of the address space advertised by that AS. Our experience suggests that, except for a few hundred ASs, most autonomous systems should be treated as "point like" and hence needn't be subdivided into smaller clusters. For these "point like" ASs, the purpose of the measurement process is to attempt to identify a seedpoint (or scanpoint) that can be viewed as near the cluster centroid. Since AS "coverage" is strictly better than prefix coverage, the expected coverage lies somewhere between the two.

Thus we have four metrics for coverage – dynamic and static coverage of address space and ASs. Our static coverage statistics were 5859/13757 AS (42%) and 1.92 Million/4.1 Million /24s (47%) (recall that we consider only those /24s that fall outside of .gov and .mil regions).

Dynamic coverage is specific to a particular traffic model. In the following table we present statistics based upon Netflow data for outbound traffic accumulated for an hour at various customers.

[3]consider the current ownership of class C space where /8 prefixes are controlled by enterprises with significantly fewer than 2^{24} hosts.

Customer	Prefix Hit Rate	AS Hit Rate
1	73.1 %	92.9 %
2	85.7 %	95.6 %
3	75.4 %	89.7 %
4	76.7 %	93.2 %
5	69.6 %	84.3 %
6	84.8 %	95.3 %

In general it is reasonable to expect dynamic coverage statistics to be higher than static statistics for most traffic models. AS level coverage is higher because the technique used for "generalization" is more permissive.

2) Consistency: The goal of analyzing the consistency of a set of clusters is to establish that each cluster contains a "reasonable" set of seedpoints and that the set of clusters generate provide "reasonable" diversity. Establishing consistency is complicated by the fact that we do not know, in general, were the seedpoints clustered by our techniques are or how they fit into the overall Internet topology. In a limited number of cases, ISPs have chosen to use geographically significant names for their routers and have also published network maps. In these cases we can analyze the clustering output with respect to the available geographical information.

AS 3356 (Level 3) apparently utilize geographically significant machine names. In those cases where the name is either non-specific, our not available, it is possible to establish the location of "adjacent" machines using traceroute. The experiment discussed in this paper generated 23 clusters for AS 3356. For each cluster generated by our techniques, we generated a list of scanpoints ordered by their distance from the cluster centroid. In Figure 6 we highlight those cities where each cluster centroid falls (some cities had more than one cluster).

The 23 clusters had an average of 67 scanpoints and a median of 55. Selecting the median size cluster, we found scanpoints in Miami (24), Orlando (27), and Jacksonville (4) with the centroid in Orlando. In the largest cluster (305 scanpoints) we found scanpoints in New York (77), Weehawken (36), Stamford (34), Philadelphia (42), Princeton (9), Baltimore (19), Washington (56), Richmond (1), Pittsburg (11), and Unknown (20) with the centroid in New York. Notice that for both of these clusters, the maximum distance between cities is approximately 350 miles.

Another way to address the consistency issue without geographic knowledge is to study the

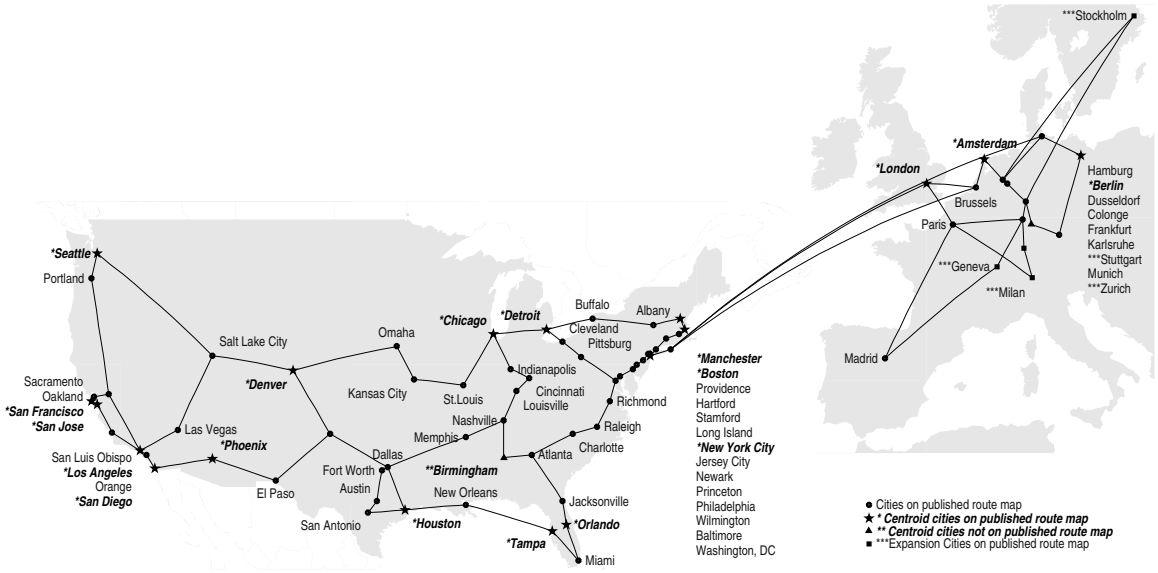

Fig. 6
GEOGRAPHIC COVERAGE AS3356[13]

"spread" of clusters in a temporal sense. Here we assume that there is a reasonable correlation between time and distance. While this may not hold in general, we are dealing with a situation where there is likely to be a large "common mode" path (that leading from a measurement source to the destination AS) with a somewhat smaller differential mode path. For AS3356, we have plotted in Figure 7 the spread in each cluster as measured from each of the 9 sources (23 * 9 data). For each data we also plot the mean measurement.

The data for AS3356 suggest relatively tight clusters with the RTT data to most clusters from single source varying by $\pm 10ms$ from the cluster mean. 10ms RTT corresponds to approximately 650 miles in fiber.

3) Uniformity Analysis: Ultimately, the most important quality metric for a cluster is that measurements to an appropriate scanpoint for a cluster from various sources provide similar relative behavior to measurements to destinations within the cluster from the same sources. We are explicitly not interested in "last mile" performance differences.

To evaluate the quality of our clusters for the purposes of performance measurement, we utilized a network of approximately 4100 destinations that we had access to through a partnership. All of these destinations are located in data centers and hence do not suffer from "last mile" effects. For each of these destinations we performed a traceroute to find the penultimate hop and we found the "best" scanpoint from our generalized cluster data. We discarded any destinations for which we had no scanpoint which left 2756 destinations with the following distribution: 638 data centers, 369 cities, 59 countries.

To each destination, penultimate hop, scanpoint we performed hourly pings for 24 hours from servers in: Seoul, Sydney, Hong Kong, San Francisco, Seattle, Chicago, Atlanta, Boston, London, Madrid, and Frankfurt. Recall that the Boston and Sydney servers where not used in clustering because of high ping loss rates at the time the clustering data were collected. For the purposes of performance analysis, they serve as "controls."

For each hour and server, we generated data pairs consisting of ping times for (destination, scanpoint) and (destination, penultimate hop) (scanpoint, penultimate hop) – we discarded any pairs for which one of the pings was lost. We then computed the correlation coefficient:

$$\frac{n\sum xy \;-\; \sum x \sum y}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

The resulting correlation coefficients were:

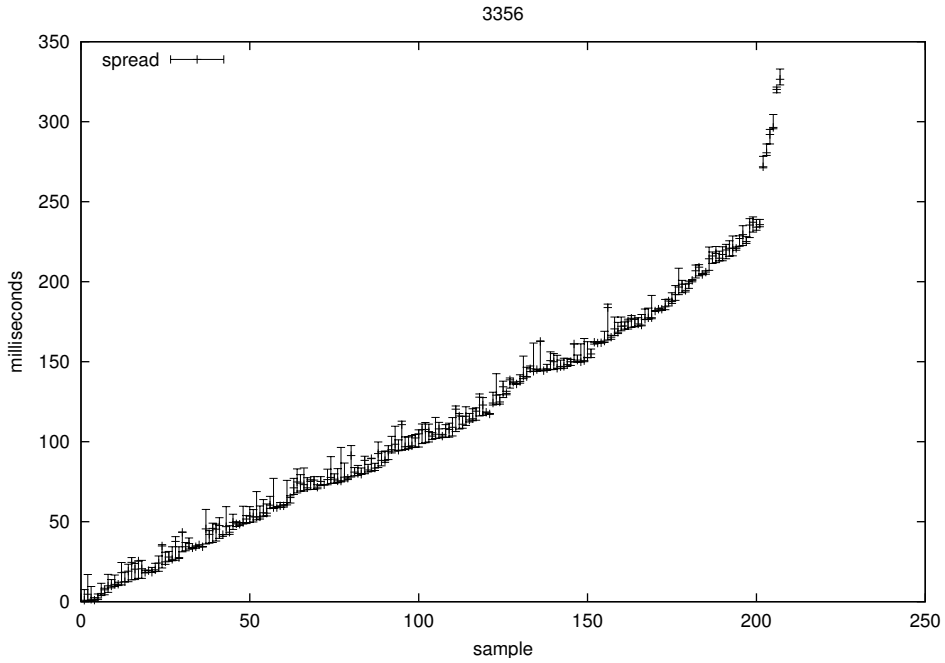

Fig. 7
CLUSTER SPREAD FOR AS3356

server	dest.-scan.	dest.-penult.	scan.-penult.
Seoul	0.83	0.91	0.82
Sydney	0.76	0.84	0.73
Hong Kong	0.83	0.91	0.83
San Francisco	0.79	0.90	0.77
Seattle	0.82	0.89	0.81
Chicago	0.80	0.90	0.81
Atlanta	0.80	0.91	0.80
Boston	0.66	0.72	0.66
London	0.83	0.91	0.82
Madrid	0.84	0.92	0.84
Frankfurt	0.84	0.91	0.84

As expected, the correlation between a destination and its penultimate hop was consistently better that the correlation between a destination and its scanpoint; however, the scanpoints generally had a good correlation with the corresponding destination. Notice that the results from Sydney and Boston were similar to those obtained with the servers that had been used in the clustering process; although they had the lowest correlation between the destination and scanpoint, they also had low correlation between destination and penultimate hop. Interestingly, the correlation coefficient between a scanpoint and a destination was very similar to the correlation coefficient between the scanpoint and the penultimate hop corresponding to the destination. This suggests the absence of "last mile" effects as expected.

IV. RELATIONSHIP BETWEEN RTT AND DISTANCE

The idea of using temporal measurements for clustering seems to depend upon the assumption that there is some reasonable correlation between distance and ping measurements. In this section we provide data that support this assumption. Similar results were presented in [9].

The data include ping measurements between 20 source clusters in 17 US cities and 878 destination clusters (in 499 destination cities, 368 in the US, and 56 countries). Each cluster consists of a small group of co-located machines which were considered equivalent in the data analysis. For each source-destination pair we obtained an average of 460 ping times from which we selected the minimum.

In estimating distance from RTT, we assumed that light in fiber travels (300/1.6) Km/msec (thus a refraction coefficient of 1.6 for fiber). Figure 8 presents histogram that illustrate the results for distances in three ranges $< 500km$, $500km < \wedge < 5000km$, and $> 5000km$. Clearly the connection between distance and time improves with distance.

This isn't too surprising given that the number of intervening routers is not linear in distance thus for longer distances a greater fraction of the time is spent in propagation through fiber.

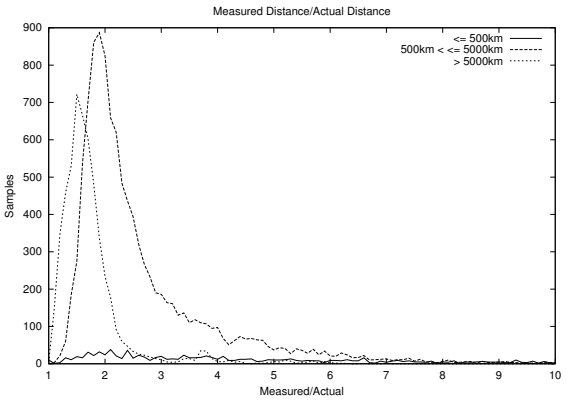

Fig. 8
HISTOGRAM OF MEASURED VS ACTUAL DISTANCES

In general, the data discussed in this section are "worst case." In our application, which only clusters within an AS, we can reasonably expect a relatively large "common mode" path between a source and seedpoints in a cluster. Thus, our clustering approach usually depends only upon the "differential" path within an AS. If the can assume that the routers being measured respond consistently and quickly then we will be able to obtain a good estimate of the difference in path lengths.

V. RELATED WORK

Structural clustering was previously considered in the context of content delivery networks [11]. We use structural clustering to aid in generalizing the results obtained through other clustering techniques, the methods described in this paper differ from previous work because they yield clusters that may span multiple structural clusters and the natural structural clusters may be divided among multiple clusters.

A good study of using temporal data for predicting the distance between points is presented in [14]. In [1] clever use of a modified DNS server is used to enable the "passive" collection of data from a set of customers accessing a set of geographically diverse web servers. The data collected from many web transactions is aggregated over time and used to form address clusters for the purposes of selecting the best web server in a content delivery network. The clusters are formed by aggregating smaller clusters (initially individual addresses) when the estimated latency from each server to each of the smaller clusters fall within some limits. There are fundamental differences between [1] and the work described in this paper. In [1] the clusters formed consist of contiguous blocks of addresses while our techniques allow non-continguous address blocks. In addition, our technques address the issue of trading cluster granularitry against uniformity. Finally, [1] attempts to discover clusters that are equivalent from the perspective of the servers performing the measurement, while our techniques are aimed at discovering clusters that are equivalent from the perspective of servers other than those performing the measurements.

[15] discusses a technique called "GeoPing" that builds a data base of temporal measurements from a set of S servers to a number of destinations at known locations. GeoPing then determines the geographical location of an arbitrary destination by measuring the latency from the servers to the destination and then finding the nearest known location by computing the Euclidean distance between the test destination and the known destinations in its database using the S dimensional distance vectors. [15] also discusses using structural clustering techniques to build a geographical database using measurements to a number of destinations with unknown locations. Specifically, the prefixes inherent in a BGP table are recursively split until all locations assocated with a prefix meet some maximum distance criterion. The fundamental differences with our work are that we generate clusters that may span multiple BGP prefixes and we provide techniques that trade cluster granularity against uniformity.

[6] describes a service, called IDMaps, to enable the estimation of distance between two arbitrary endpoints. The service operates using a network of S servers. The service maintains a database of addresses mapped to their nearest server and the "distance" (latency) to that server as well as the distance between any two servers. The endpoint-endpoint distance estimation is achieved by adding the distances of each endpoint to their nearest server and the distances between the two servers. Effectively this method partitions the Internet address space into S clusters and then divides these clusters into equivalence classes. In contrast to our tech-

niques, [6] does not provide variable granularity and does not attempt to build clusters that are equivalent from the perspective of any location in the internet – only from the perspective of the nearest of the S servers.

An interesting extension of the IDmaps approach is described in [18] which demonstrates that given an incomplete set of hop by hop temporal measurements (e.g. from traceroute) between sets of endpoints, the results can often be extended to provide delay estimates between endpoints for which no direct measurements exist. While this approach combines temporal and topological measurements, it does not appear that it applies to the clustering problem discussed in this paper.

An alternative approach to clustering is to associate endpoints (seedpoints) with their nearest DNS server and then generalize the results. This approach is explored in [17] and [3]. An implementation of this idea to estimate end-end times is presented in [8]. While clustering based upon DNS servers appears to yield reasonably accurate clusters, it does not provide variable granularity – the method does not provide information that could be used to merge clusters. The approach could be combined with the temporal technique described in this paper with measurements to DNS servers replacing measurements to scanpoints.

[5] proposes a technique called *Internet Iso-bar* which clusters end hosts together based upon the "correlation distance" between hosts which includes possible techniques such as Euclidean distance and Cosine vector similarity. One representive from each cluster is chosen as a monitoring point. The application discussed is significantly different than ours – the end hosts are active participants in a peer-peer network and clustering is suggested as a way to achieve scaling. Nevertheless there are some important similarities with our work – the use of temporal distances as a clustering technique and the observation that accuracy and granularity can be traded off. [5] also provides some support for our approach by comparing the results of clustering using various metrics; the results suggest that Euclidean distance is the best metric from a set that includes the techniqe used by IDMaps. The authors also provide an important stability study which indicates that the clusters have good long term stability. It is important to note, as the authors do, that their data set was limited to various educational and research sites connected to Internet 2. Thus, it is not clear their conclusions generalize to the clustering problem addressed by our work.

The clustering technique described in this paper assumes that the routers being measured respond quickly to pings and that the latency of a path can be estimated using a median or minimum of a number of measurements. The first question was well addressed in [7] in which it was concluded that most routers reply quickly to TTL expired and hence hop limited probes. In the remainder of this section, we address the second question.

The stability of Internet path measurement was addressed in [19] which concluded that delay is highly predictable. More directly relevant is [4] which looked at delay histograms for packets sent from a source to a destination and concluded that the typical pattern is a gamma distribution with a long tail.

VI. Extensions

While the results presented in Section III are quite encouraging, there are areas in which the techniques could be improved. The three major areas in which the our techniques could be improved are

- Measurement
- Coverage
- Clustering

The implementation described in this paper utilizes ICMP echo requests to intermediate proxies for measurement. This has two major deficiencies – a reasonably high fraction of nodes do not respond to ICMP echo requests and we cannot guarantee that the paths followed by the measurement probes are coincident with the paths followed by packets to the corresponding seedpoints. The first deficiency leads to degraded coverage while the second had the potential for degrading the accuracy of the measurements.

In our experiments we found that roughly 7% of the measurement points did not respond at all to ICMP echo requests. These points had been discovered through ICMP based trace routes and hence do respond to TTL expired messages. An alternative measurement process could be implemented using TTL limited probes which would likely increase the proportion of "good" measurement points. Recall that the measurement points are proxies for a group of "equivalent" seedpoints. Given a representative

seedpoint, a server would initially perform a trace route to the seedpoint in order to determine the hop count to the penultimate hop and then perform measurements by utilizing TTL limited probes sent to the seedpoint. Some provision is needed for detecting when the responding machine changes. Initial experiments suggest that the responding nodes tend to be stable over many hours. An additional benefit to using TTL limited probes is that the probe is guaranteed to be routed using the same path as a packet to the seedpoint. Sending probes to intermediate proxies does not have this guarantee.

It is also a common belief that many networks filter ICMP echo request messages. For both the traceroutes and TTL limited probes other IP message formats could be used.

In our experiments we also found that there were many ASs for which we could find seedpoints, but no measurement point in the same AS. We rejected these seedpoints because we could not be reasonbly certain that the AS path followed by probes to a measurement point would match the AS path to the corresponding seedpoint. We were able to improve3 on our coverage by searching for alternative seedpoints in prefixes where the traceroute to a seedpoint did not have its penultimate hop in the same AS. This may be due to a frequent use of ".1" addresses for edge routers. With TTL limited probes, this would be a non-issue.

While the results described in this document have reasonable coverage (in a traffic weighted sense) there is still considerable room for improvement. We believe that TTL limited probes would have a positive impact upon coverage.

There are several ways in which clustering can be improved – utilizing geographic knowledge to assist the clustering process and better elimination of outlier data. As more measurement servers are added there is some risk that the amount of "noise" in the clustering process will become unacceptable We have experimented with utilizing "geographic" knowledge to allow us to associate measurement points with subsets of the servers. We determine the "region" in which a point resides by finding the "closest" server. Clustering can then be performed on the measurement points in a region using the most appropriate subset of servers. Since RTT measurements provide an upper bound on distance, one can reasonably bound the location of a measurement point given the time from its nearest server.

A variation of this approach is to utilize AS information to select server subsets. There are relatively few transcontinental ASs. With the decentralization of the AS registration process, we can utilize the registration service (RIPE, ARIN, LACNIC, APNIC) as an indicator of which server subset is appropriate. Given the technique described above, it is relatively straightforward to detect ASs which are likely to be transcontental. The rest could be clustered with servers chosen by AS.

Outlier elimination can be improved using the "geographic" approaches described above. It is reasonable to discard measurement points that are not within some maximum distance (time) of some server. This requirement can be tightened if the set of servers is constrained based upon the AS number.

Finally, given a set of destinations with known geographic locations (e.g. web servers in various cities), we can determine the temporal locations using the same set of servers used to perform the temporal measurements for clustering. We call these destinations mileposts. Our clustering process provides, for each cluster, a centroid (either mean or median). The cluster can then be associated with the geographic location of the milepost that is nearest to the cluster centroid. Alternatively, one might find the centroid of the most important subset of the cluster points based upon traffic data and then associate the geographic location with that.

REFERENCES

[1] Matthew Andrews, Bruce Shepherd, Aravind Srinivasan, Peter Winkler, Francis Zane. "Clustering and Server Selection using Passive Monitoring" in *Proceedings of the IEEE Conference on Computer Communications* (INFOCOM) 2002.

[2] Paul Barford, Azer Bestavros, John Byers, and Mark Crovella. "On the marginal utility of network topology measurements" in *Proceedings of the 2001 Sigcomm Internet Measurement Workshop*, October 2001.

[3] Azer Bestavros and Sumit Mehrotra. "DNS-based Internet Client Clustering and Characterization" in *Proceedings of the 4th IEEE Workshop on Workload Characterization* (WWC) December 2001.

[4] C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, P. van Mieghem. "Analysis of End-to-end Delay Measurements in Internet" in *Workshop of Passive and Active Measurement* (PAM) 2002.

[5] Yan Chen, Khian Hao Lim, and Randy H. Katz. "On the Stability of Network Distance Estimation" in *ACM SIGMETRICS Performance Evaluation Review*, Vol. 30, Issue 2, September 2002.

[6] P. Francis, S. Jamin, C. Jin, Y.Jin, D. Raz, Y. Shavitt, L. Zhang. "IDMaps: A Global Host Distance Estimation Service" in *IEEE/ACM Transactions on Networking*, October 2001

[7] Ramesh Govindan and Vern Paxon. “Estimating Router ICMP Generation Delays” in *Workshop of Passive and Active Measurement*(PAM) 2002.

[8] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. “King: Estimating Latency between Arbitrary Internet End Hosts” in *2nd Internet Measurement Workshop*, November 2002.

[9] Bradley Huffaker, Marina Fomenkov, Daniel Plummer, David Moore, and K Claffy. “Distance Metrics in the Internet” in *IEEE International Telecommunications Symposium* 2002.

[10] Anil K. Jain and Richard C. Dubes. “Algorithms for Clustering Data”. Prentice hall, Englewood Cliffs, NJ, 1988.

[11] Balachander Krishnamurthy, Jia Wang. “On Network-Aware Clustering of Web Clients” in *Proceedings of ACM Sigcomm 2000*.

[12] A set of GPL clustering tools implementing the algorithms in [10]. [Online] Available: http://odur.let.rug.nl/∼kleiweg/clustering/.

[13] Published route information for Level 3 available [Online]: http://www.level3.com

[14] T. S. Eugene Ng, Hui Zhang. “Predicting Internet Network Distance with Coordinates-Based Approaches” in *Proceedings of the IEEE Conference on Communications* (Infocom) 2002.

[15] Venkata N. Padmanabhan, Lakshiminarayanan Subramanian. “An Investigation of Geographic Mapping Techniques for Internet Hosts” in *Proceedings of ACM Sigcomm* 2001.

[16] http://www.routeviews.org

[17] Anees Shaikh, Renu Tewari,and Mukesh Agrawal. “On the Effectiveness of DNS-based Server Selection” in *Proceedings of the IEEE Conference on Computer Communications* (INFOCOM) 2001.

[18] Yuval Shavitt, Xiaodong Sun, Avishai Wool, and Bülent Yener. “Computing the Unmeasured: An Algebraic Approach to Internet Mapping” in *Proceedings of the IEEE Conference on Computer Communications* (INFOCOM) 2001.

[19] Yin Zhang, Nick Duffield, Vern Paxson, Scott Schenker. “On the Constancy of Internet Path Properties” in *Proceedings of the ACM Sigcomm Internet Measurement Workshop* 2001.

[20] Amagad Zeitoun and Sugih Jamin. “Fast Discovery of Live Interent Address Prefixes”. [Online] Available: http://idmaps.eecs.umich.edu/papers/ap.pdf